



OS32C-DM Safety Laser Scanner

EtherNet/IP Dynamic Link Library (DLL) for Microsoft Visual Studio C++ and C# Applications



This addendum is to be used in conjunction with the
OS32C User's Manual (P/N 99863-0010)

EtherNet/IP™
conformance tested

Table of Contents

1	Introduction	3
2	OS32C Dynamic Link Library (DLL) for the EtherNet/IP Communications	3
2.1	OS32C Dynamic Link Library (DLL) Functions.....	3
2.1.1	EIP_EncapsulateListIdentityRequestMessage()	3
2.1.2	EIP_EncapsulateTcpRegisterSessionRequestMessage.....	4
2.1.3	EIP_EncapsulateTcpUnRegisterSessionRequestMessage	5
2.1.4	EIP_EncapsulateTcpSendRRDataExplicitRequestMessage	5
2.1.5	EIP_EncapsulateTcpSendUnitDataExplicitRequestMessage	6
2.1.6	EIP_EncapsulateTcpConnectionKeepAliveRequestMessage	7
2.1.7	EIP_EncapsulateTcpStartUdpIOConnectionRequestMessage	7
2.1.8	EIP_EncapsulateTcpStopUdpIOConnectionRequestMessage.....	8
2.1.9	EIP_EncapsulateUdpIOConnectionKeepAliveRequestMessage.....	9

1 Introduction

The OS32C-xxx-DM Safety Laser Scanner with EtherNet/IP and Measurement Data allows the laser scanner to be monitored by products that adhere to the ODVA guidelines for EtherNet/IP communications. The OS32C with EtherNet/IP functions as an EtherNet/IP target (slave) device to the products that function as EtherNet/IP originator (master) devices. Multiple EtherNet/IP master devices can be set up to monitor a single OS32C and a single EtherNet/IP master can be set up to monitor multiple OS32Cs.

The OS32C with EtherNet/IP uses standard EtherNet/IP communications and does not use CIP safety protocols. EtherNet/IP communications with this laser scanner are for monitoring purposes only. EtherNet/IP originator products are able to monitor the OS32C's detection zone state, input & output status, configuration checksum values, measurement data and more.

This document focuses on how to use the communication software dynamic link library written for Microsoft Visual Studio applications. For full details on the OS32C's produced data assemblies see OS32D-DM appendix B in the OS32C Safety Laser Scanner manual.

2 OS32C Dynamic Link Library (DLL) for the EtherNet/IP Communications

The dynamic link library (DLL) described in this document provides functions for establishing communication sessions, sending and receiving explicit TCP/IP messages, as well as creating UDP IO connections between a computer system and the scanner. The OS32C DLL encapsulates the required messages to be sent to the scanner but does not provide any socket programming functions. Socket functions must be provided by the target application. The OS32C client application example code provided in the OS32C EtherNet/IP documentation kit demonstrates an example of how to use the DLL along with Winsock2 programming. This client application was developed using Microsoft Visual C++ Express which is a free compiler from Microsoft and is available via the internet.

2.1 OS32C Dynamic Link Library (DLL) Functions

2.1.1 EIP_EncapsulateListIdentityRequestMessage()

The EIP_EncapsulateListIdentityRequestMessage function creates the EtherNet/IP communication packet that is sent to the target device in order to determine the device profile of the target device. This command can be used during either TCP/IP or UDP/IP communications. When used during UDP/IP communications a broadcast subnet mask can be used to identify multiple EtherNet/IP devices on a network.

PROTOTYPE

```
EIP_INT32 EIP_EncapsulateListIdentityRequestMessage(EIP_BYTE packet[], EIP_UINT16
max_response_delay);
```

ARGUMENTS

packet	Pointer to communication buffer used during transmission.
max_response_delay	Unsigned 16-bit integer to define the maximum expected response delay.

RETURNED VALUE

Unsigned 32-bit integer	Message size used during transmission.
-------------------------	--

RESPONSE MESSAGE RETURNED

structure	S_Encapsulation_ListIdentityResponse data structure.
-----------	--

2.1.2 EIP_EncapsulateTcpRegisterSessionRequestMessage

The EIP_EncapsulateTcpRegisterSessionRequestMessage function creates the EtherNet/IP communication packet that is sent to the target device in order to initiate a TCP/IP communication session with the scanner. The session handle provided in the response message is then used in all subsequent TCP/IP communications with the target device.

PROTOTYPE

```
EIP_INT32 EIP_EncapsulateTcpRegisterSessionRequestMessage(EIP_BYTE packet[])
```

ARGUMENTS

packet	Pointer to communication buffer used during transmission.
--------	---

RETURNED VALUE

Unsigned 32-bit integer	Message size used during transmission.
-------------------------	--

RESPONSE MESSAGE RETURNED

structure	S_Encapsulation_RegisterSessionResponse data structure.
-----------	---

2.1.3 EIP_EncapsulateTcpUnRegisterSessionRequestMessage

The EIP_EncapsulateTcpUnRegisterSessionRequestMessage function creates the EtherNet/IP communication packet that is sent to the target device in order to terminate a TCP/IP communication session with the scanner. The session handle provided during the initial registration is used to terminate the session and close the TCP/IP socket connection with the target device.

PROTOTYPE

```
EIP_INT32 EIP_EncapsulateTcpUnRegisterSessionRequestMessage(EIP_BYTE packet[], struct
S_Service_Code_Request_Format *request)
```

ARGUMENTS

packet	Pointer to communication buffer used during transmission.
request	Pointer to request structure which contains the session handle.

RETURNED VALUE

Unsigned 32-bit integer	Message size used during transmission.
-------------------------	--

RESPONSE MESSAGE RETURNED

none

2.1.4 EIP_EncapsulateTcpSendRRDataExplicitRequestMessage

The EIP_EncapsulateTcpSendRRDataRequestMessage function creates the EtherNet/IP communication packet that is sent to the target device in order to set or get assembly attributes using TCP/IP communication session with the scanner.

PROTOTYPE

```
EIP_INT32 EIP_EncapsulateTcpSendRRDataRequestMessage(EIP_BYTE packet[], struct
S_Service_Code_Request_Format *request)
```

ARGUMENTS

packet	Pointer to communication buffer used during transmission.
--------	---

request	Pointer to request structure which contains the session handle and service code parameters required to set or get assembly attributes.
---------	--

RETURNED VALUE

Unsigned 32-bit integer	Message size used during transmission.
-------------------------	--

RESPONSE MESSAGE RETURNED

structure	S_Encapsulation_UCMMResponse data structure.
-----------	--

2.1.5 EIP_EncapsulateTcpSendUnitDataExplicitRequestMessage

The EIP_EncapsulateTcpSendUnitDataRequestMessage function creates the EtherNet/IP communication packet that is sent to the target device in order to get assembly attributes using TCP/IP communication session with the scanner.

PROTOTYPE

```
EIP_INT32 EIP_EncapsulateTcpSendUnitDataRequestMessage(EIP_BYTE packet[], struct
S_Service_Code_Request_Format *request)
```

ARGUMENTS

packet	Pointer to communication buffer used during transmission.
request	Pointer to request structure which contains the session handle and service code parameters required to get assembly attributes.

RETURNED VALUE

Unsigned 32-bit integer	Message size used during transmission.
-------------------------	--

RESPONSE MESSAGE RETURNED

structure	S_Encapsulation_UCMMResponse data structure.
-----------	--

2.1.6 EIP_EncapsulateTcpConnectionKeepAliveRequestMessage

The EIP_EncapsulateTcpConnectionKeepAliveRequestMessage function creates the EtherNet/IP communication packet that is sent to the target device in order to maintain the TCP/IP communication session with the scanner. This function sends an EtherNet/IP NOP command to the target device.

PROTOTYPE

```
EIP_INT32 EIP_EncapsulateTcpConnectionKeepAliveRequestMessage(EIP_BYTE packet[], struct
S_Service_Code_Request_Format *request)
```

ARGUMENTS

packet	Pointer to communication buffer used during transmission.
request	Pointer to request structure which contains the session handle.

RETURNED VALUE

Unsigned 32-bit integer	Message size used during transmission.
-------------------------	--

RESPONSE MESSAGE RETURNED

none

2.1.7 EIP_EncapsulateTcpStartUdpIOConnectionRequestMessage

The EIP_EncapsulateTcpStartUdpIOConnectionRequestMessage function creates the EtherNet/IP communication packet that is sent to the target device in order to initiate a UDP IO connection with the scanner in order to stream assembly data structures to the client application at a define repetition rate.

PROTOTYPE

```
EIP_INT32 EIP_EncapsulateTcpStartUdpIOConnectionRequestMessage(EIP_BYTE packet[], struct
S_Service_Code_Request_Format *request)
```

ARGUMENTS

packet	Pointer to communication buffer used during transmission.
request	Pointer to request structure which contains the session handle and service code parameters required to stream assembly data.

RETURNED VALUE

Unsigned 32-bit integer Message size used during transmission.

RESPONSE MESSAGE RETURNED

structure S_Service_Reply_ForwardOpen data structure via the TCP/IP socket.

structure S_Connected_UDP_IO_Packet data structure via the UDP/IP IO socket at the define target repetition rate.

2.1.8 EIP_EncapsulateTcpStopUdpIOConnectionRequestMessage

The EIP_EncapsulateTcpStopUdpIOConnectionRequestMessage function creates the EtherNet/IP communication packet that is sent to the target device in order to terminate the UDP IO connection with the scanner.

PROTOTYPE

```
EIP_INT32 EIP_EncapsulateTcpStopUdpIOConnectionRequestMessage(EIP_BYTE packet[], struct
S_Service_Code_Request_Format *request)
```

ARGUMENTS

packet Pointer to communication buffer used during transmission.

request Pointer to request structure which contains the session handle, interface handle and connection parameters.

RETURNED VALUE

Unsigned 32-bit integer Message size used during transmission.

RESPONSE MESSAGE RETURNED

structure S_Encapsulation_UCMMResponse data structure.

2.1.9 EIP_EncapsulateUdpIOConnectionKeepAliveRequestMessage

The EIP_EncapsulateUdpIOConnectionKeepAliveRequestMessage function creates the EtherNet/IP communication packet that is sent to the target device in order to maintain the UDP IO communication connection with the scanner. This command must be sent at the originator repetition rate defined during the creation of the IO connection.

PROTOTYPE

```
EIP_INT32 EIP_EncapsulateUdpIOConnectionKeepAliveRequestMessage(EIP_BYTE packet[], struct
S_Service_Code_Request_Format *request)
```

ARGUMENTS

packet	Pointer to communication buffer used during transmission.
request	Pointer to request structure which contains the session handle, interface handle and connection parameters.

RETURNED VALUE

Unsigned 32-bit integer	Message size used during transmission.
-------------------------	--

RESPONSE MESSAGE RETURNED

structure	S_Connected_UDP_IO_Packet data structure via the UDP/IP IO socket at the define target repetition rate.
-----------	---