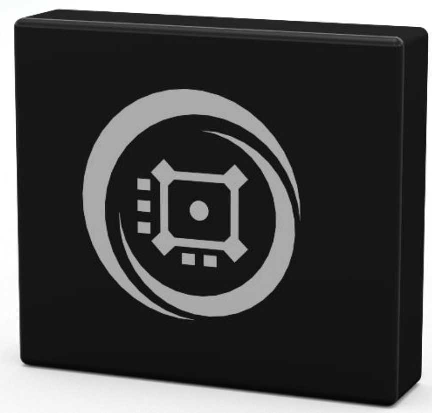
OCULII - EAGLE
USER MANUAL



EDITION 0.5.41
Dec 2021

Oculii Corporation

829 Space Drive, Beavercreek, Oh 45434, USA

Phone: 937-829-0383
Email: info@oculii.com
Website: www.oculii.com

Table of Contents

1. OVERVIEW

Powered by **Virtual Aperture Imaging,** Oculii's EAGLE 77GHz Point Cloud Radars can deliver thousands of points per second, capturing all relevant environmental information. Oculii's radar point clouds perform in all weather conditions, and each point directly measures highly accurate doppler information, enabling immediate separation and efficient tracking of any moving targets. **Virtual Aperture Imaging (VAI)** is an array multiplier technique that can be used on any transceiver architecture. The Oculii EAGLE is a single chip, automotive grade sensor. Using VAI, Oculii's EAGLE can achieve <1^0 angular resolution across an Azimuth Field of View of 110 degrees and <1^0 angular resolution across an Elevation Field of View of 45 degrees.

2. RADAR TECHNICAL SPECIFICATIONS

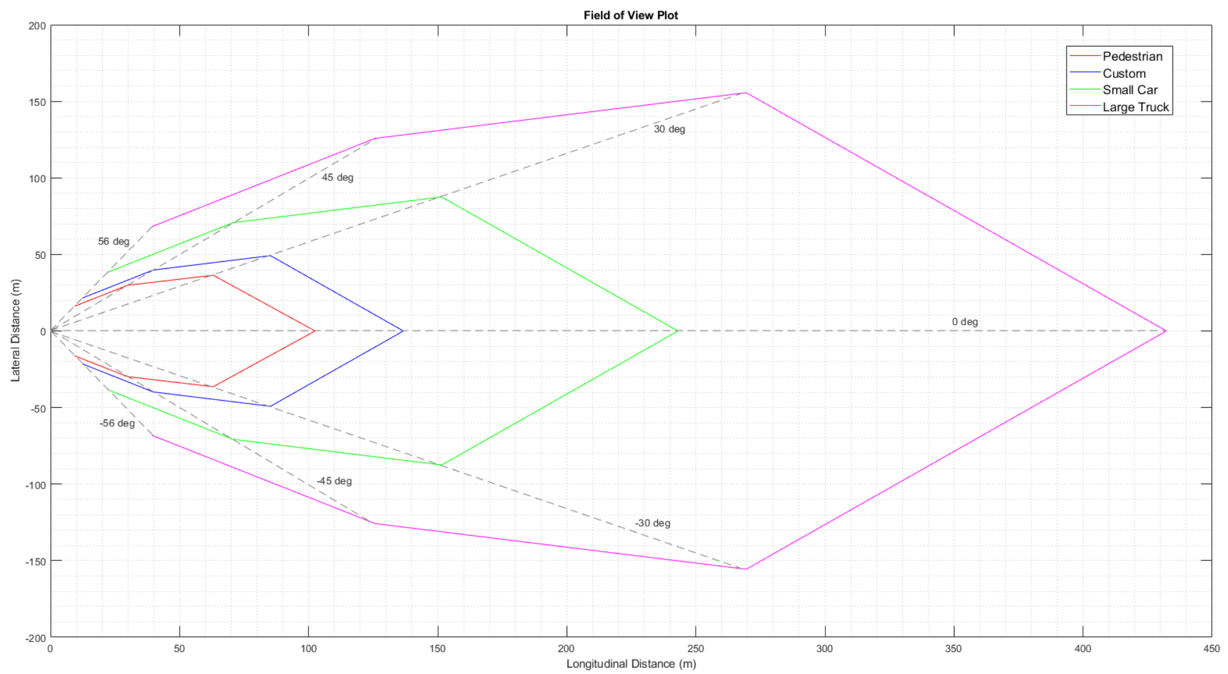
Frequency	77.0 – 79.0GHz
Detection Range	0m – 400m
Range Resolution	0.86m
Range Accuracy	0.86m
Azimuth Angle Range	-56.5^0 to $+56.5^0$
Azimuth Angle Resolution	$< 1^0$
Azimuth Angle Accuracy	0.44^0
Elevation Angle Range	-22.5^0 to 22.5^0
Elevation Angle Resolution	$< 1^0$
Elevation Angle Accuracy	0.175^0
Max Speed Range	-86.8m/s to +86.8m/s
Speed Resolution	0.27 m/s
Speed Accuracy	0.09 m/s
Points Per Second	21,000
Cycle Time	66ms (15Hz)
Data Output Format	UDP
Weight	644g
Dimensions (WxHxD)	130 x 115 x 31.6 mm
Power Consumption	7.5 W
Operation Temperature	-40 to +105 C

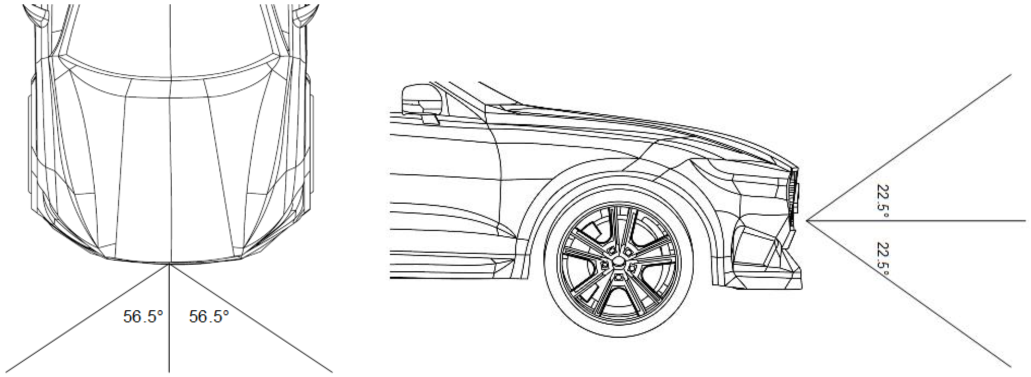
3. PRODUCT

3.1 Version

Product Version Number: EAGLE

3.2 Field of View





3.3 Application Examples

The EAGLE is equipped both with a point cloud capable of outputting thousands of points per second, as well as an embedded tracker capable of simultaneously tracking up to 100 objects. This allows the EAGLE to be used for several applications, some of which are listed below:

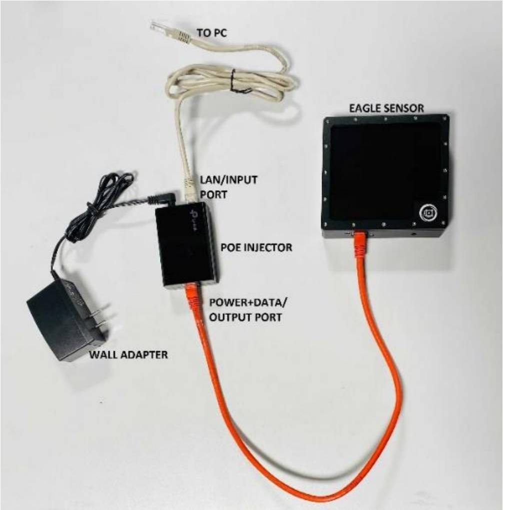
- Level 1-5 autonomous driving applications.
- Simultaneous Localization and Mapping (SLAM).
- Intelligent Transportation Systems (ITS).

3.4 What is in the Box?

Oculii will supply the following in the EAGLE box:

- Oculii EAGLE Sensor
- Power over Ethernet converter supplied from tp-link (part number: TL-POE150S)
- Cat 6 Ethernet cable, 2 foot (x1)

3.5 Connection



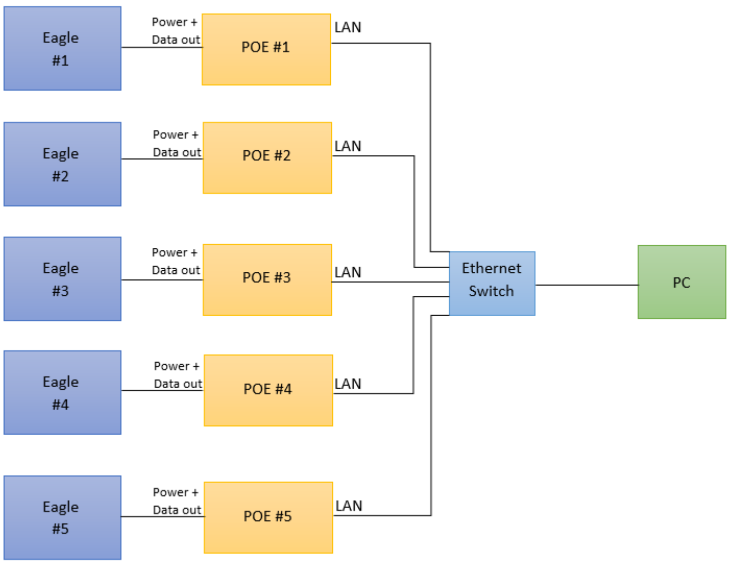
- **Power**
  - Wall Adapter Input: 100 – 240V AC
  - Wall Adapter Output: 48V DC 0.5A
  - Power Over Ethernet (POE) Injector Input: 48V, 0.5A
  - Power Over Ethernet (POE) Injector Output: 48V, 0.35A
  - Standard: IEEE802.3
  - Length of Ethernet cable: 0 - 100m
  - The wall adapter is powered through a 100-240V AC supply. The injector is powered through a 48V DC supply received from the wall adapter. The sensor is powered through a Cat 6 Ethernet cable connected from the 'Power + Data Out' port on the POE injector.

- I/O
    - Cat 6 Ethernet cable is used as a connection between the injector and the PC through the 'LAN IN' side. Data received from the sensor is transferred through the injector to the PC. (For settings on the PC to receive the data from the sensor please refer section 4).

- Circuit Protection
    - The POE injector used is an active POE which has an inbuilt circuit protection. If a short circuit is detected or the wrong ethernet connection is made, the injector does not supply power to the circuit. Before powering up, the active PoE switch will test and check to ensure the electrical power is compatible between the switch and the remote device. If it is not, the active PoE switch will not deliver power, preventing any potential damage to the non-PoE device. Despite the protection, take extra caution to connect the 'Power + Data' port to the sensor and the 'LAN' port to the PC.

3.6 Multiple Sensor Connection

When more than one sensor is used (n sensors) in the installation the connection can be made as shown in the figure below which shows the connection for n=5. Instead of connecting the 'LAN' port of the sensor to the PC, it can be connected to a switch, with one ethernet from the switch connected to the PC. Each ethernet is capable of supplying 100Mbits of data, so the switch should be capable of supporting up to n*100Mbits of data on each port (since the sum of data is sent to the PC through 1 port). The PC ethernet port should also be able to support n*100Mbits of data.



4. I/O CONFIGURATION

4.1 IP Address and UDP Port

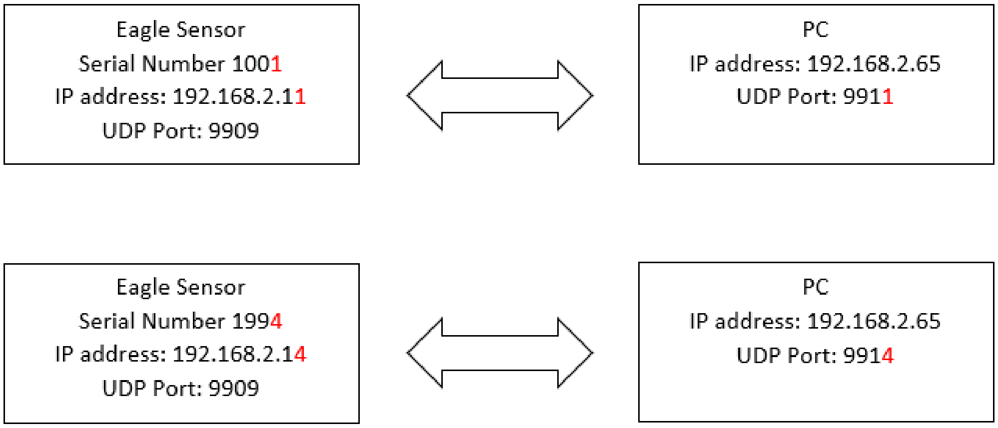
The sensor IP address and UDP port number to **receive data** are linked to the serial number of the EAGLE sensor that you receive. To retrieve your IP address please look at the back of your case and make a note of the unit digit. We use 10 different IP addresses and UDP port numbers.

Unit Digit of Serial Number	IP Address	UDP Port Number
0	192.168.2.10	9910
1	192.168.2.11	9911
2	192.168.2.12	9912
3	192.168.2.13	9913
4	192.168.2.14	9914
5	192.168.2.15	9915
6	192.168.2.16	9916
7	192.168.2.17	9917
8	192.168.2.18	9918
9	192.168.2.19	9919

In the above example the sensor serial number is 1016. The **unit digit** is 6. Thus, the IP address will be 192.168.2.16 and the UDP port number will be 9916.

To send commands (refer to section 'Radar Commands') to the EAGLE Sensor you will need to send from **the IP address 192.168.2.65** and sent to the **UDP port 9909**.

4.2 I/O Block Diagram



The block shows the IP address and UDP port configurations for the Sensor with serial number ending with **'1'** and serial number ending with **'4'**. The Static IP of the PC and the UDP port to receive commands at the sensor are fixed at 192.168.2.65 and 9909, respectively. The IP address and UDP port over which data is sent is configurable based on the unit digit of the serial number as can be seen above.

4.3 Static IP

To receive data on your PC please set the Static IP address to 192.168.2.65

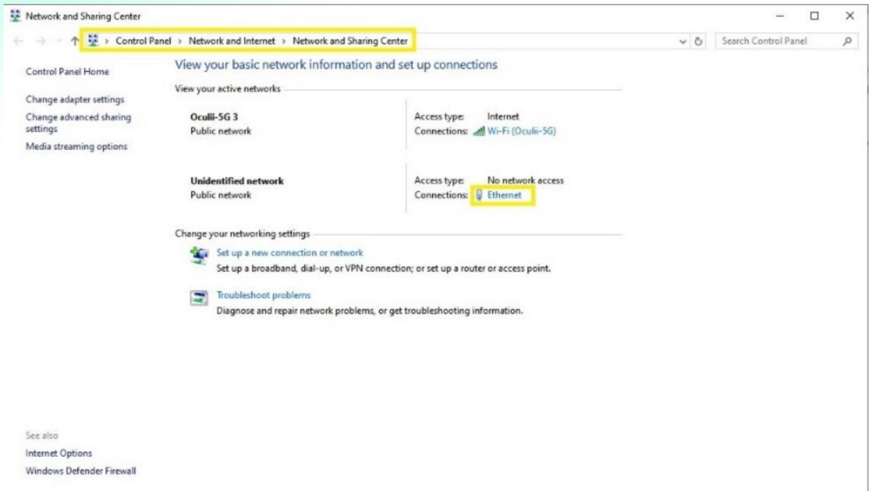
It is suggested to keep the PC off of any VPN connection which is known to interfere with the sensor communication.

To set the Static IP follow the steps listed below:

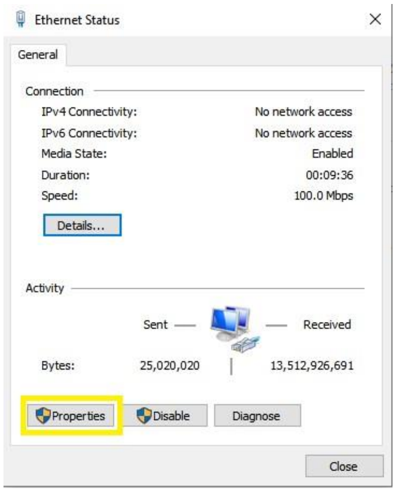
i) Make sure your sensor is powered and connected to the Ethernet port of your PC.

ii) Visit the 'Network and Sharing Center'.

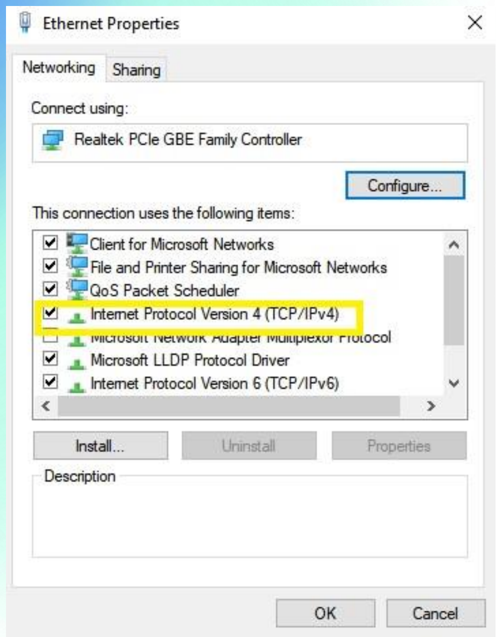
iii) Double tap on 'Ethernet'



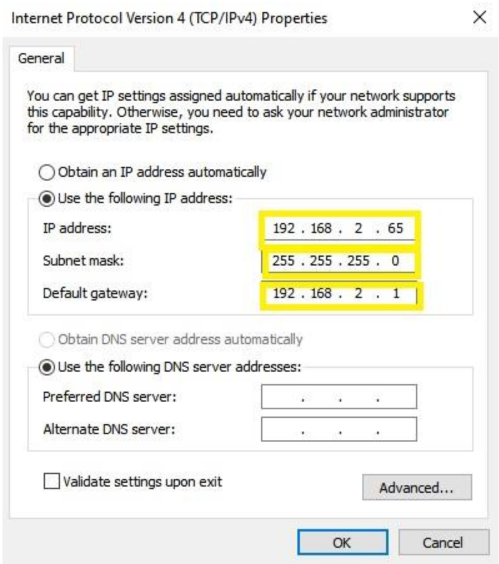
iv) Go to 'Properties'



v) Scroll to find 'Internet Protocol Version 4' and double tap.



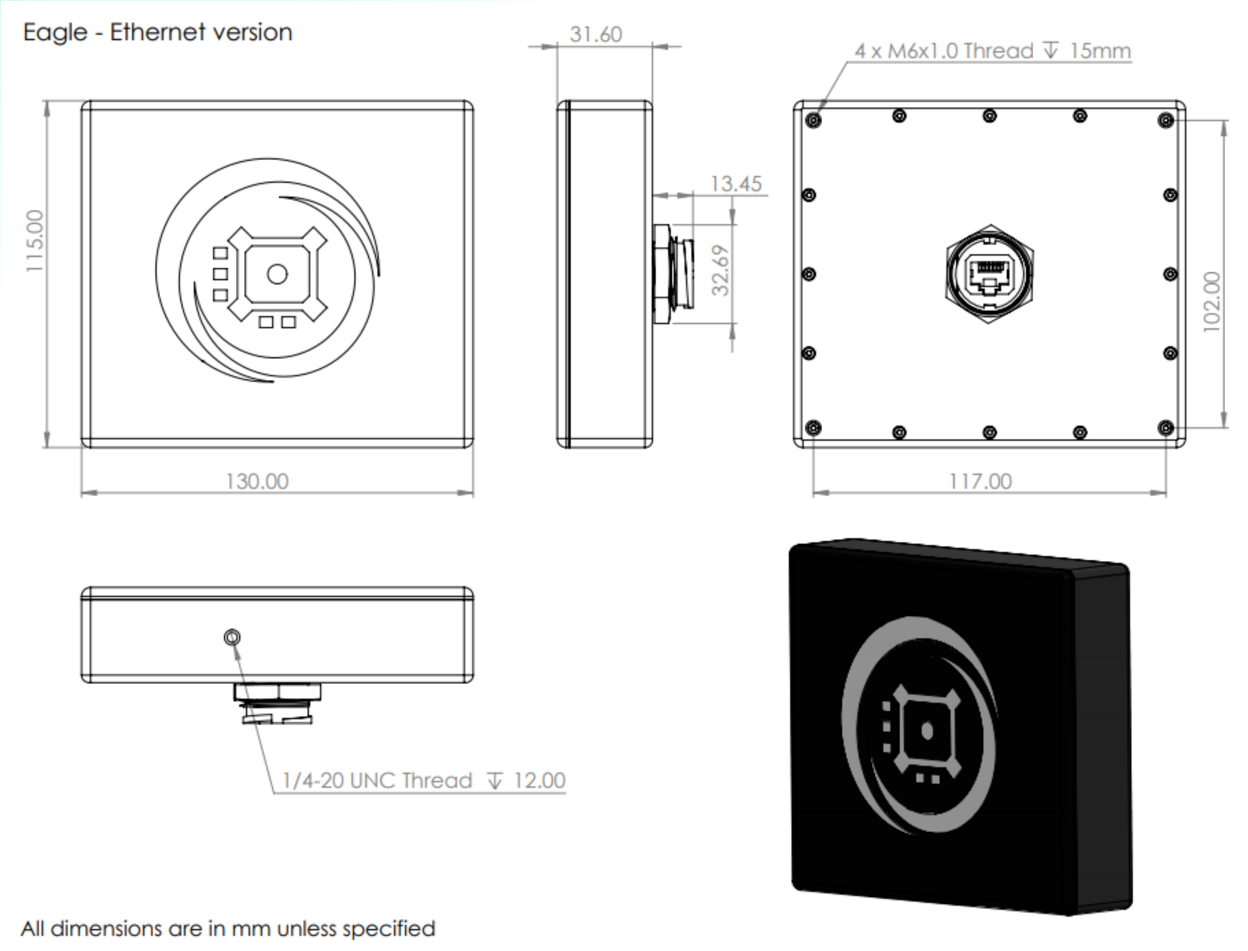
vi) Set the IP address to 192.168.2.65, the Subnet to 255.255.255.0 and Gateway to 192.168.2.1



5. MECHANICAL DIMENSIONS & MOUNTING

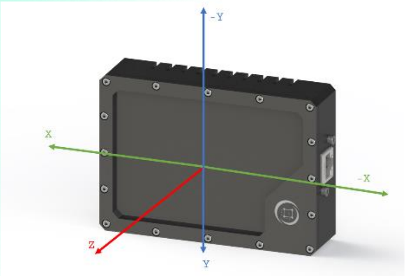
5.1 Mechanical Dimension

EAGLE – Dimensions



5.2 Sensor Coordinate System

The sensor coordinate system is defined in the image below. The sensor orientation is with a quarter inch screw hole in the bottom and the ethernet connector on the right-hand side of the sensor (seen from front). All the angular measurements input to the system should be between 0° and 360°

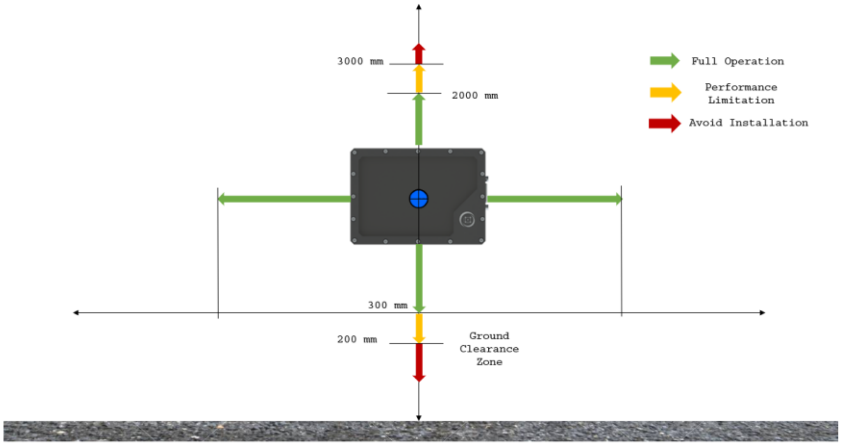


5.3 Location Mounting Example

The location mounting example includes a description for the horizontal and vertical position of the sensor location.

i) At 0-degree vertical angular rotation (no upward or downward tilt)

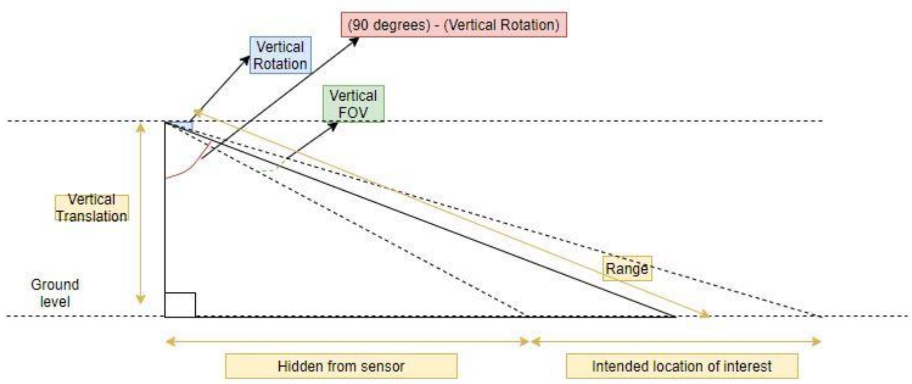
Horizontal Translation: Restriction is dependent on length of Ethernet POE. Under 100m. Note that the sensor data will be with respect to the sensor and not the vehicle center.
Horizontal Rotation: Should be rotated less than 30 degrees in either direction from the intended location of maximum power.
Vertical Translation: 0.3m - 3.0m above ground level.
Vertical Rotation: 0 degrees.



In the diagram above the mounting tolerance of the sensor is provided. With no vertical rotation, the sensor can be placed from 0.3m to 3.0m above the ground and has full operation in the horizontal plane. The sensor can also be rotated on the horizontal plane but should be within 30 degrees of intended location of interest to ensure that maximum power is directed towards to region of interest.

ii) With angular rotation (with an upwards or downward tilt)

Horizontal Translation: Restriction is dependent on length of Ethernet POE. Under 100m. Note that the sensor data will be with respect to the sensor and not the vehicle center.
Horizontal Rotation: Should be rotated less than 30 degrees in either direction from the intended location of maximum power.
Vertical Translation: Maximum installation height can be up to 50m (recommended 0 to 15m).
Vertical Rotation: Rotation should depend on the intended location of maximum power and should be calculated with respect to the height the sensor is positioned at.



In the figure above the sensor is placed at a height and is tilted down in order to see a specified region (an over bridge traffic monitoring example). Similarly, by adjusting the height and the tilt angle, the EAGLE could be oriented for different applications requiring different intended regions of interest (examples include a small tilt upwards from the bumper to eliminate ground reflection, or a small tilt down when mounted on top of trucks to see objects close to the bus) . Note that the data received is with respect to the center of the sensor. Any physical adjustments must be accounted on the user/customers end.

5.4 Secondary Surface Mounting (RADOME)

The sensor can be mounted with a surface between the radar and the outside environment (for example – vehicle bumper). This enclosure is called a Radar Dome (RADOME). Configuration of the RADOME for optimal performance depends on several factors listed below.

i) Material type
ii) Thickness
iii) Distance
iv) Tilt/Curvature of surface

v) Paint

Following are short descriptions on how these factors impact the performance of the sensor and how they can be configured for optimal performance.
Note: The radar is equipped with a front plate so a secondary surface mount for initial testing and validation is not recommended as performance degrades with addition of any secondary surface.

i) Material Type

Material properties should only be considered at the 77GHz frequency band and should have the following properties:

- Low di-electric constant (for low surface reflection)
- Low di-electric loss (for reduced transition damping of signal)
- Roughness less than lambda/10 (~400um)

Below is a table of commonly used secondary surface materials

Material	Di-electric constant (ε_r) at 77Ghz
Polypropylene	2.35
Polyamide	2.75
Polycarbonate	2.80
PC-PBT (Polycarbonate Type)	2.90
ABS (Acrylnitril-Butadien-Styrol)	3.12
PMMA (Poly Methyl Methacrylate)	3.40
ASA (Acrylonitrile Styrene Acrylate)	3.80

Note: Di-electric constant (permittivity) of materials differ among manufacturers. Ensure that the material properties are quoted by all manufacturers at 77GHz bandwidth.

ii) Material Thickness

For optimal permittivity, the thickness of the material needs to be calculated. The material thickness can be any integer multiple of the optimal thickness (n*optimal thickness). A higher integer increases the attenuation, thus smaller thickness is recommended.
Using the following 3 formulas we can calculate the thickness of the surface:

A. $\lambda_o = c_0 / f_c$
B. $\lambda_m = \lambda_o / (\text{sqrt } \varepsilon_r)$
C. $T_m = \lambda_m / 2$

λ_0 : Wavelength, c_0 : Speed of light ($3 * 10^8$), f_c = 77GHz, ε_r : Relative permittivity in material, λ_m : Wavelength in material, T_m : Optimal Thickness of Material
In the table below we calculate the thickness for the materials mentioned before:

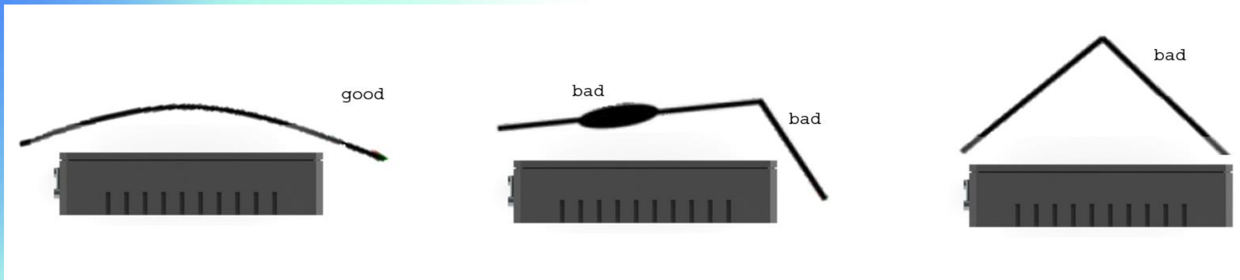
Material	Di-electric constant (ε_r) at 77Ghz	Optimal Thickness of Material (mm)
Polypropylene	2.35	1.27
Polyamide	2.75	1.18
Polycarbonate	2.80	1.17
PC-PBT (Polycarbonate Type)	2.90	1.15
ABS (Acrylnitril-Butadien-Styrol)	3.12	1.10
PMMA (Poly Methyl Methacrylate)	3.40	1.06
ASA (Acrylonitrile Styrene Acrylate)	3.80	1.00

iii) Distance

- The distance between the radar sensor and the RADOME needs to be an integer multiple of lambda/2 which is ~1.95mm, thus n*1.95mm is acceptable.
- The distance should be large enough to avoid mechanical vibrations, mechanical interference, and mechanical stress.
- The distance should be enough to avoid thin films of water, snow, or mud from developing on the sensor front plate.

iv) Tilt/Curvature of the surface

The angle between the sensor and the secondary surface is the tilt angle. The tilt angle should always be between 0 to 30 degrees. Note that smaller tilt angles could introduce multipath, raise noise levels, or introduce false targets. Higher tilt angles increase the effective thickness and thus the damping effect of the material increases.
To prevent distortion of the signal, the secondary surface needs to be smooth with roughness less than lambda/10 (~400um). The curvature should be uniform and avoid sharp edges or abrupt thickness changes as shown in the image below.



v) Paint

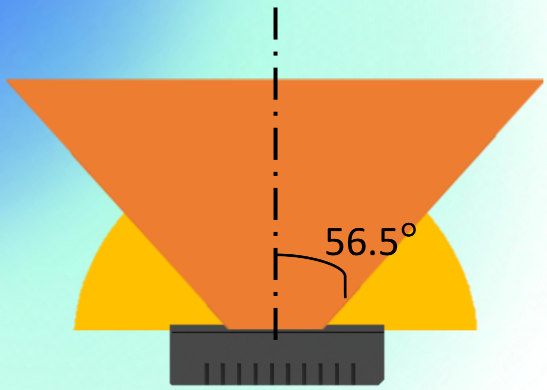
For layers of paint, varnish, primer to be introduced on the secondary surface a study on the degradation of the signal would need to be performed. These layers would influence the performance of the sensor and need to be taken into consideration when choosing a secondary surface. Up to 4 layers of paint could be allowed if attenuation is kept within spec in the entire system.
Further, for metallic paint, more considerations will need to take place. The percentage of metal in the paint, the size of the metal particles in the paint and the thickness of the different layers of paint can all influence the attenuation of the radar and will need to be studied and kept within spec of the radar.

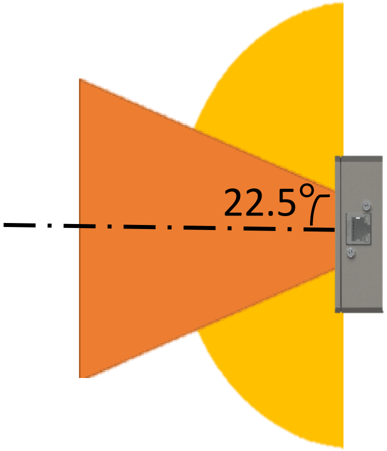
The max attenuation acceptable for the Oculii EAGLE to maintain its performance is described below:

- One-way attenuation shall not exceed 2dB
- Max reflectivity co-efficient shall not exceed -2.2dB.

5.5 Sensor Radiation Cone

The sensor radiation cone is shown in the following diagrams. The sensor radiation cone shall be kept free from metallic objects, or any objects that can interfere with the radar performance. The horizontal field of view for the sensor is 120 degrees and the vertical field of view is 40 degrees, both symmetric to the center of the sensor.



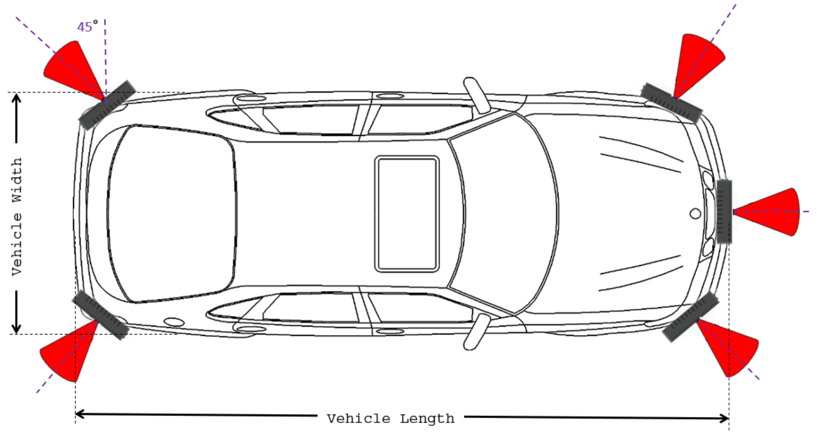


5.6 Sensor Operation Temperature

The sensor is designed to operate in the temperature range between - 40 to +105 Celsius. The metallic body of the sensor is the main medium of heat transfer. To ensure proper heat transfer the sensor should be exposed to cool free flowing air and should be in direct contact with thermal heat transferring materials.

5.7 Multiple sensors mounting

When mounting multiple sensors note that each sensor behaves like an individual unit with the data output with respect to the center of the sensor. Multiple orientations are possible if the above consideration is followed. Note the rotation and translation of the physical mounted sensors and adjust the radar data to the center of the vehicle as required. The figure below shows one installation scenario. In this case the sensors are installed at the four corners of the vehicle at 45 degrees with respect to the direction of motion of the vehicle. A fifth sensor is installed at the center facing the direction of motion. More information on how to measure the rotation and translation of the sensors can be found under section 8.2.



5.8 Manual Alignment

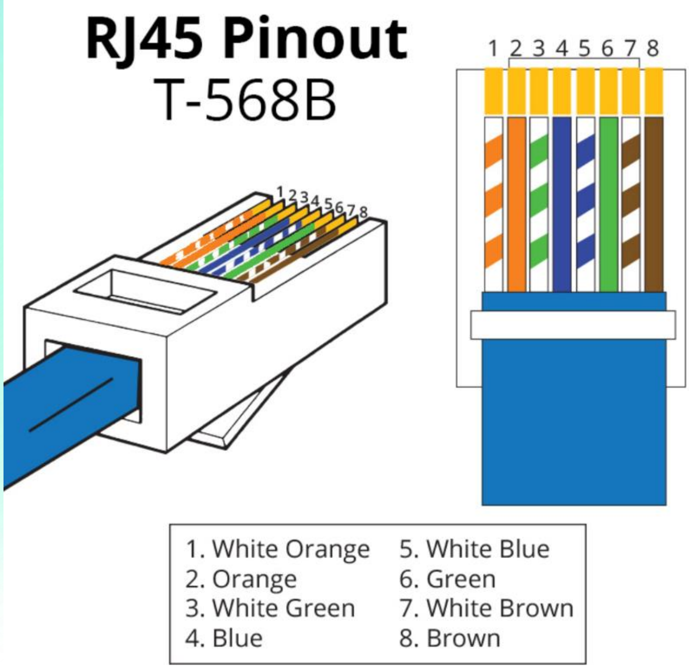
The EAGLE sensor needs to be manually adjusted along all 3 planes. Misalignment can be a result of many factors such as static deviations due to sensor and mechanical fixture tolerance, and dynamic deviations in loading and suspension. Variations in temperature and mechanical vibrations over time could also influence the calibration. The introduction of a secondary surface could also have an effect.

The user/customer shall determine misalignment via simulation or measurement. The user could validate the alignment with a simple test as described below.

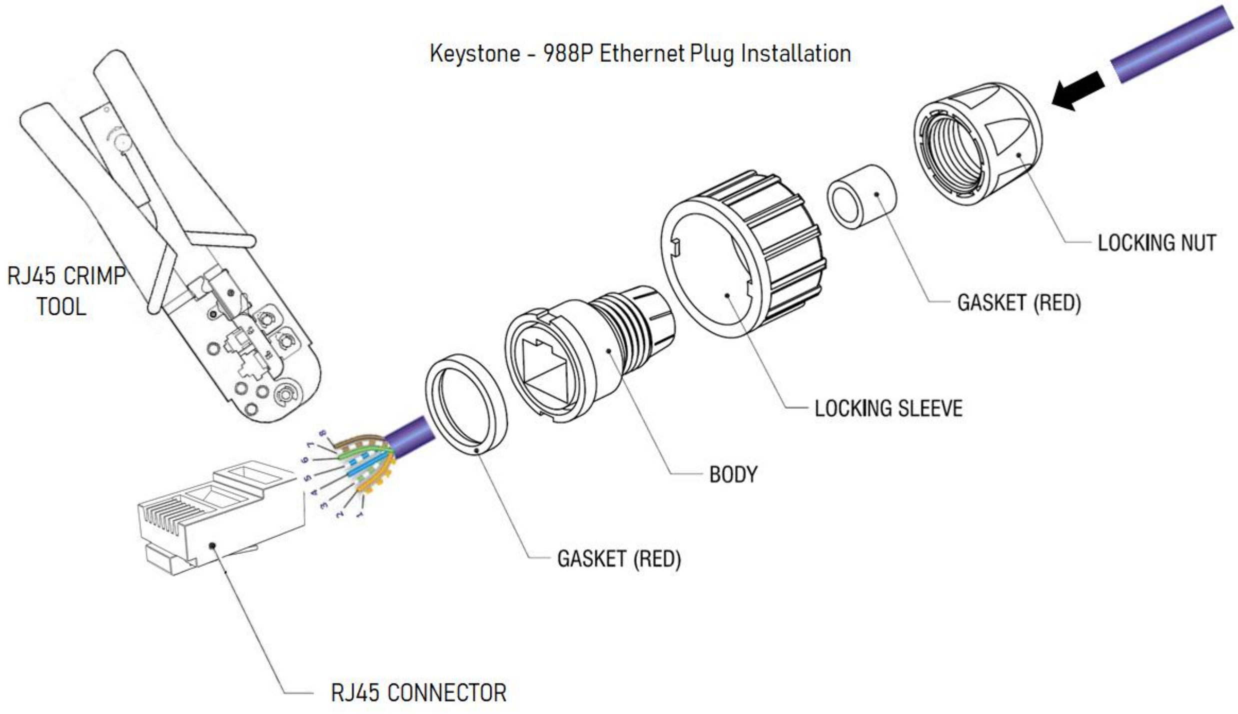
After mounting the sensor place a corner reflector at 0 degrees in front of the sensor at 4.0m distance and same height from the ground. Measure the physical range between the center of the sensor and the center of the corner reflector. Look at the radar data and ensure that the z value equals 4m and the x and y values equal 0m.

5.9 Waterproof Ethernet Plug

The EAGLE sensor would include a waterproof ethernet plug (Keystone 988P) inside the box, please ensure to install this as this is a critical point in waterproofing the device. We recommend using a Cat 5e or Cat6 ethernet patch cable and installing the ethernet plug by cutting off the head of the patch cable and to crimp a RJ45 connector.



Please ensure the cable crimping pin order to match the T-568B (Straight through - Patch cable) and use a RJ45 Crimp tool to crimp the cable as shown in the installation diagram below.



6. RADAR INTERFACE AND DECODE

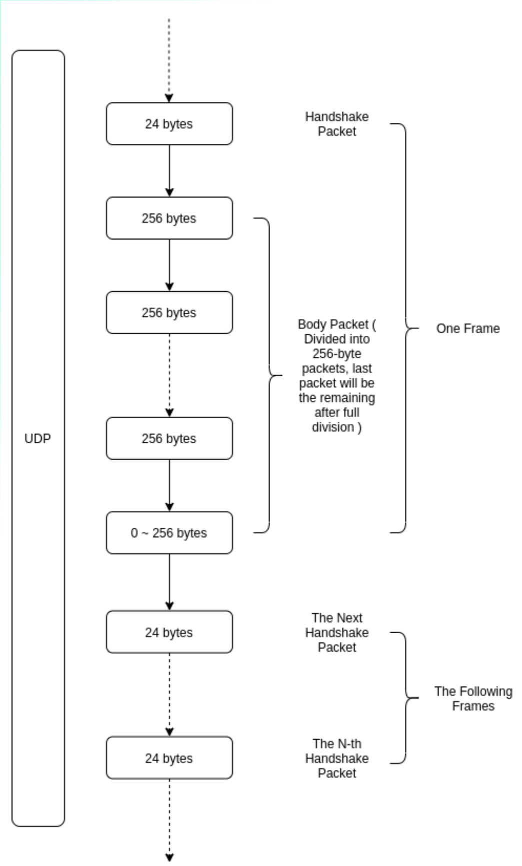
This section briefs the protocol used by Oculii for the transmission of detection and tracker data frames from the Oculii Radar Sensor.

The EAGLE radar sensor outputs the output list (Header + Detections + Tracks + Footer) at the configured frame rate. The output list is comprised of the Header, Detection information and Track information.

The Header is 48 Bytes and contains the Frame Number, Version Number, Number of Detection, Number of Tracks, Host Speed, Host Angle, and the accuracy values for Range, Doppler, Alpha and Beta. It also contains a few reserved fields for future use.

Each detection information is packaged into 8 Bytes and each track is packaged into 32 Bytes. The footer is 32 bytes. The number of detection field and the number of tracks field in the header provide the value for decoding detection and tracks, respectively.

6.1 Radar Output Communication Flow



Handshake Structure:

Magic 8	Frame Data Length (Dynamic) 4	Reserved 12

Magic : 8 bytes, [1 9 8 9 1 0 2 2]
Frame Data Length (Dynamic) : 4 bytes, uint_32, frame size
Reserved : 12 bytes

Frame Data Length is in the little-endian format and is the length of the packet structure. The data is dynamic and thus the length needs to be read in between frames. Since the data packet is truncated in chunks of 256 bytes, a calculation would need to be done to determine the number of 256 chunks as well as the length of the last chunk of data. For example, if the length is 1000, the data will come as 3 chunks of 256 bytes and 1 chunk of 232 bytes.

6.2 Output List: Header + Detection + Tracker

Output List: Header + Detection + Tracker + Footer

Header Block (48 bytes)						Detection Block (8 * N)				Tracker Block (32 * Nt)				Footer Block (32 Bytes)	
Magic	Frame Num	Version Num	Num Det	Num Tracks	Decode/ Parse Info	Det 1	Det 2	...	Det N	Track 1	Track 2	...	Track Nt	CRC	Reserved
8B	4B	4B	2B	2B	28B	8B	8B		8B	32B	32B		32B	4B	28B

Header: 384 bits (48 Bytes)
Detection : 64 bits (8 Bytes) * N
Tracks : 256 bits (32 Bytes) * Nt
Footer : (32 Bytes)

6.3 Header and Footer Structure

Header Block (48 bytes)

<table>
<tr><td colspan="8">Magic
(8 bytes : 0x21436587)</td></tr>
<tr><td colspan="4">Frame Number
(4 bytes) (uint_32)</td><td colspan="4">Version Number
(4 bytes) (uint_32)</td></tr>
<tr><td colspan="2">Number of Detection
(2 bytes) (uint_16)</td><td colspan="2">Number of Tracks
(2 bytes) (uint_16)</td><td colspan="2">Host Speed
(2 bytes) (int_16)
(Divide by 100 to get value in meters/second)</td><td colspan="2">Host Angle
(2 bytes) (int_16)
(Divide by 100 to get value in degrees)</td></tr>
<tr><td colspan="2">Reserved
(2 bytes)</td><td colspan="2">Reserved
(2 bytes)</td><td colspan="2">Reserved
(2 bytes)</td><td colspan="2">Reserved
(2 bytes)</td></tr>
<tr><td colspan="2">Range Accuracy Idx 0
(2 bytes) (uint_16)
(Divide by 10000 to get value in meters)</td><td colspan="2">Doppler Accuracy Idx 0
(2 bytes) (uint_16)
(Divide by 10000 to get value in meters/second)</td><td colspan="2">Azimuth Accuracy Idx 0
(2 bytes) (uint_16)
(Divide by 10000 to get value in degrees)</td><td colspan="2">Elevation Accuracy Idx 0
(2 bytes) (uint_16)
(Divide by 10000 to get value in degrees)</td></tr>
<tr><td>DSP workload (%)</td><td>ARM workload (%)</td><td>Byte 43</td><td>Byte 44</td><td>Byte 45</td><td>Byte 46</td><td>Byte 47</td><td>Byte 48</td></tr>
</table>

Magic Header : 8 Bytes, [2 1 4 3 6 5 8 7]
Frame Number : 4 Bytes, uint32_t, frame number
Version number : 4 Bytes, uint32_t, Version number, uint32_t format: MMddhhmm
NumDetections : 2 Bytes, uint16_t, Number of Detection in the frame
NumTracks : 2 Bytes, uint16_t, Number of Tracks in the frame
Host Speed : 2 Bytes, int16_t, divide by 100 to get ego speed in m/s
(e.g. 6453 = 64.53 m/s, e.g. -2456 = -24.56 m/s)
Host Angle : 2 Bytes, int16_t, divide by 100 to get host angle in degree. Clockwise rotation corresponds to positive rotation.
(e.g. 1400 = 14 deg)
Reserved*4 : 8 Bytes, currently reserved as 4 uint, data type could change.
Range Accuracy Idx0 : 2 Bytes, uint16_t, divide by 10000 to get value in m
Doppler Accuracy Idx0 : 2 Bytes, uint16_t, divide by 10000 to get value in m/s
Azimuth Accuracy Idx0 : 2 Bytes, uint16_t, divide by 10000 to get value in degree
Elevation Accuracy Idx0 : 2 Bytes, uint16_t, divide by 10000 to get value in degree
DSP workload : 1 Byte, double
ARM workload : 1 Byte, double
Reserved : 6 Bytes, variable data type.

Footer Block (32 bytes)

Reserved (4 Bytes)				Reserved (4 bytes)			
Range Accuracy Idx 1 (2 bytes) (uint_16) (Divide by 10000 to get value in meters)		Doppler Accuracy Idx 1 (2 bytes) (uint_16) (Divide by 10000 to get value in meters/second)		Azimuth Accuracy Idx 1 (2 bytes) (uint_16) (Divide by 10000 to get value in degrees)		Elevation Accuracy Idx 1 (2 bytes) (uint_16) (Divide by 10000 to get value in degrees)	
Byte 17	Byte 18	Byte 19	Byte 20	Byte 21	Byte 22	Byte 23	Byte 24
Byte 25	Byte 26	Byte 27	Byte 28	Byte 29	Byte 30	Byte 31	Byte 32

Range Accuracy Idx1 : 2 Bytes, uint16_t, divide by 10000 to get value in m
Doppler Accuracy Idx1: 2 Bytes, uint16_t, divide by 10000 to get value in m/s
Azimuth Accuracy Idx1 : 2 Bytes, uint16_t, divide by 10000 to get value in degree
Elevation Accuracy Idx1 : 2 Bytes, uint16_t, divide by 10000 to get value in degree

6.4 Detection Structure:

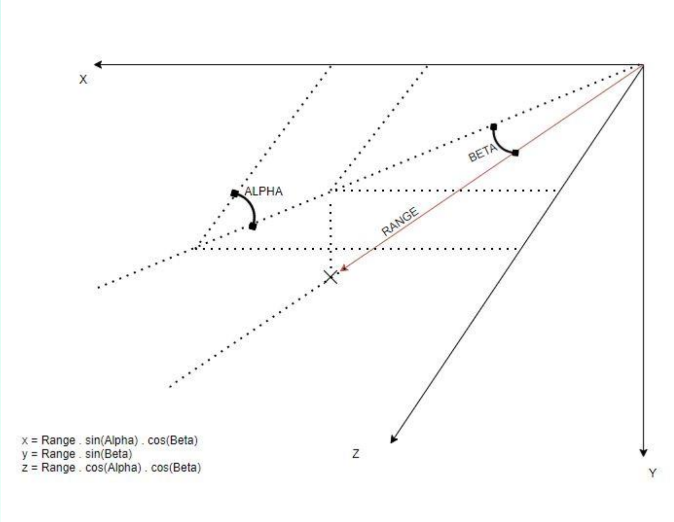
<table>
<tr><td>Reserved
(1 bit)</td><td>Flag
(1 bit)</td><td>Reserved
(6 bit)</td><td colspan="3">Power Value
(16 bits)
(0 to 65535)</td><td>Beta Index
(10 bits)
(-511 to 512)</td></tr>
<tr><td colspan="2">Beta cont..</td><td colspan="2">Azimuth Index
(10 bits)
(-511 to 512)</td><td>Doppler Index
(10 bits)
(-511 to 512)</td><td colspan="2">Range Index
(10 bits)
(0 to 1023)</td></tr>
</table>

- Range Index: 10 bits, multiply with Range Accuracy value to get Range value in m.
- Doppler Index: 10 bits, multiply with Doppler Accuracy value to get Doppler value in m/s.
- Azimuth Index: 10 bits, multiply with Azimuth Accuracy value to get Azimuth (alpha) value in degree.
- Beta Index: 10 bits, multiply with Elevation Accuracy value to get Elevation (beta) value in degree.
- Power: 16 bits, 2 bytes, uint16_t, 0 to 65535, Power value is in dB scale and represents the signal to noise ratio of the detection. Divide by 100 to decode power. (e.g. 208 = 2.08dB)
- Flag: 1 bit, used for decoding. Source code provided in the SDK.

  If bit is 0: Use Index 0 accuracy values from the header to obtain range, doppler, alpha and beta.
  If bit is 1: Use Index 1 accuracy values from the footer to obtain range, doppler, alpha and beta.

- Reserved*: 7 bits.
- XYZ: The XYZ coordinates can be retrieved using the following equation:
  X = Range * Sin(Alpha) * Cos(Beta)
  Y = Range * Sin(Beta)
  Z = Range * Cos(Alpha) * Cos(Beta)

Below is a diagram showing the same.



6.5 Tracker Structure:

Each Track (32 Bytes, 256 bits)							
Track ID (4 bytes)		XPos (2 bytes)	YPos (2 bytes)	ZPos (2 bytes)	XDot (2 bytes)	YDot (2 bytes)	ZDot (2 bytes)
Res 1 (2 bytes)	Res 2 (2 bytes)	Res 3 (2 bytes)	Flag (2 bytes)	Class (2 bytes)	Conf. (2 bytes)	Res 4 (2 bytes)	Res 5 (2 bytes)

- Track ID: 32 bits, 4 bytes, uint32_t, 0 to 4294967296 Track ID
- Track XPos: 16 bits, 2 bytes, int16_t, -32768 to 32767, divide by 100 to get x in m (e.g. 6453 = 64.53 m, e.g.: -2456 = -24.56 m)
- Track YPos: 16 bits, 2 bytes, int16_t, -32768 to 32767, divide by 100 to get y in m (e.g. 6453 = 64.53 m, e.g. -2456 = -24.56 m)
- Track ZPos; 16 bits, 2 bytes, uint16_t, 0 to 65535, divide by 100 to get z in m (e.g. 12654 = 126.54 m)
- Track XDot: 16 bits, 2 bytes, int16_t, -32768 to 32767, divide by 100 to get speed in the x direction in m/s (e.g. 6453 = 64.53 m/s, e.g. -2456 = -24.56 m/s)
- Track YDot: 16 bits, 2 bytes, int16_t, -32768 to 32767, divide by 100 to get speed in the y direction in m/s (e.g. 6453 = 64.53 m/s, e.g. -2456 = -24.56 m/s)

- **Track ZDot:** 16 bits, 2 bytes, int16_t, -32768 to 32767, divide by 100 to get speed in the z direction in m/s (e.g. 6453 = 64.53 m/s, e.g. -2456 = -24.56 m/s)

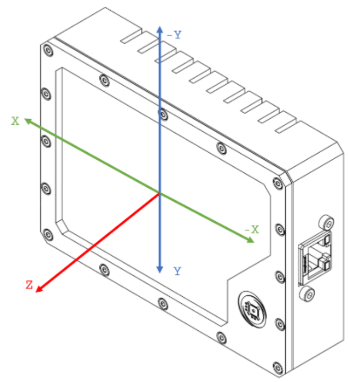
- **Reserved*5** : 2 Bytes, currently reserved as uint16_t, data type could change.

- **Track Flag:** 16 bits, 2 bytes, uint16_t, 0 to 65535. 16 1-bit flags, the flag definitions are internal to Oculii.
  Bit 3 – Bit 15 : Reserved
  Bit 0, Bit 1, Bit 2 : Track Quality. Currently only values 1 and 2 are used. Filter to use only value '2'.
- **Track Class:** 16 bits, 2 bytes, unit16_t, 0 to 5. Reserved for future.
  - 0: Unknown Class
  - 1: Pedestrian
  - 2: Motorcycle/Bike
  - 3: Vehicle and SUV
  - 4: Bus and Truck
  - 5: Background

- **Track Class Confidence Score:** 16 bits, 2 bytes, unit16_t, 80-99, assigns a confidence score for the classification made in Track Class – higher score corresponds to higher confidence

6.6: Radar Coordinate System

Below is a diagram showing the radar coordinate system used in the EAGLE sensors.



7. RADAR CONFIGURATION MODES AND COMMANDS

7.1 Synchronous Trigger Command – PTP Time Synchronization

To send a trigger command that maintains synchronization the following command should be input.

Command	Read/ Write	Length	CRC	Payload
0X0A	0	0x1E	0	Byte 7 – Byte 30

Byte 7: Wait on PTP Sync (0 – No, 1- Yes)
Byte 8: If byte 7 is 0 Byte 8 value does not matter. If byte 7 is 1 Byte 8 determines PTP mode. (0 – unicast, 1- multicast)
Byte 9-12: If byte 7 is 0 or byte 8 is 1. Otherwise, IP Address PTP Master (IPV4, delimiter “:”). Example IP 192.168.2.10 will be {0xC0, 0xA8, 0x02, 0x0A}
Byte 13-16: Hard Coded to match frame period of the configuration. Period in microseconds (uint32) {100000 = 0x000186A0 = 100ms = 10Hz}
Byte 17-20: Delay in microseconds (uint32) {10000 = 0x00002710 = 10ms}
Byte 21-30: Reserved, all 0x00.

Note: If you do not wait for PTP (byte 7) the sensor will internally trigger itself to match the time-period set in the front-end configuration.

Note: The cycle time of PTP synchronization needs to match the period configured in the Firmware by Oculii. For example if the Firmware runs at 10Hz, PTP synchronization (byte 13-16) of 100ms is required.

Note: If there is an intended use for PTP, please contact Oculii for additional documentation.

7.2 Radar Command HostSetupCfg

HostSetupCfg command is used to set the mounting angle of the sensor at the start of data collection. Once the Start Receive button on the visualizer (Section 8) is clicked to start the data collection process, the visualizer will prompt the user to input the sensor mounting angle. If the user is not using Oculii provided applications and SDK, this command is required at the boot-up sequence at every instance of initialization.

Following are the setup of the HostSetupCfg command:

1. Host Setup Config: 'HostSetupCfg <mounting_angle> 3 1 1\r'
    - For example, if the sensor is mounted at 45 degrees clockwise and the user in inputting the host information this command is: 'HostSetupCfg 45 3 1 1\r'.

Note: Uppercase and lowercase letters have different ascii bytes assigned to them
Note: The space (ASCII: 0x20) needs to be input in the command if required.
Note: The mounting angle should be between 0 and 360 degrees (clockwise).

To send Radar Commands the user must follow the following data transfer protocol.

The command includes a 6 byte header and the payload.
Header:

1. Byte 0: Command
    - 0x02: Radar Command Send
    - 0x03: Acknowledgement
    - 0xCD: Process Report Complete
2. Byte 1: Read/Write
    - 0x00: Sending command from PC to Sensor
    - 0x01: Receiving command from Sensor
3. Byte 2,3,4: Length of the payload in Little Endian format
4. Byte 5: Error.
    - 0x00: No error
    - 0x01: Error

Payload:
The command is sent as bytes following the ASCII string and ends with a null character '\r'.
Example of header (6 bytes) + payload and length:
HostSetupCfg 0 3 1 1 = new byte[] {0x02, 0x00, 0x15, 0x00, 0x00, 0x00, 0x48, 0x6f, 0x73, 0x74, 0x53, 0x65, 0x74, 0x75, 0x70, 0x43, 0x66, 0x67, 0x20, 0x30, 0x20, 0x33, 0x20, 0x31, 0x20, 0x31, 0x0D };
Length of payload: 21

Example: Command for “HostSetupCfg 0 3 1 1\r” would be as follows:

Step 1: Send the command to the sensor. (Byte 0: 0x02, Byte 1: 0x00, Byte 2-4, Length of command, Byte 5: 0x00, Byte 6 onwards: Radar command ASCII ending with ‘\r’).

Command	Read/ Write	Length	CRC	Payload(last byte: ‘\r’)
2	**0**	**21**	**0**	**HostSetupCfg 0 3 1 1’0x0D’**

Step 2: Wait for the sensor to repeat the same command back to you (except byte 1 which would change from 0x00 to 0x01). Wait to read the same packet length that you sent and check byte by byte if the command returned matched the command you sent.

Command	Read/ Write	Length	CRC	Payload
2	**1**	**21**	**0**	**HostSetupCfg 0 3 1 1’0x0D’**

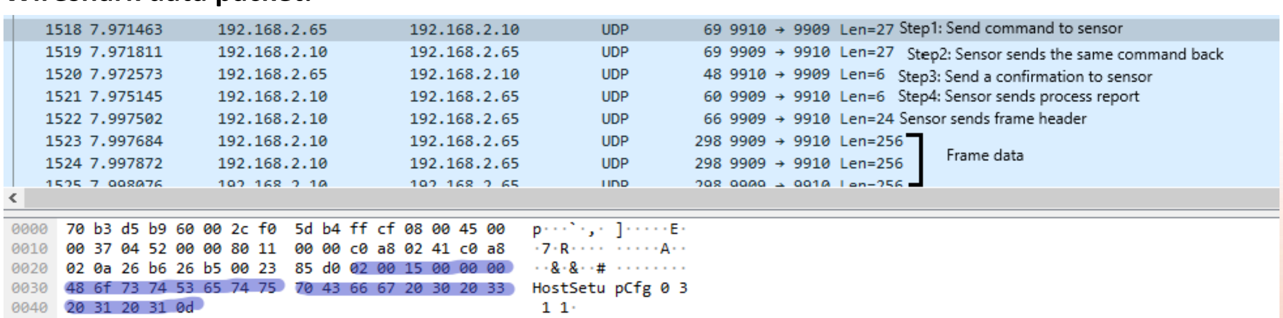
Step 3: Send the sensor a confirmation if the commands match. (Byte 0: 0x03, Byte 1-5: 0x00)

Command	Read/ Write	Length	CRC	Payload
3	**0**	**0**	**0**	**None**

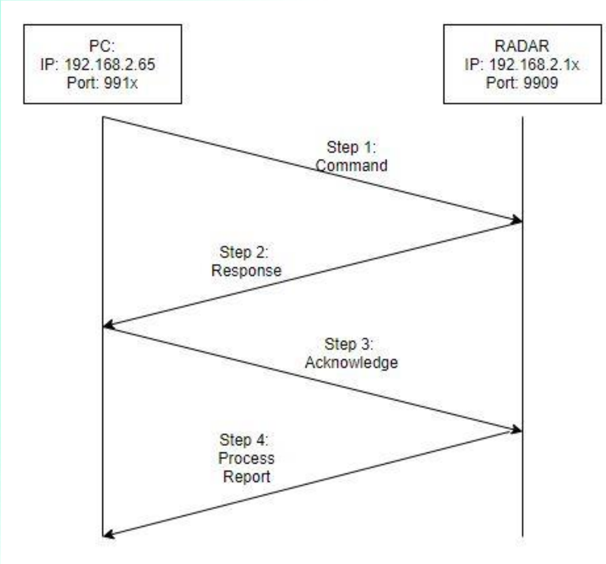
Step 4 (optional): Wait to receive the status packet from the sensor

Command	Read/ Write	Length	CRC	Payload
CD	**1**	**6**	**0**	**None**

Wireshark data packet:



The chart below explains the communication between the PC and the Radar for the steps listed above.



Note: All data is transmitted in **Little Endian** format.

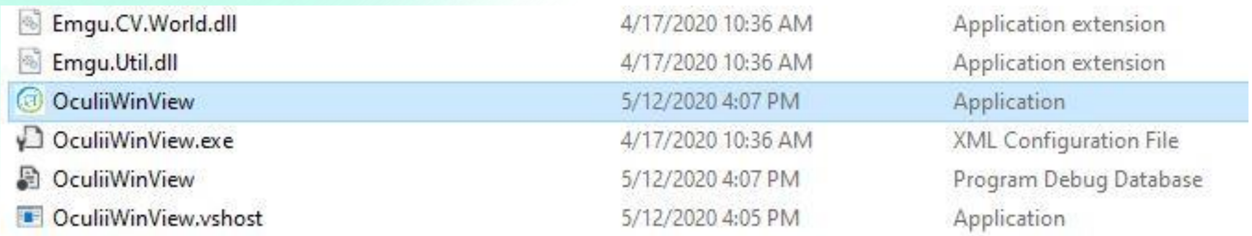
8. WINDOWS VISUALIZER

8.1 Getting Started

To run the Windows Visualizer, follow the steps below:

Step 1: Plug the Flash Drive into your computer and copy the Oculii folder to your PC's C drive.

Step 2: Follow the path C:\Oculii\Visualizer under which you will find the OculiiWinView application file. Run the application file.



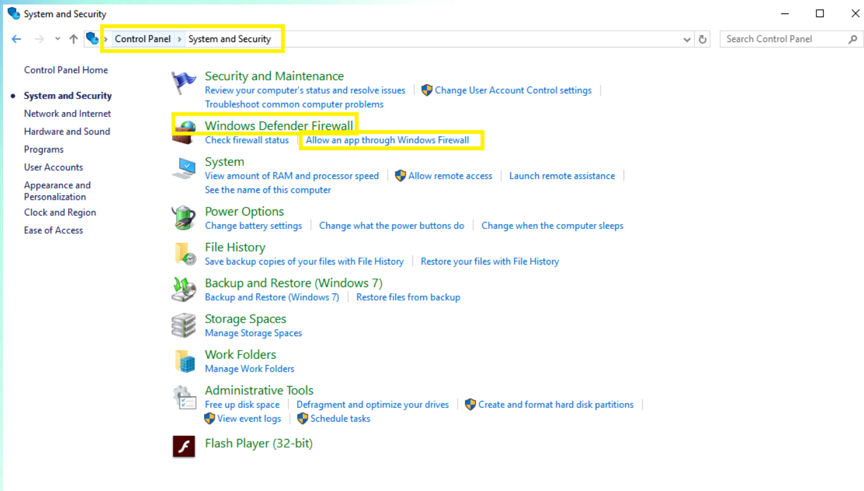
Emgu.CV.World.dll	4/17/2020 10:36 AM	Application extension
Emgu.Util.dll	4/17/2020 10:36 AM	Application extension
OculiiWinView	5/12/2020 4:07 PM	Application
OculiiWinView.exe	4/17/2020 10:36 AM	XML Configuration File
OculiiWinView	5/12/2020 4:07 PM	Program Debug Database
OculiiWinView.vshost	5/12/2020 4:05 PM	Application

Step 3: Dependency required for installation - Visual C++ Redistributable for Visual Studio 2012 which should be pre-installed in Windows, but if it is not it can be downloaded from the following link: https://www.microsoft.com/en-us/download/confirmation.aspx?id=30679

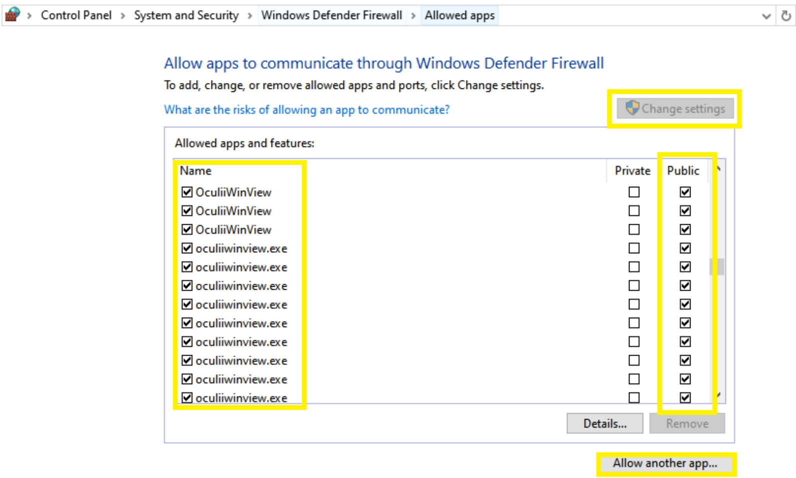
Step 4: Provide Firewall access to the application when prompted. If your PC does not prompt you to do the same, and if you cannot receive data on the visualizer follow the following steps to allow Firewall access.

Step 4i): Open Control Panel => System and Security => Windows Firewall => Allowed Programs.

Step 4ii) Click the Change Settings button.

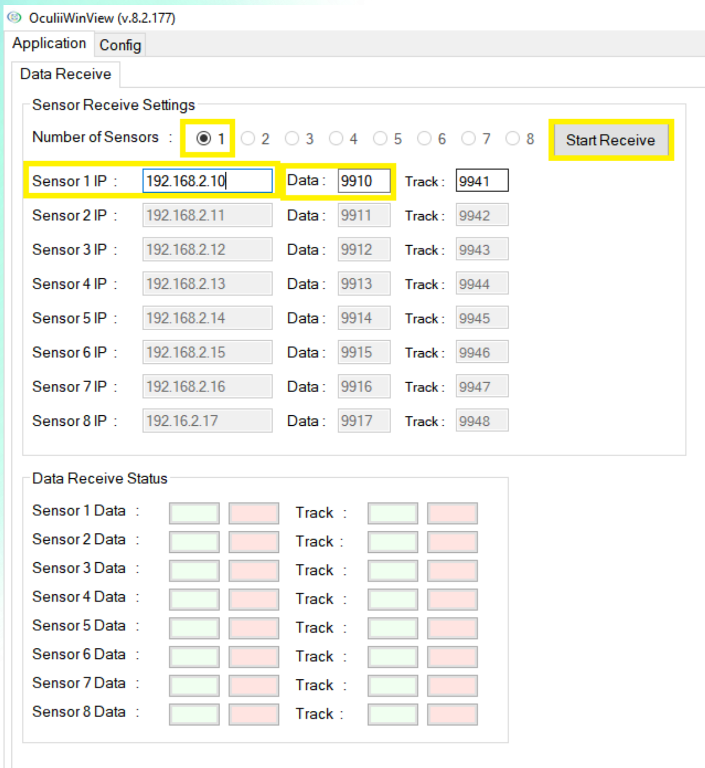


Step 4iii) Check all the Application programs OculiiWinView.exe in the list showing in the figure below.

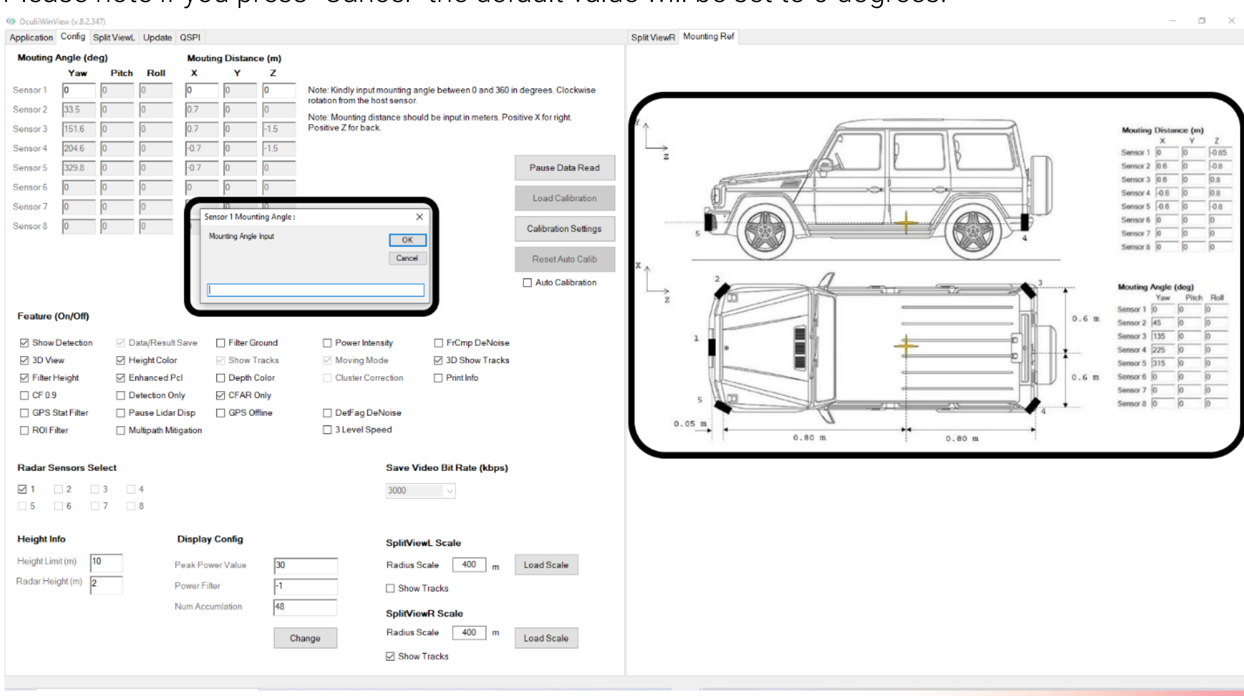


Step 5: Enter the IP address and UDP port of the sensor provided.

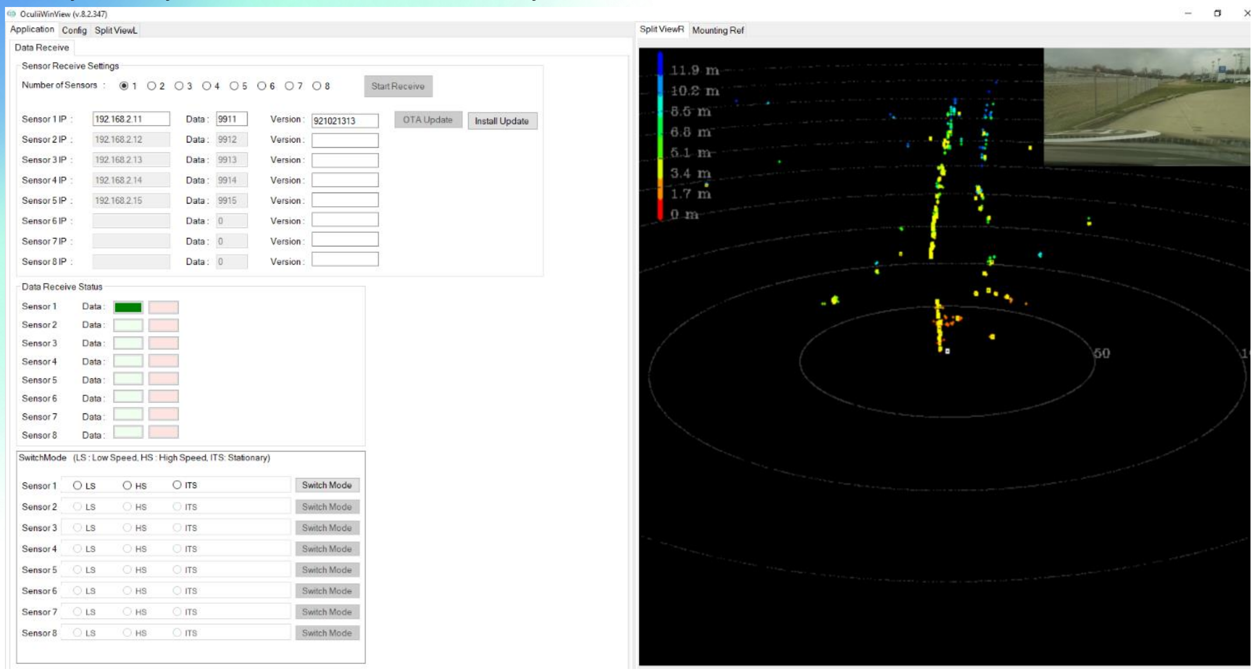
1. IP address is 192.168.2.1x,
2. UDP port is 991x
3. 'x' is the unit place digit of the sensor serial number and the port.



Step 6: Press the 'Start Receive' button after which you will be directed to sensor Mounting information pop-up window. Input the correct mounting information, followed by 'Ok' button. Please note if you press "Cancel" the default value will be set to 0 degrees.



Step 7: You should receive data which can be seen on the right-hand side of the visualizer. Please note you may need to wait 20 sec before you receive data.



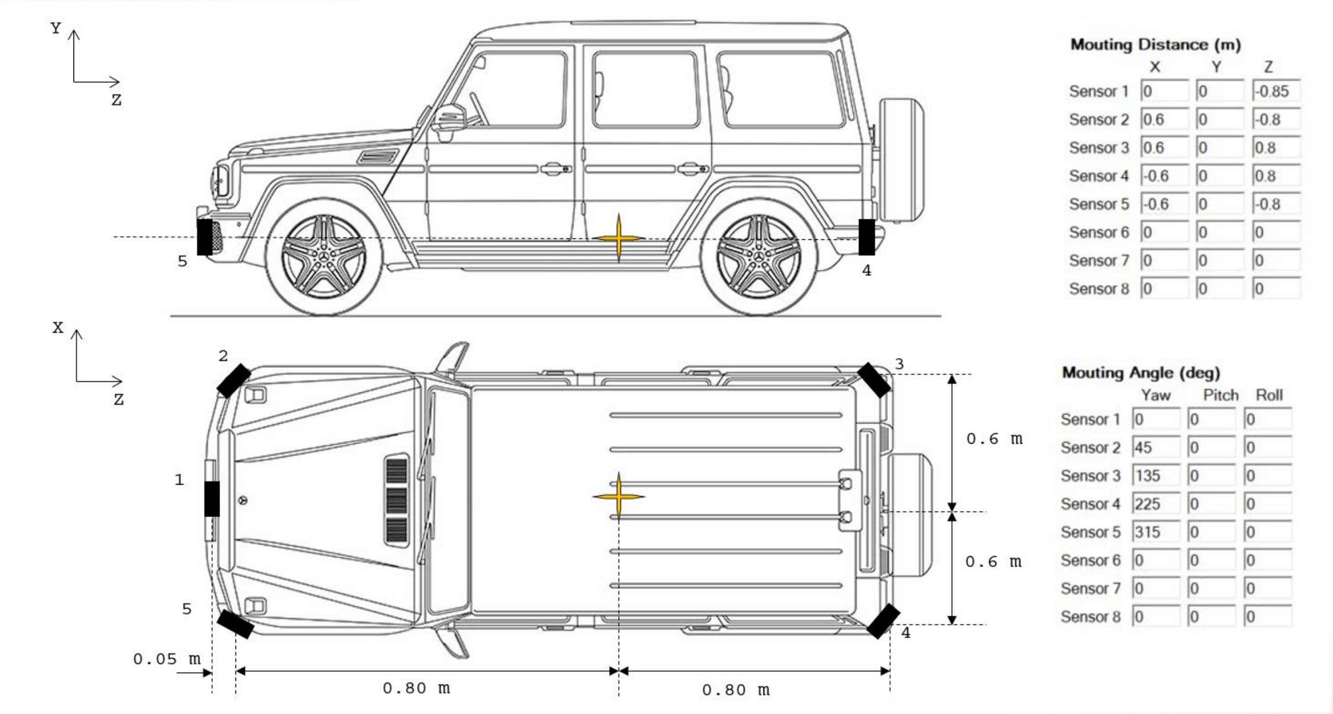
`

8.2 Settings

In the Config, Visualizer and Application tab, there are 7 fields as described below:

i) Mounting Angle (deg) and Mounting Distance (m)
ii) Feature (On/Off)
iii) 3D View
iv) Enhanced Point Cloud
v) Heatmap View Scale
vi) Display Config
vii) Radar Sensors Select

i) Mounting Angle and Mounting Distance (deg):



For multiple sensors connected please note the physical length (z), width (x) and height (y) from the user defined center. In the above diagram the center is defined to be at the same level of installation as the five sensors and at the center of the car. The coordinate axis to be used is defined in the figure as well. The translation values for the five sensors for the figure would be:

Mouting Distance (m)

	X	Y	Z
Sensor 1	0	0	-0.85
Sensor 2	0.6	0	-0.8
Sensor 3	0.6	0	-0.8
Sensor 4	-0.6	0	0.8
Sensor 5	-0.6	0	0.8
Sensor 6	0	0	0
Sensor 7	0	0	0
Sensor 8	0	0	0

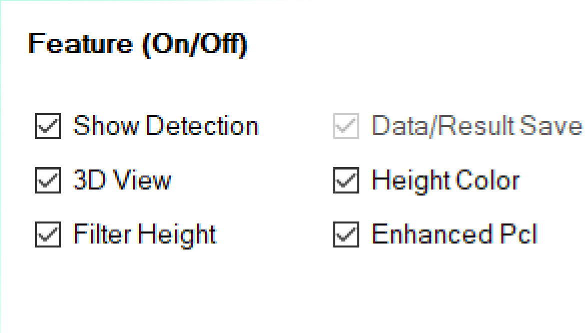
For multiple sensors connected please note the physical yaw (if the sensor is mounted flat at ground level and perpendicular to the ground) or the yaw, pitch, and roll. In the visualizer, clockwise rotation is positive from 0 to 360 degrees. Note that these measurements should be with respect to the center that the user has fixed. The sensors individually always report data to the center of the sensor. In the figure above the sensors are installed at ground level and are perpendicular to the ground. The first sensor is at the center and the rest are at 45 degrees from the plane of the vehicle front and back. Thus, the pitch and roll would be 0. The rotation values for the five sensors for the figure would be:

Mouting Angle (deg)

	Yaw	Pitch	Roll
Sensor 1	0	0	0
Sensor 2	45	0	0
Sensor 3	135	0	0
Sensor 4	225	0	0
Sensor 5	315	0	0
Sensor 6	0	0	0
Sensor 7	0	0	0
Sensor 8	0	0	0

ii) Feature (On/Off):

There are 3 features which can be enabled and disabled in Realtime. If you check the respective checkbox, the feature is enabled. If the checkbox is unchecked, the feature is disabled.



a) Show Detection:

Checked: Point Cloud data is visualized.
Not Checked: Point Cloud data is hidden.

b) Data Save:

Checked: Data is saved while visualizer is running.
Unchecked: Data is not saved while visualizer is running to preserve disc space.

c) Filter Height:

Checked: The visualizer filters the objects above a set height (height set in the Display Config section)
Unchecked: No height filter on the visualizer.

d) Height Color:

Checked: Height color scale applied to the 3D visualization to distinguish objects at different heights.
Unchecked: No height color scale applied to the 3D visualization. Default color scheme is power based.

iii) 3D View:

Under the Feature section there is an option to view the data as a 3D point cloud. Check the box "3D" to proceed. The view on the right-hand side of the visualizer will change to 3D. Here is a list of commands that control the viewing of the 3D viewer.

- Right Click and Drag: Pitch

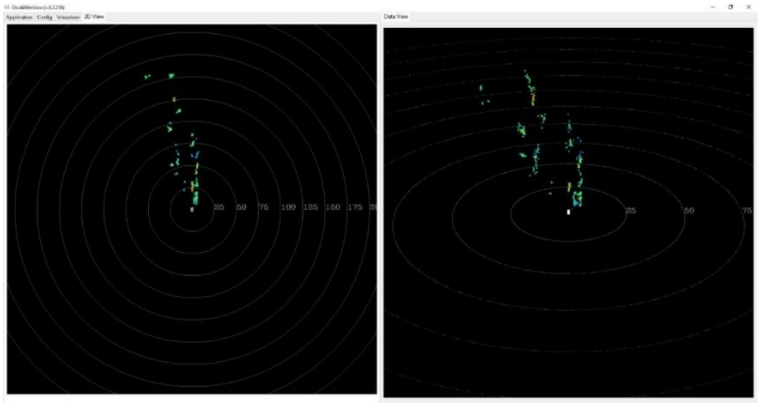
o Left Click and Drag: Yaw
o W: Zoom in
o S: Zoom out
o A: Translation. Move view left (+x direction in radar coordinate)
o D: Translation. Move view right (-x direction in radar coordinate)
o Q: Translation. Move plane down (-y direction in radar coordinate)
o E: Translation. Move plane up (+y direction in radar coordinate)

Under the 3D view there are a few different features that the user can enable/disable certain features as well.

o Height Filter: It removes all the points above the set maximum (maximum height can be set in the Display Config) as well as all points below the ground (0m).
o Height Color: In the 3D view the color is based on height with the red spectrum on VIBGYOR representing targets with a lower height value and the violet spectrum representing targets with a higher height value.
o Show 3D Tracks: Plots the tracks in the 3D Bitmap.

Note: Height filter and Height color will only work together
Note: Height filter and Height color are accurate only when the radar mounting angle has no depression/elevation angle.

The user can also simultaneously view the 2D and 3D point cloud by checking the 3D box and then the 2D Viewer tab found on the top left section of the visualizer. The left-hand side display is the 2D display and the right-hand side display is the 3D display.

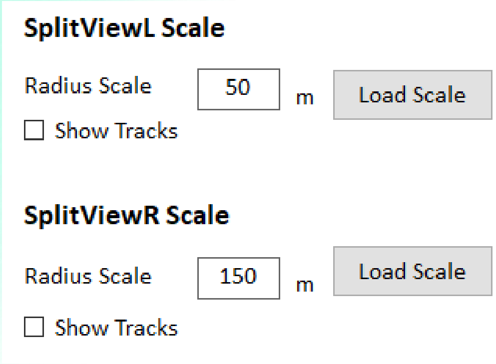


iv) Enhanced Point Cloud

This SDK feature uses the host kinematics and neighborhood information to increase the density of the point cloud information and provide more accurate results with higher confidence. **Checking the enhanced point cloud is recommended on the visualizer.**

v) Heatmap Scale View:

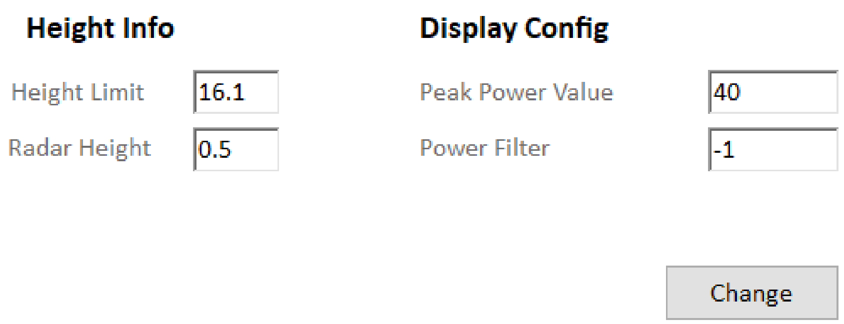
This field is for controlling the scale in the Data View. We have three parameters to control it.



(a) SplitViewL Scale: This scale controls the left bitmap display.
(b) SplitViewR Scale: This scale controls the right bitmap display.
(c) Show Tracks: This button click allows for tracks to be displayed in their respective bitmaps.

vi) Display Config:

There are two parameters which can be controlled for the display config



(i) Peak Power: The Peak power varies from 0 to 100. It can be controlled to show the higher peak points on the display in Data View. There is a color pattern for each peak power range where colors on the red spectrum represent higher power, and on the violet spectrum represent lower power. This field depends on the power threshold filter entered below.

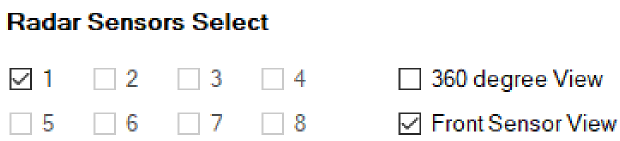
(ii) Power Threshold Filter: The Power Threshold filter varies from 0 to 10. This filters below the specific threshold in dB entered in the textbox. Example: If the power threshold filter is entered as 5, the targets below 5dB are filtered. Default recommended is 0-1dB.

(iii) Height Limit: Used to filter out height information beyond a certain height, the limit of which is set here.

(iv) Radar Height: For correct visualization of the 3D point cloud information, the correct height of the radar from the ground would be required. The grid in the 3D display is the actual ground, and the point cloud information is displayed from that reference.

vii) Radar Sensors Select:

There are two checkboxes in the field which is chosen for the Front Sensor (Single Sensor) and 360-degree (Multi Sensor) views.
Front Sensor View Checked: Radar 1 is automatically checked and only the first radar will display data on the Data View tab.
360 Degree View Checked: All the radars are automatically checked, and they all display data on the Data View tab.



8.3 Saved Results

Oculii saves the following results for the user.

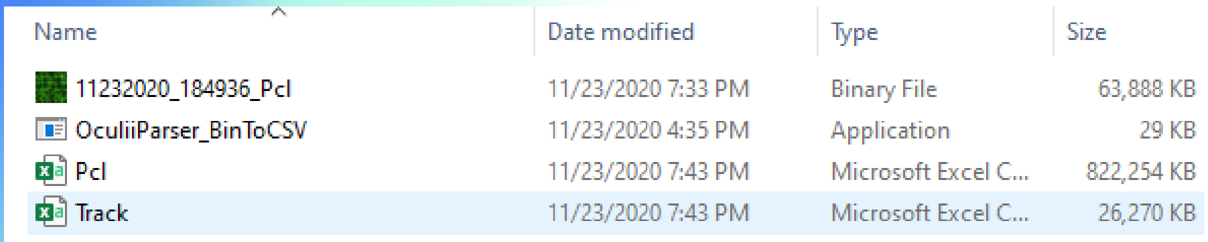
a. Binary Point Cloud Data when the visualizer was running.
b. Point Cloud Data under a .csv format to view on a spreadsheet.
c. Tracker Data under a .csv format to view on a spreadsheet.
d. Saved Point Cloud Video from the visualizer.

The data will be saved under C:\Oculii\Saved Results. A folder will be created with the date the visualizer was run under the [YYYY-MM-DD] format followed by another folder under the [YYYY-MM-DD-HH-MM-SS] format.

8.4 Binary to CSV file Converter

The binary file saved by the visualizer can be converted to CSV to read the data.
To run the Binary to CSV parser:

a. Copy the 'OculiiParser_BinToCSV' executable to the Sensor - Data folder where you will find a file ending with _Pcl.bin.
b. Run the executable



Both the Point Cloud and Tracker CSV file will be generated from the saved binary.

9. FIRMWARE UPDATE ON VISUALIZER

The firmware and configuration files can be updated by following the below procedure.

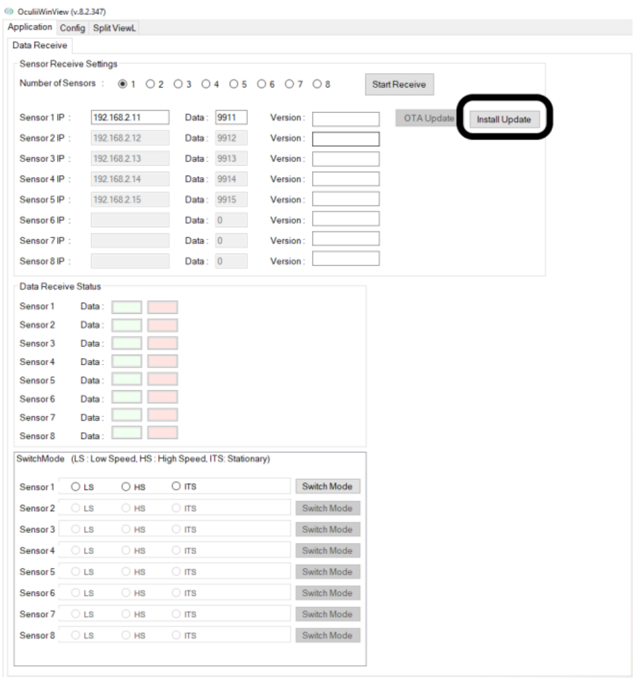
1. The provided Update_Eagle.bin file should be copied to the visualizer folder.

2. Run the Windows Visualizer.

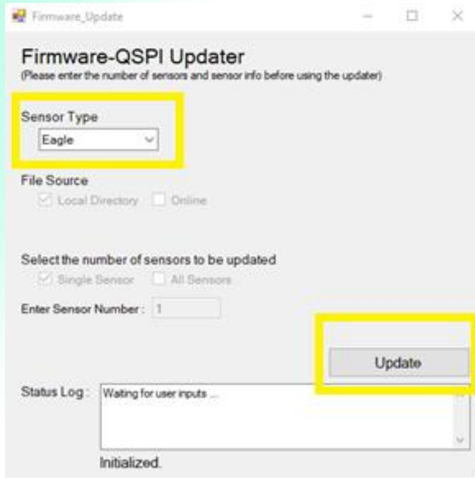
3. Enter the sensor IP and data as shown with the last digit replaced by the last digit of the sensor.

4. Click on the Start Receive button.

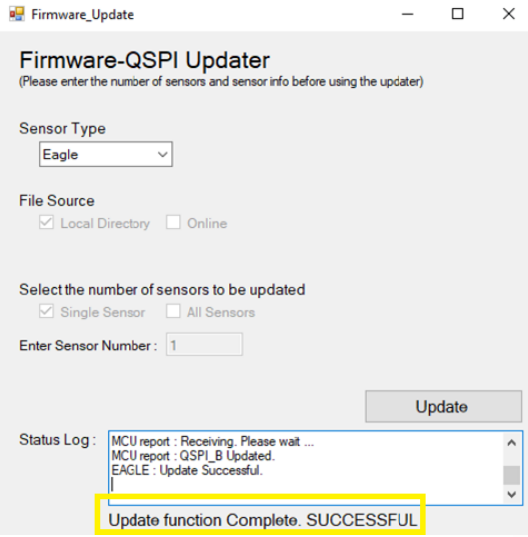
5. Once you receive data and the UDP connection is live, click on the Install Update button.



6. The following pop-up will appear. Select sensor type (Example: **EAGLE)** and then click the Update button.



Note: Please do not close the window before the update completes. Please close the window "Firmware_Update" after update finishes and it is recommended to close the window and click the update button for every update.
If the update is not successful after approx. 3 minutes, please close the entire application and power cycle the sensor before attempting a new update.



7. Once the update is complete, the pop-up displays that the update succeeded as below and the data will be visible on the client. The data and new version appears after appox. 10 seconds.

10. SAFETY AND RISK

10.1 Note and Scope of Responsibility

This section specifies practices the user should adhere to and the risks operators should recognize when operating the Oculii EAGLE sensor. This device should only be handled by technical operators with basic technical knowledge. The owner of the sensor module is responsible for the device and understands and observes the safety notes. If the EAGLE sensor is a part of a larger system, the system manufacturer is responsible for ensuring that all safety features are practiced. The owner of the device is responsible for ensuring that the device is used for its intended purpose and for the actions of their employees while operating the sensor. The owner is also responsible to notify Oculii as soon as the sensor displays any safety defects.

10.2 Operating Risk

i) The system is not defined for free use in safety critical systems. However, it can be integrated into a system by the user, to adhere to safety regulation systems, features, or applications.

ii) The user should be aware of falsified information that could occur over time due to external factors such as mechanical stress, accident, accidental drop, or natural disasters. The user should thus adopt practices that to measure the data output periodically and ensure that it meets expected results. The user should also ensure that any surfaces in the field of view of the sensor are kept clean and metal free. If the sensor starts displaying false information the user should report the same to Oculii.

iii) Lack of knowledge or incomplete training of employees operating the sensor could result in accidents, injury or damage to property, assets, or the environment. Users should ensure all personnel operating the sensor are equipped with the technical knowledge and training to do so.

iv) When deploying multiple sensors ensure that there is no falsified information that is present, either from the interruption of the radar beams or from interference from other sensors or electromagnetic radiations from other devices.

v) When installing the sensor ensure that the front plate or any secondary surface in the field of view is free from the potential accumulation of ice, water or mud films.

vi) Ensure that the power supplied is enough to operate the sensor and that the POE injector ports are matched to their intended destination.

vii) Ensure that the housing of the sensor develops no scratches or cracks as that can impact signal integrity and heat transfer properties.

10.3 Service

Oculii must be contacted before the manipulation of the sensor for certain applications. Oculii must be contacted if the sensor starts displaying falsified information over time. For any other service needs and feature requests you could contact Oculii at techsupport@oculii.com.