

ROS Delphi ESR Driver

The Delphi ESR driver reads data from the radar over CAN or Ethernet and publishes them as ROS topics. With the CAN version, it also exposes topics to allow vehicle motion information to be sent to the radar to improve tracking.

The main design approach is to be as true to the original data as possible.

This way, the original device documentation can be used to understand the output from the ROS driver.

This means the ROS messages used (`delphi_esr_msgs`) are essentially a direct port of the data available over CAN or Ethernet.

Any generic ROS message types (`radar_msgs` and `derived_object_msgs`) conform to ROS standard units and types.

Supported Hardware

- Delphi ESR 2.5 12V
- Delphi ESR 2.5 24V
- Delphi ESR 9.21.21 (CAN Only)
- Delphi ESR 9.21.15 (CAN Only)

CAN Driver (`delphi_esr_can`)

The CAN driver publishes ROS messages that replicate the data from the device.

In some cases, the driver will not perfectly replicate the same data from the device:

- The `esr_track` topic will only publish valid tracks, instead of all 64 tracks reported by the device. Valid tracks are any tracks with a status value greater than zero.

Launch Arguments

- **frame_id**: The frame id of the sensor, topics will be namespaced accordingly and published message headers will use this frame. Defaults to `radar_1`.
- **esr_upside_down**: Flag that can be sent to the ESR to orient the device and receive accurate information regarding track position. Defaults to `false`.
- **use_kvaser**: Set this to true if a Kvaser CAN device is being used with Kvaser canlib drivers to connect to the radar. Defaults to `false`.
- **kvaser_hardware_id**: The hardware id of the kvaser device, only applies if `use_kvaser` is true.
- **kvaser_circuit_id**: The circuit/channel id that the radar is plugged into on the kvaser device, only applies if `use_kvaser` is true.
- **use_socketcan**: Set this to true if Linux SocketCAN drivers are being used to connect to the radar. Defaults to `false`.
- **socketcan_device**: The device id of the SocketCAN channel the radar is plugged into.

Published Topics

Delphi ESR data topics:

- `esr_status_1` (`delphi_esr_msgs/EsrStatus1`)
- `esr_status_2` (`delphi_esr_msgs/EsrStatus2`)
- `esr_status_3` (`delphi_esr_msgs/EsrStatus3`)
- `esr_status_4` (`delphi_esr_msgs/EsrStatus4`)
- `esr_status_5` (`delphi_esr_msgs/EsrStatus5`)
- `esr_status_6` (`delphi_esr_msgs/EsrStatus6`)
- `esr_status_7` (`delphi_esr_msgs/EsrStatus7`)
- `esr_status_8` (`delphi_esr_msgs/EsrStatus8`)
- `esr_status_9` (`delphi_esr_msgs/EsrStatus9`)

- `esr_valid_1` ([delphi_esr_msgs/EsrValid1](#))
- `esr_valid_2` ([delphi_esr_msgs/EsrValid2](#))
- `esr_track` ([delphi_esr_msgs/EsrTrack](#))
- `esr_track_motion_power_group` ([delphi_esr_msgs/EsrTrackMotionPowerGroup](#))

Generic ROS message topics:

- `objects` ([derived_object_msgs/ObjectWithCovarianceArray](#))
- `radar_tracks` ([radar_msgs/RadarTracks](#))

Subscribed Topics

Delphi ESR data topics:

- `esr_vehicle_1` ([delphi_esr_msgs/EsrVehicle1](#))
- `esr_vehicle_2` ([delphi_esr_msgs/EsrVehicle2](#))
- `esr_vehicle_3` ([delphi_esr_msgs/EsrVehicle3](#))
- `esr_vehicle_4` ([delphi_esr_msgs/EsrVehicle4](#))
- `esr_vehicle_5` ([delphi_esr_msgs/EsrVehicle5](#))

Generic ROS message topics:

- `vehicle_motion` ([geometry_msgs/TwistStamped](#))

Ethernet Driver (delphi_esr_eth)

The ethernet driver publishes all received data via the `esr_eth_tx` topic.

The ethernet data is not the same as the CAN data and only contains a minimal amount of raw detection information.

Launch Arguments

- **frame_id**: The frame id of the sensor, topics will be namespaced accordingly and published message headers will use this frame. Defaults to `radar_1`.
- **ip_address**: The ip address of the radar's ethernet connection. Defaults to `169.254.145.71`.
- **port**: The port of the radar's ethernet connection. Defaults to `5555`.

Published Topics

Delphi ESR data topics:

- `raw_data` ([network_interface/TCPFrame](#))
- `esr_eth_tx` ([delphi_esr_msgs/EsrEthTx](#))

Generic ROS message topics:

- `esr_eth_tx` ([radar_msgs/RadarScan](#))

Visualization Node (delphi_esr_viz)

The visualization node subscribes to data published by the `delphi_esr_can` and `delphi_esr_eth` drivers and publishes visualization markers for use in RViz.

Launch Arguments

- **frame_id**: The frame id of the sensor, topics will be namespaced accordingly and published message headers will use this frame. Defaults to `radar_1`.
- **track_status_filter**: A filter for only displaying tracks with a status value equal to or greater than this value. Defaults to `3`.
- **track_amplitude_filter**: A filter for only displaying tracks with an amplitude value equal to or greater than this value. Defaults to `-10` dB.

Published Topics

- `can_viz_markers` ([visualization_msgs/MarkerArray](#))
- `eth_viz_markers` ([visualization_msgs/MarkerArray](#))

Subscribed Topics

- `esr_track`
- `esr_track_motion_power_group`
- `esr_eth_tx`