USER MANUAL

3D-MIMO-RADAR

Devices: *iSYS-5021*

Revision: *1*

Date: *22.11.2018*

Table of content

History

Document revision	Date	Change log	Author
1	2018-11-22	First release	JK

Documents

[1] iSYS serial interface protocol rev19.pdf

[2] serial radarSDK readMe rev16.pdf

[3] target viewer manual rev13.pdf

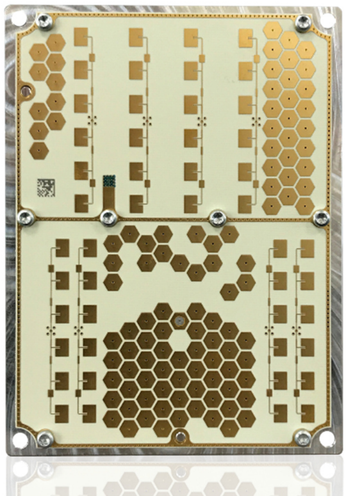
[4] EthernetAPI readMe rev4.pdf

[5] iSYS-5xxx Ethernet Targetlist Protocol_rev4.pdf

[6] smart tracker user manual rev1.pdf

1. Brief description

The iSYS-5021 is a 24 GHz MIMO (multiple input multiple output) RADAR system with integrated signal processing for security applications.



Features:

- Simultaneous capture of speed, range and angle information for up to 256 targets
- Detection range up to 150 m
- Velocity measurement up to 35.3 km/h
- Field of view 150° (±75°) in azimuth with a resolution of max. 12°
- Target list output on SPI (default) or Ethernet
- UART communication for RADAR sensor configuration
- Temperature range from -40°C to +70°C.
  (Extended temperature -40°C - +85°C with SPI only on request)
- Includes InnoSenT Smart Tracker license

2. Technical specifications

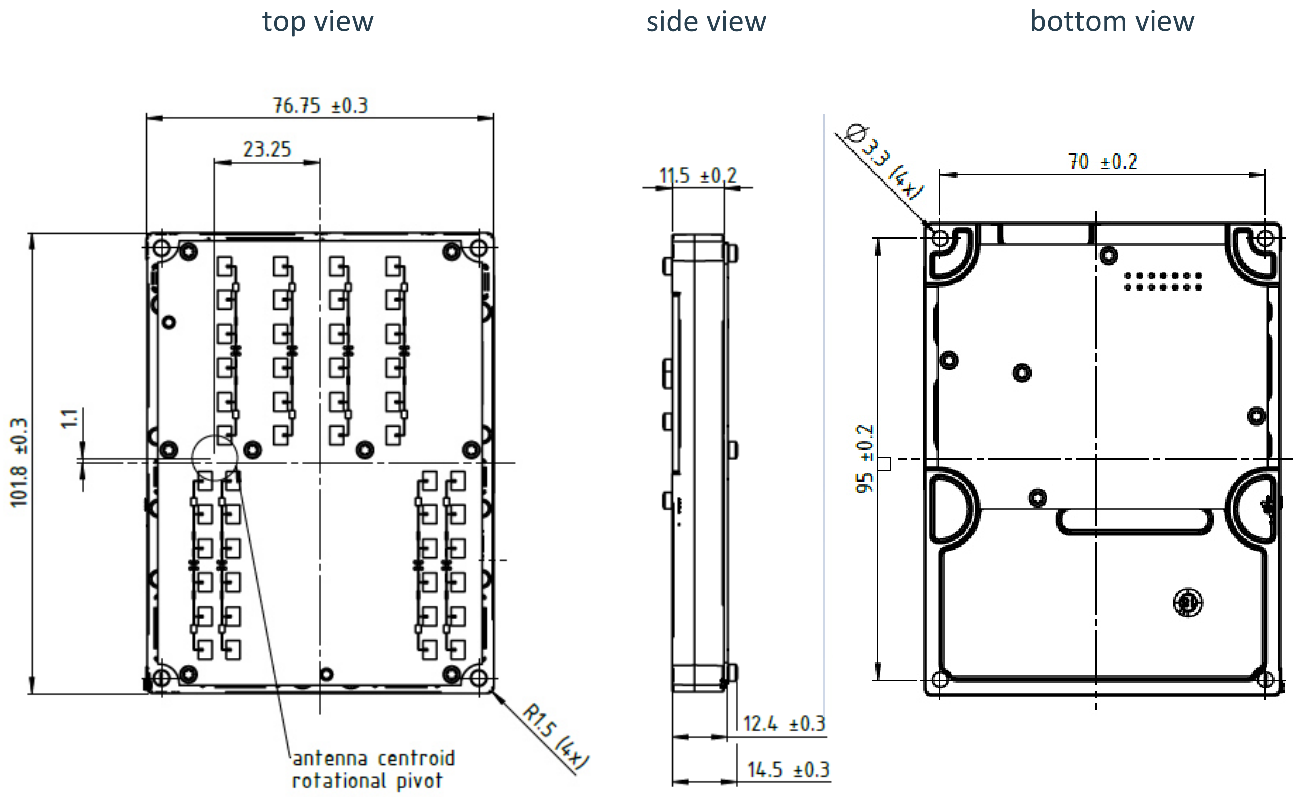
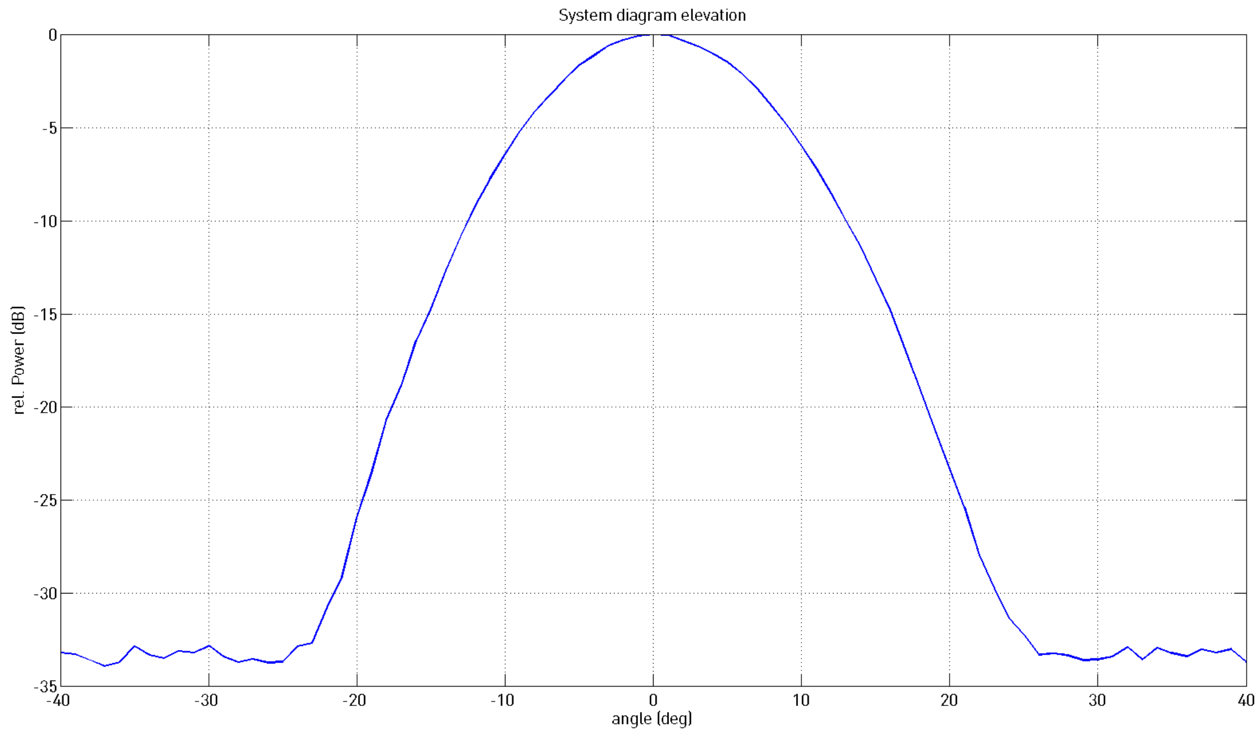


Figure 1: Mechanical schematic

2.1. System parameters

2.1.1. Antenna specification

The antenna patterns for the transmitters and the receivers are shown below:



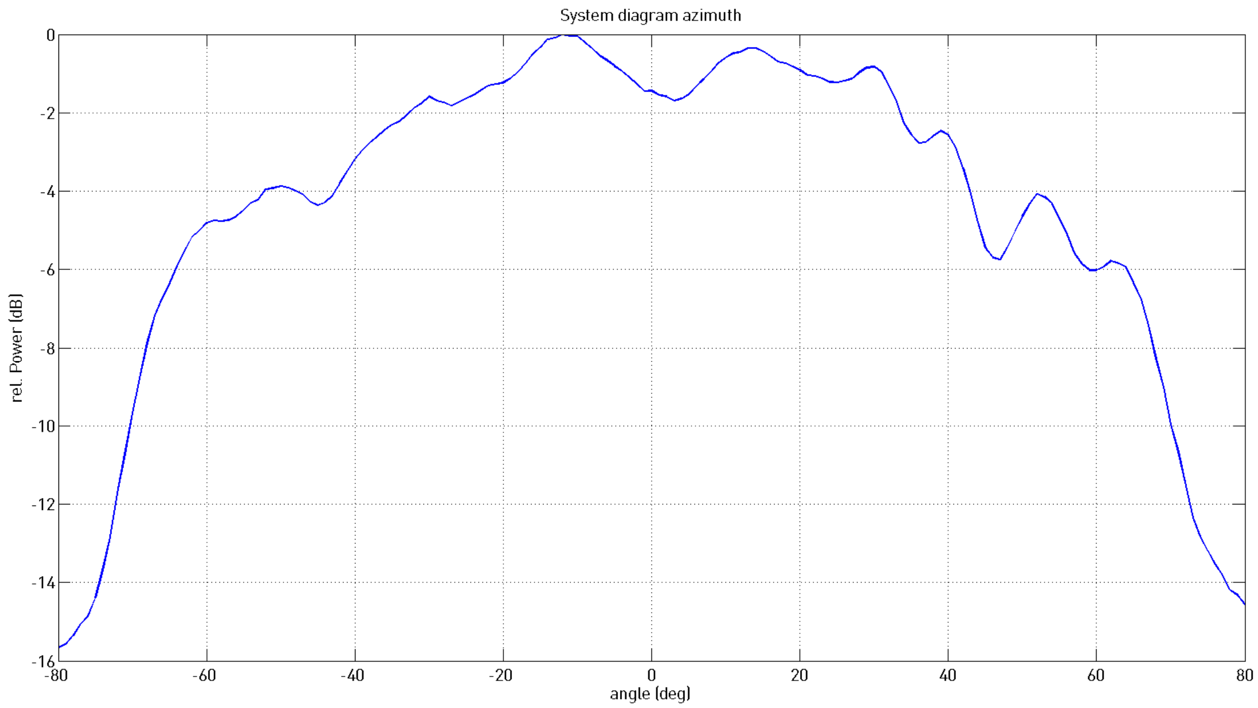


Table 1: Transmitters – Receivers Antenna Pattern

2.1.2. Angle measurement

The iSYS-5021 is specified with the following angle characteristics:

Measurement range	Resolution (max.)	Error
± 55 °	12 °	- 1.0 ° ... + 1.0°
± 75 °	12 °	not specified

Table 2: Angle measurement

2.1.3. Range measurement

The iSYS-5021 RADAR sensor is specified for the range:

Measurement range	Resolution	Error
1 m ... 150 m	1.26 m	± 1.5 m

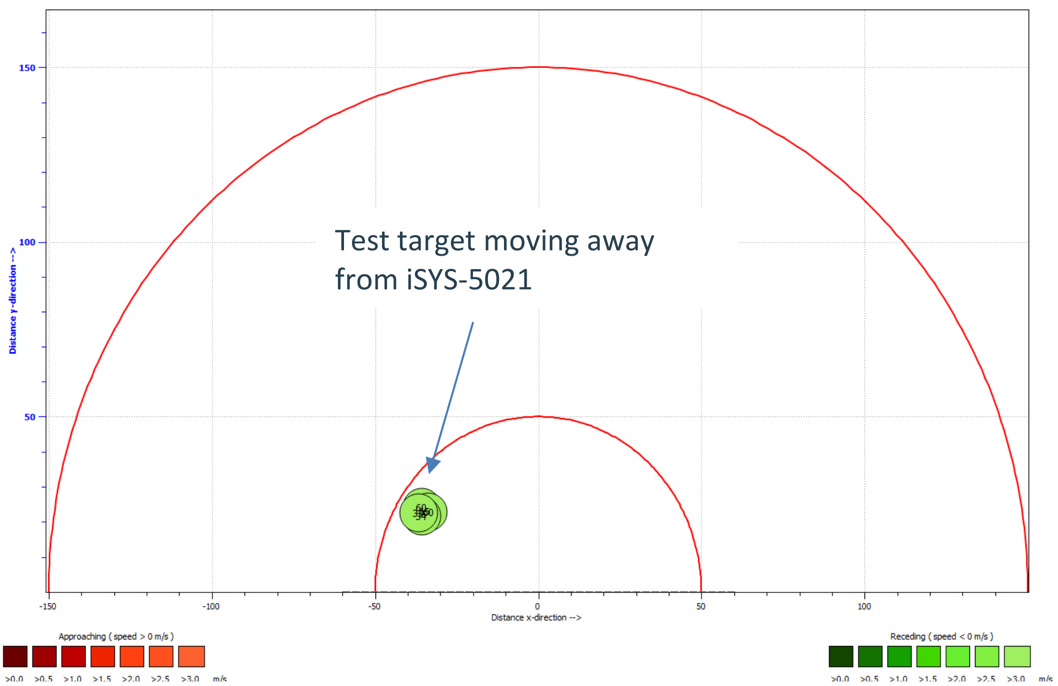
Table 3: Range measurement

The measurement precision is ensured in the range of 1 m to 150 m.

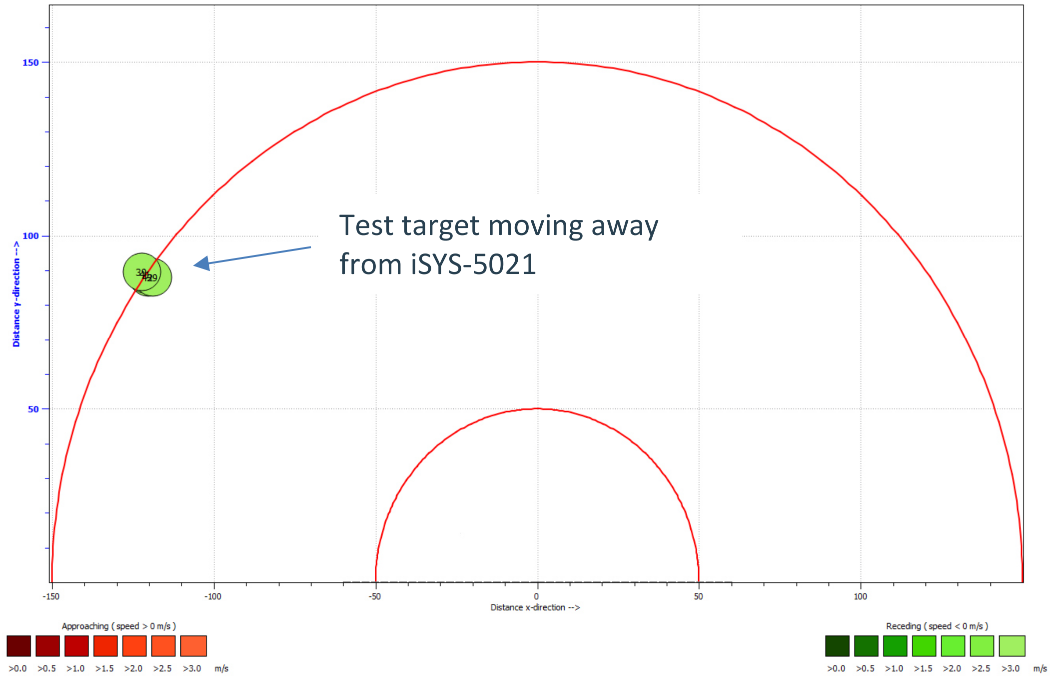
2.1.4. Range ambiguity detection scheme

With our specific iSYS-5021 modulation scheme, the issue of range ambiguity is addressed. Range ambiguity is encountered when a huge target inside radar field of view (FOV) is located further away than maximum specified measurement range (s. Table 3: Range measurement).

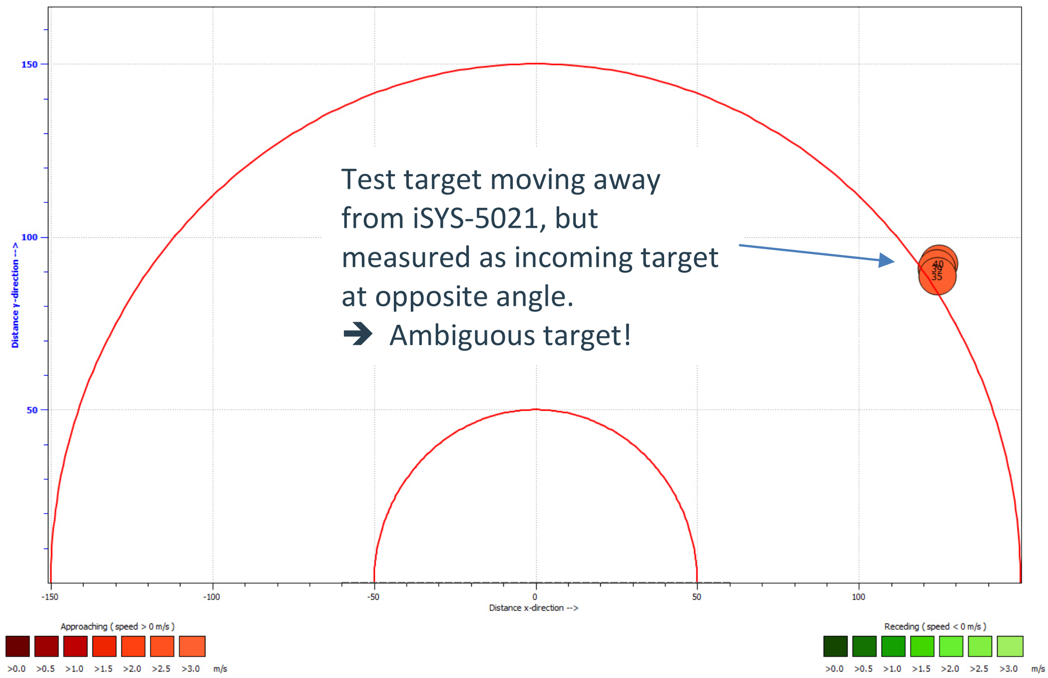
The effect of the range ambiguity is as shown with the following example, where a huge target is moving away from iSYS-5021:



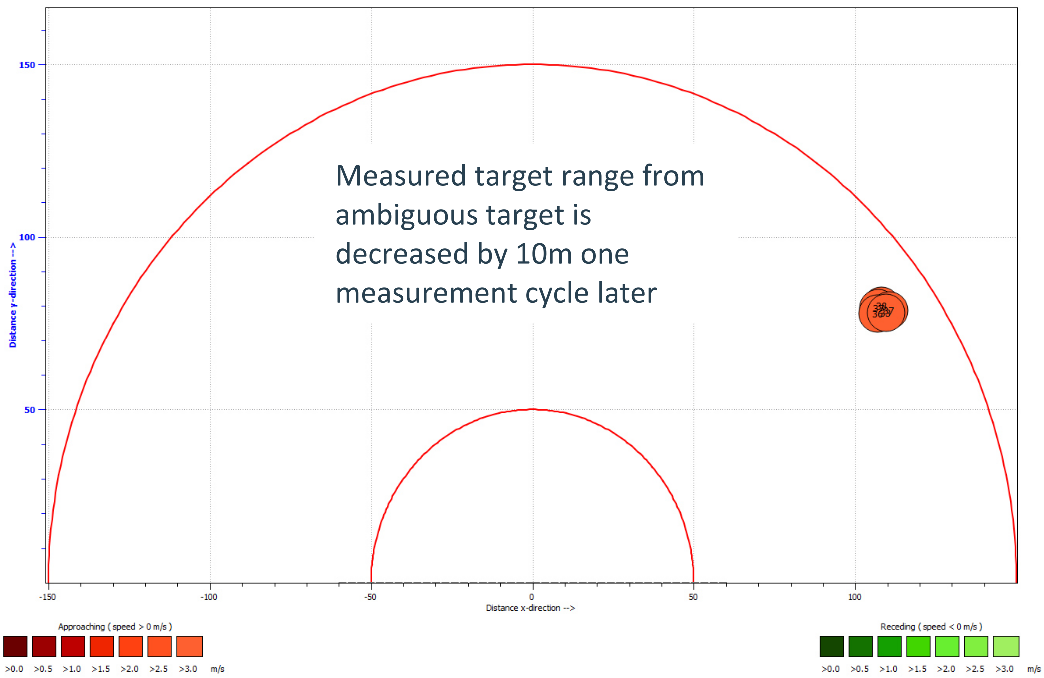
A couple of seconds later the target reached the maximum distance of 150m:



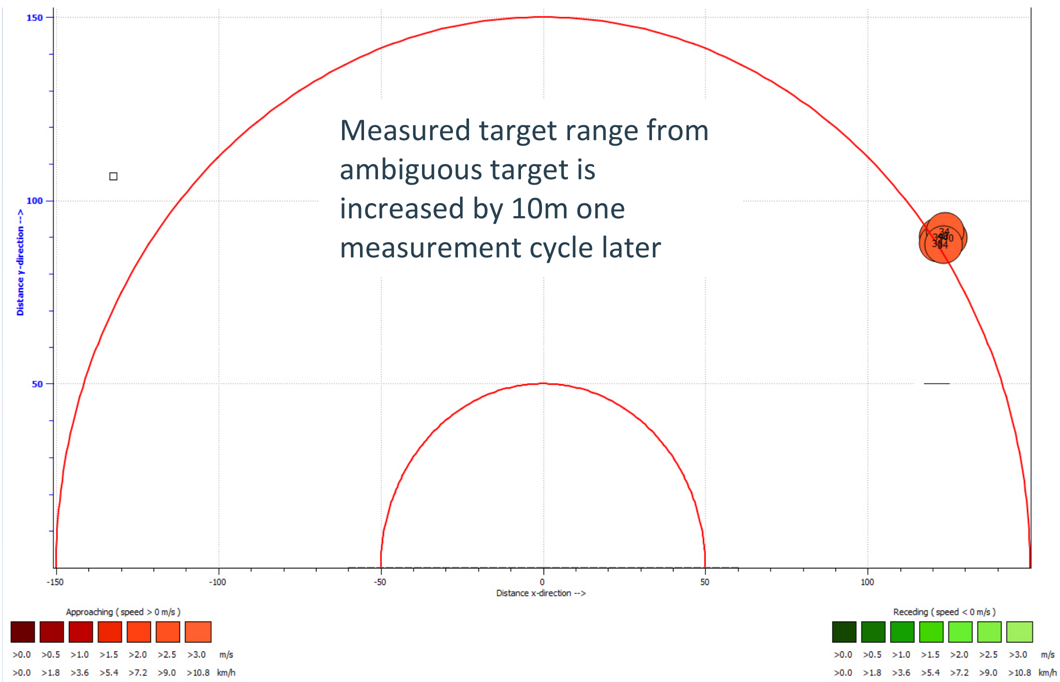
The target is moving further away from the radar and is now detected with wrong range, speed and angle:



Due to our special modulation implemented in iSYS-5021 the ambiguous target jumps between two consecutive measurements about 10m back and forth. This is not the case for the unambiguous target.



The next measurement cycle shows this clearly:



Your application software (e.g. tracking) will take advantage of that jumping behavior of an ambiguous target and thus false detections can be avoided.

2.2. *Functions*

The iSYS-5021 radar sensor has the ability to be configured through the UART communication port.
The following functions are available and can be activated or deactivated by commands:

- Frequency channels:
  set a different frequency channel for iSYS-5021
- Clustering:
  reduce the amount of targets in the target list by clustering nearby targets together
- RCS calculation:
  choose between signal strength information and RCS (i.e. radar cross section) information of the targets

2.2.1. **Frequency channels**

There are two channels available for the iSYS-5021. The use of different channels is recommended when two radar systems are working in the same area. Yet, two iSYS-5021 may interfere with each other, when they are working in close proximity.

3. Installation

The test installation is shown below in Figure 3. Typically, the responses of the targets of interest for the RADAR are considered to be at least at 1m in height. This reference height corresponds to the torso, for a person, or the mean response height for a car. In Table 4, several installation configurations are suggested. At InnoSenT, the RADAR sensor is typically installed on a pole at a height h of 3m and an angle position α of 1.5°. This configuration has been chosen in order to optimize the maximum range of the iSYS-5021 RADAR sensor.

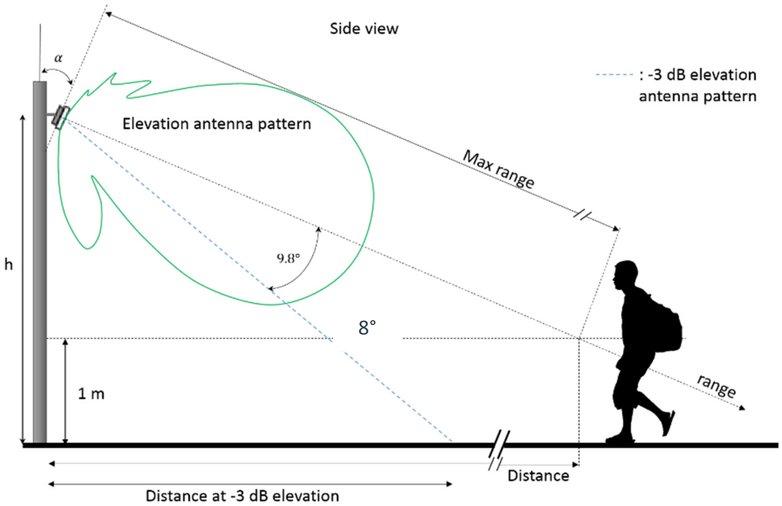


Figure 3: Installation example

Height h	Angle α	typ. Detection distance for a person[1]	Distance at – 3 dB elevation
3 m	1.5°	≥ 120m	12.0 m

Table 4: Installation parameters

The proper functionality of the iSYS-5021 RADAR sensor can only be achieved if all installation parameters are set correctly. Thus, please ensure that the complete field of view is clear without any obstructions. Otherwise, shadowing effects may occur that limit the detection performance.

[1] Target angle from -55deg to + 55deg. Determined by application tests at InnoSenT in Donnersdorf – Germany.

4. Connection

4.1. Connector placement

The iSYS-5021 provides a 14 Pin 2.54 mm pitch female header. This connector (W+P 3492-14-3-00-00) is a dual entry type and is mounted on the inner side of the DSP-board. InnoSenT uses a gold-plated connector. The length of the mating connector should be 5.5mm ± 0.5mm (e.g. W+P 3132-12-14-00-0-ST). The pin description is given in Table 5.

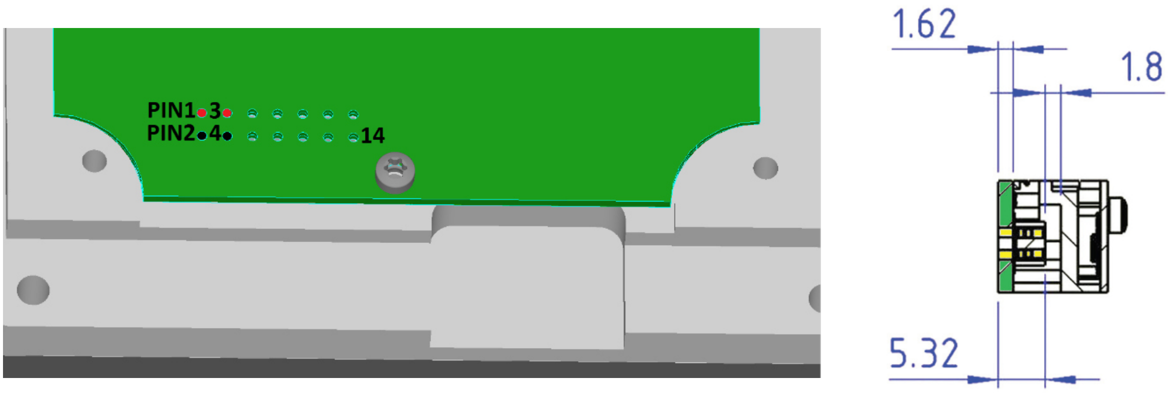


Figure 4: iSYS-5021 connector

PIN	NAME	DESCRIPTION
#1 ●	6V2_IN	Power Supply, 1000mA max.
#2 ●	GND	
#3 ●	6V2_IN	
#4 ●	GND	
#5	SPI_CLK	SPI → default interface for target list output, CLK 5 MHz max.
#6	ETHERNET TPIA_P	ETH → secondary interface for target list output, 100 Mbit/s
#7	SPI_CS	SPI → default interface for target list output
#8	ETHERNET TPIA_N	ETH → secondary interface for target list output
#9	SPI_MOSI	SPI → default interface for target list output
#10	ETHERNET TPIB_P	ETH → secondary interface for target list output
#11	SPI_MISO	SPI → default interface for target list output
#12	ETHERNET TPIB_N	ETH → secondary interface for target list output
#13	UART_RX	UART → command interface for configuration (3.3 V)
#14	UART_TX	

Table 5: Connector Pin-out

4.2. Power supply

	Comment	Symbol	min.	typ.	max.	Unit
Supply voltage		V_{CC}	6.1	6.25	6.4	V
Ripple		V_{ripple}			5	mV
Supply current		I_{cc}			1000	mA

Table 6: Power supply parameters

5. Communication

The iSYS-5021 has two interfaces:

- UART — command interface (bidirectional)
- Ethernet / SPI — target list output interface

  (default interface is SPI; can be changed with target viewer software - see. document 'target viewer manual')

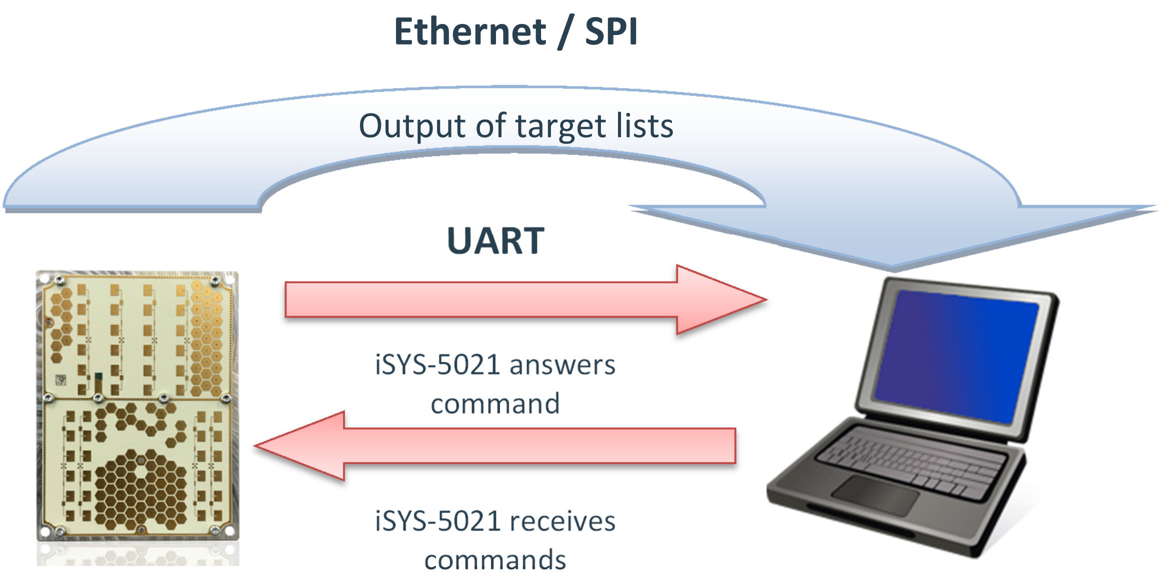


Figure 5: UART and Ethernet / SPI communication

5.1. UART command interface

The UART interface is available for device configuration and firmware update purposes.

There are two different ways for implementation served by InnoSenT:

- serial protocol description including commands for self-implementation
- serial RadarAPI with C++ example project

5.1.1. UART configuration

	value	comment
Transmission direction	full duplex	
Baud rate	115200	Baud
Data	8	
Stop-bits	1	
Parity	none	
Voltage level	3.3	V, CMOS

Table 7: UART configuration

5.1.2. Serial protocol description

InnoSenT uses a companywide serial protocol for all iSYS devices with a serial interface. The iSYS serial interface protocol document [1] describes this protocol and all commands.

Note that each device uses only a small subset of this commands. The supported commands are listed in the appendix of the document [1].

5.1.3. Serial RadarAPI description

InnoSenT provides a C++ serial RadarAPI. This API contains the functions for the complete InnoSenT serial interface protocol. The serial RadarAPI is explained in detail in document [2].

The iSYS-5021 supports following commands from the serial RadarAPI:

- iSYS_ReadDeviceName
- iSYS_getDeviceAddress
- iSYS_StartAcquisition / iSYS_StopAcquisition
- iSYS_setFrequencyChannel / iSYS_getFrequencyChannel
- iSYS_setIPConfig / iSYS_getIPConfig
- iSYS_setIPDestination / iSYS_getIPDestination
- iSYS_setTargetListInterface / iSYS_getTargetListInterface
- iSYS_getFirmwareVersion
- iSYS_getSerialNumber
- iSYS_getDspHardwareVersion
- iSYS_getRfeHardwareVersion
- iSYS_getProductInfo

5.2. SPI target list interface

The iSYS-5021 processes the radar signals to produce a target list which will be sent every 100ms via SPI interface. One complete target list will be sent within one frame with a constant offset of a measurement cycle. Each frame consists of a sum of sub-frames (start/end sub frame, target sub frame). Each sub-frame will be sent within one chip-select signal. A schematic SPI target list transmission is shown in Figure 6 below.

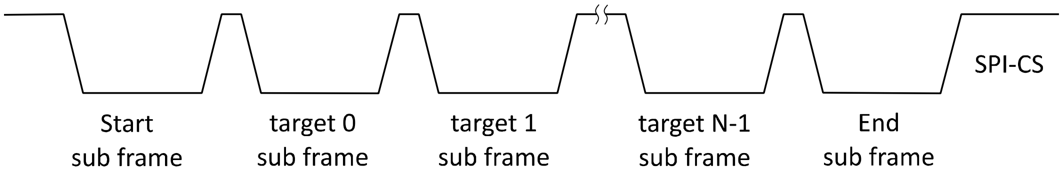


Figure 6: schematic example of SPI target list frame

5.2.1. SPI configuration

	value	comment
Transmission direction	simplex, Master	**iSYS-5021 is Master device**
Bit order		MSB first
SCLK frequency	5	MHz
Data word size	32	Bit
CS		Chip select active low
CPOL	0	Clock idle low
CPHA	1	Falling edge
Voltage level	3.3	V, CMOS
Max number of targets	256	

Figure 7: SPI communication parameter

Frame transmission time for maximum number of targets:

The needed transmission time for the maximum target list with 256 targets can be calculated with the formula below. The factor 1.01 is used for the overhead caused by inter sub frame chip selects.

$$\frac{(256\,targets \cdot 6\,DWORD + 4\,DWORD) \cdot 32\frac{Bit}{DWORD}}{5\,MHz} \cdot 1.01 = 9.95\,ms \approx 10ms$$

5.2.2. Target-List information content definitions

The target list consists of up to 256 targets. For each target the signal strength / RCS, range, velocity and angle is transmitted. The signal strength corresponds to the scaled FFT magnitude calculated during signal processing. It is also possible to transmit the RCS instead of the signal strength. For changing the output between signal strength and RCS (RADAR cross section) refer to the serial RadarAPI [2] or the serial interface protocol [1]. The definition of the sign for the transmitted velocity and angle value is shown in Figure 8:

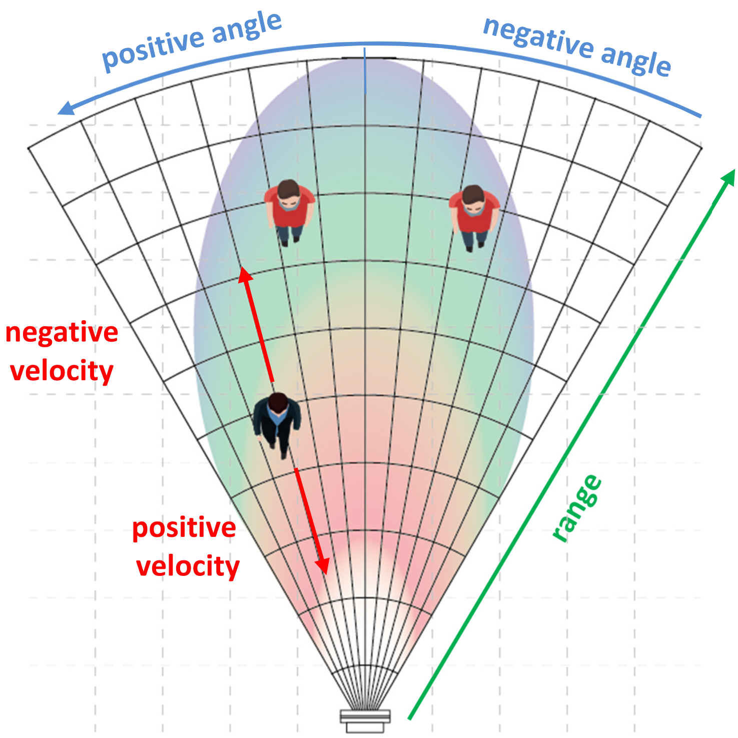


Figure 8: target information contained in target sub frame

5.2.2.1. SPI Target list frame

The SPI target list uses three different sub frame types:

- **Start sub frame**

This sub frame is used to mark the start of a target list frame.

Double word number	**Data type**	**description**
0	Uint32_t	Start identifier 0xFEED5021
1	Uint32_t	number of targets N

Table 8: SPI target list start sub frame

- **Target sub frame**

This sub frames contain the information for one of the N targets. For each target one target sub frame is transmitted.

Double word number	**Data type**	**description**
0	Float32_t	Signal strength in dB or RCS in m^2
1	Float32_t	Range in m
2	Float32_t	Velocity in m/s
3	Float32_t	Angle azimuth in degree
4	Float32_t	Reserved 1
5	Float32_t	Reserved 2

Table 9: SPI target list target sub frame

- **End sub frame**

This sub frame is used to mark the end of a target list frame. It contains the checksum of the target list frame. The checksum is the sum of the bytes over all target sub frames.

Double word number	**Data type**	**description**
0	Uint32_t	End identifier 0xFEED0000
1	Uint32_t	Target list checksum

Table 10: SPI target list end sub frame

Example for SPI target list

The following example show a possible target list with two targets. An example for calculation the target list checksum is also included below

Sub frame	Sub frame content		
Start frame	Start identifier	0xFEED5021	
	Target number	0x00000002	
Target frame 0	RCS	0x41200000	Data used for checksum calculation
	Range	0x41c9999a	
	Velocity	0x3f800000	
	Angle azimuth	0x4121eb85	
	Reserved 1	0x00000000	
	Reserved 2	0x00000000	
Target frame 1	RCS	0x40a00000	
	Range	0x420e6666	
	Velocity	0x3dcccccd	
	Angle azimuth	0xc1700000	
	Reserved 1	0x00000000	
	Reserved 2	0x00000000	
End frame	End identifier	0xFEED0000	
	Target list checksum	0x00000AFE	

Table 11: Example target list

Calculation of the frame checksum:

$$\begin{aligned} CKSUM = {} & 0x41 + 0x20 + 0x41 + 0xc9 + 0x99 + 0x9a + 0x3f + 0x80 + 0x41 + 0x21 \\ & + \mathrm{0xeb} + \mathrm{0x85} + \mathrm{0x40} + \mathrm{0xa0} + \mathrm{0x42} + \mathrm{0x0e} + \mathrm{0x66} + \mathrm{0x66} \\ & + \mathrm{0x3d} + \mathrm{0xcc} + \mathrm{0xcc} + \mathrm{0xcd} + \mathrm{0xc1} + \mathrm{0x70} = \mathrm{0xAFE} \end{aligned}$$

5.2.2.2. Decoding a target list sub frame

Following code can be used to decode the target sub frame

```
inputArray[0] = static_cast<float32_t>( ((rxArray[0] BIT_AND 0x000000FF) << 24) BIT_OR  /* signal strength  or rcs*/
                ((rxArray[1] BIT_AND 0x000000FF) << 16) BIT_OR
                ((rxArray[2] BIT_AND 0x000000FF) << 8)  BIT_OR
                (rxArray[3] BIT_AND 0x000000FF) );

inputArray[1] = static_cast<float32_t>( ((rxArray[4] BIT_AND 0x000000FF) << 24) BIT_OR  /* range */
                ((rxArray[5] BIT_AND 0x000000FF) << 16) BIT_OR
                ((rxArray[6] BIT_AND 0x000000FF) << 8)  BIT_OR
                (rxArray[7] BIT_AND 0x000000FF) );

inputArray[2] = static_cast<float32_t>( ((rxArray[8] BIT_AND 0x000000FF) << 24) BIT_OR  /* velocity */
                ((rxArray[9] BIT_AND 0x000000FF) << 16) BIT_OR
                ((rxArray[10] BIT_AND 0x000000FF) << 8)  BIT_OR
                (rxArray[11] BIT_AND 0x000000FF) );

inputArray[3] = static_cast<float32_t>( ((rxArray[12] BIT_AND 0x000000FF) << 24) BIT_OR /* angle azimuth*/
                ((rxArray[13] BIT_AND 0x000000FF) << 16) BIT_OR
                ((rxArray[14] BIT_AND 0x000000FF) << 8)  BIT_OR
                (rxArray[15] BIT_AND 0x000000FF) );

inputArray[4] = static_cast<float32_t>( ((rxArray[16] BIT_AND 0x000000FF) << 24) BIT_OR /* reserved1 */
                ((rxArray[17] BIT_AND 0x000000FF) << 16) BIT_OR
                ((rxArray[18] BIT_AND 0x000000FF) << 8)  BIT_OR
                (rxArray[19] BIT_AND 0x000000FF) );

inputArray[5] = static_cast<float32_t>( ((rxArray[20] BIT_AND 0x000000FF) << 24) BIT_OR /* reserved2*/
                ((rxArray[21] BIT_AND 0x000000FF) << 16) BIT_OR
                ((rxArray[22] BIT_AND 0x000000FF) << 8)  BIT_OR
(rxArray[23] BIT_AND 0x000000FF) );
```

Table 12: Decode target information of a target sub-frame

5.3. Ethernet target list interface

InnoSenT provides a C++ Ethernet API. This API contains the functions to receive the sensor target list. The Ethernet API is explained in detail in document [4].
For detailed information about the Ethernet targetlist protocol please refer to document [5].

5.4. Target-Viewer

InnoSenT provides a graphical user interface called Target-Viewer for displaying the targets and configuring the iSYS-5021 by UART communication.

In SPI target list mode the Target-Viewer works only with the Beagle SPI Protocol-Analyzer from Total-Phase.

In Ethernet target list mode no special hardware is required.

For further information regard document [3].

6. Smart Tracker License

The license of the Smart Tracker that is included in the iSYS-5011 delivery package, allows the usage of the Smart Tracker which can easily be integrated in various applications. The Smart Tracker is integrated in the InnoSenT Tracker Library (ITL) which can be integrated easily in specific programs and interfaces. Therefore the tracking process does not run on the iSYS-5011 itself.

The Smart Tracker processing includes several post-processing algorithms such as target filtering and transformation, object plausibilization, object recognition and location over time and classification. The Smart Tracker is developed for urban environment and perimeter protection. For this reason it is optimized for pedestrian recognition. Every tracked object gives information about object class, velocity, direction, position, the age and quality of the track and has a unique object ID for identification. For further information how to integrate the ITL in own application regard document [6].

The InnoSenT Tracker Library does not support all processor families.

InnoSenT GmbH
Am Roedertor 30
97499 Donnersdorf
GERMANY
Tel.: +49 95289518-0
E-Mail: info@innosent.de
www.innosent.de