RobotFramework_DoIP

v. 0.1.0

Hua Van Thong

20.09.2023

Contents

Chapter 1

Introduction

1.1 Introduction TODO

RobotFramework_DoIP is a Robot Framework library specifically designed for interacting with Electronic Control Units (ECUs) using the Diagnostics over Internet Protocol (DoIP).

At its core, DoIP serves as a communication bridge between external diagnostic tools and a vehicle's ECUs. This library, RobotFrameworkDoIP, provides a set of keywords that enable users to perform diagnostic operations and engage with ECUs, facilitating automated testing processes and interaction with vehicles through the DoIP protocol.

The **RobotFramework_DoIP** sources can be found in repository **robotframework-doip**: DoIP

Chapter 2

Description

2.1 Description TODO

Chapter 3

DoipKeywords.py

3.1 Class: DoipKeywords

Imported by:

```
from RobotFramework_DoIP.DoipKeywords import DoipKeywords
```

3.1.1 Method: connect_to_ecu

Description:

Establishing a connection to an (ECU) within the context of automotive communication.

Parameters:

- **param `ecu_ip_address` (required): The IP address of the ECU to establish a connection. This shoul**
  address like "192.168.1.1" or an IPv6 address like "2001:db8::".
- type `ecu_ip_address`: str
- param `ecu_logical_address` (required): The logical address of the ECU.
- type `ecu_logical_address`: int
- param `tcp_port` (optional): The TCP port used for unsecured data communication (default is **TCP_DATA_UNSECURED**).
- type `tcp_port`: int
- param `udp_port` (optional): The UDP port used for ECU discovery (default is **UDP_DISCOVERY**).
- type `udp_port`: int
- param `activation_type` (optional): The type of activation, which can be the default value (ActivationTypeDefault) or a specific value based on application-specific settings.
- type `activation_type`: RoutingActivationRequest.ActivationType,
- param `protocol_version` (optional): The version of the protocol used for the connection (default is 0x02).
- type `protocol_version`: int
- **param `client_logical_address` (optional): The logical address that this DoIP client will use to ide**
  this should be 0x0E00 to 0x0FFF. Can typically be left as default.
- type `client_logical_address`: int
- **param `client_ip_address` (optional): If specified, attempts to bind to this IP as the source for both**
  Useful if you have multiple network adapters. Can be an IPv4 or IPv6 address just like ecu_ip_address, though the type should match.
- type `client_ip_address`: str
- **param `use_secure` (optional): Enables TLS. If set to True, a default SSL context is used. For more**
  SSL context can be passed directly. Untested. Should be combined with changing tcp_port to 3496.

- type `use_secure`: Union[bool,ssl.SSLContext]
- param `auto_reconnect_tcp` (optional): Attempt to automatically reconnect TCP sockets that were closed by peer
- type `auto_reconnect_tcp`: bool

Return:

None

Usage:

Explicitly specifies all establishing a connection

- Connect To ECU | 172.17.0.111 | ${1863} |
- Connect To ECU | 172.17.0.111 | ${1863} | client_ip_address=172.17.0.5 | client_logical_address=1895 |
- Connect To ECU | 172.17.0.111 | ${1863} | client_ip_address=172.17.0.5 | client_logical_address=1895 | activation_type=${0} |

3.1.2 Method: send_diagnostic_message

Description:

Send a raw diagnostic payload (ie: UDS) to the ECU.

Parameters:

- param `diagnostic_payload`: UDS payload to transmit to the ECU
- type `diagnostic_payload`: bytearray
- param `timeout`: send diagnostic time out (default: A_PROCESSING_TIME)
- type `timeout`: int (s)

Return:

None

Exception:

raises IOError: DoIP negative acknowledgement received

Usage:

Explicitly specifies all diagnostic message properties

- Send Diagnostic Message | 1040 |
- Send Diagnostic Message | 1040 | timeout=10 |

3.1.3 Method: receive_diagnostic_message

Description:

Receive a raw diagnostic payload (ie: UDS) from the ECU.

Parameters:

- param `timeout`: time waiting diagnostic message (default: None)
- type `timeout`: int (s)

Return:

None

Exception:

raises IOError: DoIP negative acknowledgement received

Usage:

Explicitly specifies all diagnostic message properties

- Receive Diagnostic Message |
- Receive Diagnostic Message | timeout=10 |

3.1.4 Method: reconnect_to_ecu

Description:

Attempts to re-establish the connection. Useful after an ECU reset

Parameters:

- param `close_delay`: Time to wait between closing and re-opening socket (default: **A_PROCESSING_TIME**)
- type `close_delay`: int (s)

Return: None

Exception:

raises ConnectionRefusedError: DoIP negative acknowledgement received

Usage:

Explicitly specifies all diagnostic message properties

- Reconnect To Ecu |
- Receive Diagnostic Message | timeout=10 |

3.1.5 Method: disconnect

Description:

Close the DoIP client

Parameters:

None

Return:

None

Exception:

None

Usage:

Explicitly specifies all diagnostic message properties

- Disconnect

3.1.6 Method: await_vehicle_announcement

Description:

When an ECU first turns on, it's supposed to broadcast a Vehicle Announcement Message over UDP 3 times to assist DoIP clients in determining ECU IP's and Logical Addresses. Will use an IPv4 socket by default, though this can be overridden with the ipv6 parameter.

Parameters:

- param `udp_port`: The UDP port to listen on. Per the spec this should be 13400, but some VM's use a custom
- one.
- type `udp_port`: int, optional
- param `timeout`: Maximum amount of time to wait for message
- type `timeout`: float, optional
- param `ipv6`: Bool forcing IPV6 socket instead of IPV4 socket
- type `ipv6`: bool, optional
- **param `source_interface`: Interface name (like "eth0") to bind to for use with IPv6. Defaults to No** will use the default interface (which may not be the one connected to the ECU). Does nothing for IPv4, which will bind to all interfaces uses INADDR_ANY.
- type `source_interface`: str, optional

Return:

- return: IP Address of ECU and VehicleAnnouncementMessage object
- rtype: tuple

Exception:

raises TimeoutError: If vehicle announcement not received in time

Usage:

Explicitly specifies all diagnostic message properties

- Await Vehicle Annoucement
- Await Vehicle Annoucement | timeout=10

3.1.7 Method: get_entity

Description:

Sends a VehicleIdentificationRequest and awaits a VehicleIdentificationResponse from the ECU, either with a specified VIN, EIN, or nothing. Equivalent to the request_vehicle_identification() method but can be called without instantiation

Parameters:

- param `udp_port`: The UDP port to listen on. Per the spec this should be 13400, but some VM's use a custom
- one.
- type `udp_port`: int, optional
- param `timeout`: Maximum amount of time to wait for message
- type `timeout`: float, optional
- param `ipv6`: Bool forcing IPV6 socket instead of IPV4 socket

- type ipv6: bool, optional
- **param source_interface: Interface name (like "eth0") to bind to for use with IPv6. Defaults to No**
  will use the default interface (which may not be the one connected to the ECU). Does nothing for IPv4, which will bind to all interfaces uses INADDR_ANY.
- type source_interface: str, optional

Return:

- return: IP Address of ECU and VehicleAnnouncementMessage object
- rtype: tuple

Exception:

raises TimeoutError: If vehicle announcement not received in time

Usage:

- Get Entity |
- Get Entity | ecu_ip_address=172.17.0.111 |
- Get Entity | ecu_ip_address=172.17.0.111 | protocol_version=0x02

3.1.8 Method: request_entity_status

Description:

Request that the ECU send a DoIP Entity Status Response

Parameters:

None

Return:

None

Exception:

None

Usage:

- Request Entity Status

3.1.9 Method: request_vehicle_identification

Description:

Sends a VehicleIdentificationRequest and awaits a VehicleIdentificationResponse from the ECU, either with a specified VIN, EIN, or nothing

Parameters:

param eid EID of the Vehicle

type eid bytes, optional

param vin VIN of the Vehicle

type vin str, optional

Return:

None

Exception:

None

Usage:

- Request Vehicle Identification
- Request Vehicle Identification | eid=0x123456789abc
- Request Vehicle Identification | vin=0x123456789abc

3.1.10 Method: request_alive_check

Description:

Request that the ECU send an alive check response

Parameters:

None

Return:

None

Exception:

None

Usage:

- Request Vehicle Identification
- Request Vehicle Identification | eid=0x123456789abc
- Request Vehicle Identification | vin=0x123456789abc

3.1.11 Method: request_activation

Description:

Requests a given activation type from the ECU for this connection using payload type 0x0005

Parameters:

- **param `activation_type` (required): The type of activation to request - see Table 47 ("Routing** activation request activation types") of ISO-13400, but should generally be 0 (default) or 1 (regulatory diagnostics)
- type `activation_type`: RoutingActivationRequest.ActivationType
- param `vm_specific` (optional): 4 byte long int
- type `vm_specific`: int, optional
- **param `disable_retry`: Disables retry regardless of auto_reconnect_tcp flag. This is used by activatio** requests during connect/reconnect.
- type `disable_retry`: bool, optional

Return:

None

Exception:

None

Usage:

- Request Routing Activation | ${0x02}
- Request Routing Activation | vm_specific=
- Request Routing Activation | vin=0x123456789abc

3.1.12 Method: request_diagnostic_power_mode

Description:

Request that the ECU send a Diagnostic Power Mode response

Parameters:

None

Return:

None

Exception:

None

Usage:

- Request Diagnostic Power Mode

Chapter 4

RobotFramework_DoIP.py

4.1 Function: get_version

4.2 Function: get_version_date

Chapter 5

Appendix

About this package:

Table 5.1: Package setup

Setup parameter	Value
Name	RobotFramework_DoIP
Version	0.1.0
Date	20.09.2023
Description	RobotFramework for DoIP Client
Package URL	robotframework-doip
Author	Hua Van Thong
Email	thong.huavan@vn.bosch.com
Language	Programming Language :: Python :: 3
License	License :: OSI Approved :: Apache Software License
OS	Operating System :: OS Independent
Python required	>=3.0
Development status	Development Status :: 4 - Beta
Intended audience	Intended Audience :: Developers
Topic	Topic :: Software Development

Chapter 6

History

0.1.0	09/2023
Initial version	

RobotFramework_DoIP.pdf
Created at 04.01.2024 - 18:36:52
by GenPackageDoc v. 0.41.1