

Software Development Guidance

Introduction

The [Navtech Radar SDK](#) provides a simple interface to communicate with Radar sensors. As well as ROS and ROS2 examples, the SDK includes source code for Python, C++ & a simple .NET DLL that can easily be integrated into applications running on Windows and Linux.

Navtech Radar sensors provide output in one of four modes:

1. Colossus [FFT data](#) (sometimes referred to as “raw” radar data. Communication with Navtech Radar sensors is over a TCP/IP connection and utilises a proprietary binary communication protocol called [Colossus Network Protocol](#). The SDK abstracts this protocol to avoid having to develop the low level socket and messaging processing code. However you can develop your own interface without the SDK if required and the protocol documentation will provide all the information you need to achieve this. Code downloads are accessible [here](#).
2. [Navigation output](#) - a simple point output format from the radar using it's [Colossus Network Protocol](#). In this mode, the radar applies a threshold to the data representing its' field of view, and a range value in units of metres is returned, corresponding to the “centre” of each peak in the radar's measurements. The SDK provides support for this mode, allowing developers to receive either type of data from the sensor and also send control messages back.
3. ASTERIX Cat240 (Radar Video) - An industry standard for the “transmission of rotating radar video” over a UDP network connection.
4. CFAR Point Cloud output - a UDP stream of bearing and range (measured in integer “bins”) point-pairs corresponding to any “bins” of data that pass a Cell-Averaging CFAR extraction process. When a radar is configured to operate in this mode via its' web interface, the configuration parameters for this mode are sent to the radar via the [Colossus Network Protocol](#) over a TCP/IP connection.

i Note that when consuming radar output in the CFAR Point Cloud mode or as “Raw” data, conversion from integer bin “number” to range in metres can be done by using the multiplication factor named “*Bin Size / Resolution*” within the radar's [configuration](#) message.

Range-reporting accuracy can be further improved by using the radar-unique calibration coefficients “*Range gain*” and “*Range offset*”, also reported within the payload of the configuration message. Applying these coefficients will ensure that any device-specific tolerances are accounted for when converting from bin “number” to range in metres.

Contents

- [IA Sensor Modes](#)
 - [FFT Data](#)
 - [Navigation Data](#)
-

IA Sensor Modes

FFT Data

In this mode the radar outputs FFT processed data, which is amplitude and range. Data is sampled 400x per radar rotation and packet is sent across the network for each sample. This packet includes the sample azimuth enabling the receiver to analyse target data based on range, bearing and amplitude. The data output rate for native mode is quite high. The normal data rate is between 11Mbps - 38Mbps.

To use native mode you will need to understand and use the *Colossus Network Protocol*. The protocol describes the format of the messages sent across the network and also explains how to control functions on the sensor. For example, after you connect to the radar you must ask the device to start sending FFT data. This ensures the receiving client is in full control of when it starts and receives raw data.

When working with native data the developer is responsible for performing all the signal processing to extract targets and establishing logic to discriminate targets from unwanted clutter.

The SDK provides a simple hook to enable your application to receive this data and also provides the appropriate methods to control the radar, such as switching the data stream on and off.

Navigation Data

In this mode the radar will analyse the data in real time and extract targets on each azimuth that are above a specified threshold. Messages describing these plots are then sent across the network as they are extracted.

In addition, this process will sub-resolve the range of the plot using a curve-fitting algorithm on the signal peak. This enables the sensor to achieve a far higher range resolution than the native signal processing normally allows. The algorithm will provide all targets in an azimuth that are above the specified threshold.

The target data are returned with the following information:

- Azimuth
- Seconds
- SplitSeconds
- List of targets
 - Range (m)
 - Power (dB half steps e.g 152 = 76dB)

The seconds and split seconds are using the UNIX epoch of the 1st Jan 1970 00:00:00; split seconds is the number of elapsed nano seconds since the last second

The mode of the radar's operation provides data at a much lower rate than the raw FFT data output and is ideally suited to support collision avoidance navigation and positioning applications without much further development.

The SDK provides a simple hook to enable your application to receive this data and also provides the appropriate methods to control the radar. This includes the ability to switch the plot extraction on and off and also allows the threshold to be set.