

# **CMX 1553 MIL-STD-1553 MODULE**

## **OPERATING MANUAL**

<b>Doküman Ref.:</b>	KNK - CMX - 2502
<b>Version:</b>	1
<b>Tarih:</b>	28.03.2025

### Revision Table

Date	Revision	Page No	Ver.
25.02.2025	First release	-	01

## Index

Revision Table .....	2
Index .....	3
List of Figures .....	7
List of Tables.....	8
1. Introduction .....	9
2. Description.....	9
3. Technical Specifications .....	10
4. Pinout.....	11
5. Setup and Configuration .....	12
5.1. Hardware Setup.....	12
5.2. Software Setup .....	12
5.3. Configuration .....	13
6. Data Acquisition and Usage Scenarios .....	14
7. CmxWinSDK Driver Software .....	14
7.1 Cmx1553OpenBoard.....	14
7.2 Cmx1553BoardSetMode .....	15
7.3 Cmx1553BoardGetMode .....	15
7.4 Cmx1553Reset.....	16
7.5 Cmx1553SharedInit.....	16

7.6 Cmx1553BcInitMsg.....	16
7.7 Cmx1553BcUpdateMsg.....	18
7.8 Cmx1553BcInitMsgList .....	20
7.9 Cmx1553BcSetTTAG .....	20
7.10 Cmx1553BcGetMsgInProgress .....	21
7.11 Cmx1553BcGetMsgBlock.....	21
7.12 Cmx1553BcGetMsgData .....	22
7.13 Cmx1553BcEnable .....	22
7.14 Cmx1553BcStop.....	22
7.15 Cmx1553BcTrigger.....	23
7.16 Cmx1553BcStart.....	23
7.17 Cmx1553MtSetTTAG .....	23
7.18 Cmx1553MtSetFilter .....	24
7.19 Cmx1553MtGetLastCommand .....	24
7.20 Cmx1553MtGetLastMessage .....	25
7.21 Cmx1553MtEnable .....	25
7.22 Cmx1553MtStop .....	25
7.23 Cmx1553Rt1Init.....	25
7.24 Cmx1553Rt1Init1 .....	26
7.25 Cmx1553Rt1Init2 .....	26
7.26 Cmx1553Rt1SetAddr.....	27

7.27 Cmx1553Rt1SetTTAG.....	27
7.28 Cmx1553Rt1SetTXData .....	28
7.29 Cmx1553Rt1Enable.....	28
7.30 Cmx1553Rt1Stop .....	28
7.31 Cmx1553Rt1Start .....	29
7.32 Cmx1553Rt2Init.....	29
7.33 Cmx1553Rt2Init1 .....	30
7.34 Cmx1553Rt2Init2 .....	30
7.35 Cmx1553Rt2SetAddr.....	30
7.36 Cmx1553Rt2SetTTAG.....	31
7.37 Cmx1553Rt2SetTXData .....	31
7.38 Cmx1553Rt2Enable.....	32
7.39 Cmx1553Rt2Stop .....	32
7.40 Cmx1553Rt2Start .....	32
7.41 Cmx1553RtCheckAddr .....	33
7.42 Cmx1553RtGetMessage .....	33
7.43 Cmx1553DiscardBoard.....	33
7.44 Cmx1553SetIP.....	34
7.45 Cmx1553GetIP .....	34
7.46 Cmx1553ResetIP.....	34
7.47 Cmx1553GetFirmware.....	35

7.48 CmxDASetMAC .....	35
7.49 Cmx1553GetMAC.....	35
8. CMXAPI Driver Software .....	36
9. CmxWinSDK LabVIEW API Library .....	36
10. Error Codes .....	36
11. Safety Precautions.....	36
12. Usage in Hazardous Locations.....	37
13. Compliance with Electronic Standards .....	37
14. Maintenance and Troubleshooting .....	37
15. Manufacturer Warranty and Warranty Exclusions .....	38
16. Use in Critical Applications .....	39
17. Legal Disclaimer .....	39
18. Restricted and Unauthorized Use Disclaimer .....	40
Support and Contact Information .....	41
Ordering Guide .....	41

### List of Figures

Figure 1 - CMX 1553 Board .....	10
Figure 2 - CMX IP Configurator .....	13

### List of Tables

Table 1 - DB15 Pinout.....	11
Table 2 - DB25 Pinout.....	11



## 1. Introduction

This operating manual provides information on the usage of the CMX 1553 MIL-STD-1553 module. It details the principle of operation, the technical specifications, physical structure, communication interfaces, configuration settings and software interface of the device.

## 2. Description

CMX 1553 is part of the modular test and measurement hardware platform developed by Konaka Defense. It is used in defense, aerospace, and industrial applications for testing, measurement, data acquisition, and control systems.

CMX 1553 operates as a MIL-STD-1553 protocol interface card configured as a Bus Controller (BC), Dual Remote Terminal (RT), or Bus Monitor (BM). The module features two independent, dual-redundant data bus interfaces (A - B), ensuring reliable communication in critical applications.

The Bus Controller (BC) functions as a programmable message sequencing device. In MIL-STD-1553B applications, the BC can use six internal or external clock options with either 16-bit or 32-bit configurations. These options can be selected by the user via the software interface. The BC data bus interface supports BC-RT, RT-BC, broadcast messages, and mode code messages.

- The programmable Bus Controller autonomously supports multi-frame message scheduling, data storage, host error reporting, and message retry functionalities.
- CMX 1553 can be configured to operate as one or two Remote Terminals (RT1 - RT2). RTs handle communication between the data bus and subsystems, containing the necessary electronics. Each RT address can include up to 32 different sub addresses. RTs can process messages received from the BC, provide status responses, or send messages to the BC.
- When configured as a Bus Monitor (BM), CMX 1553 acts as a specialized remote terminal that monitors and records all or specific messages on the bus based on user-defined requirements. It is typically used for two purposes: data recording and replay in offline applications and as a backup for BCs.

CMX 1553 supports buffered data acquisition. The device processes commands received via Ethernet, USB, SPI, or RS485 and updates the message structure accordingly.



*Figure 1 - CMX 1553 Board*

While CMX 1553 can be used in a chassis combined with the CMX MB2 processor, it also supports standalone operation. By incorporating Ethernet, USB, or RS485 communication interfaces, the CMX 1553 can be used without the CMX MB2 processor and CMX CH8 chassis. This capability allows CMX 1553 to function as an OEM board.

### 3. Technical Specifications

- 1 independent dual-redundant A-B bus interfaces
- Multifunction operation (BC – RT – BM)
- Direct-coupled MIL-STD-1553 terminal
- Operating temperature range: -40 to +70°C

#### Communication Interface:

- Ethernet (10 Mbit)
- USB (FS, Virtual Com Port)
- RS 485 (115200 kbps, 8 bits, 1 stop bit, no parity), voltage level is 5V. 120 ohm terminated.
- SPI supported when using with CMX MB2 processor board

#### Power requirements:

- 18-36VDC input power
- 1.5-Watt typical power consumption (24VDC supply reference)

#### Physical dimensions:

- Board dimensions are 10 x 10 cm.

## Environmental:

- Both commercial and industrial temperature ranges are available. Refer to ordering guide.

## 4. Pinout

CMX 1553 has 2 connectors on it. One is 15 pin Dsub male, the other one is 25 pin Dsub male. When using CMX CH8 chassis, DB15 connector is plugged into backplane, and it is powered by chassis.

Table 1 - DB15 Pinout

Pin No	Description
1	Power Supply (+)
2	Power Supply (-)
3	NC
4	GND
5	RS 485 (+)
6	RS 485 (-)
7	GND
8	GPIO (3.3V)
9	Reserved for SPI Bus
10	
11	
12	
13	
14	
15	GND

Table 2 - DB25 Pinout

Pin No	Description
1	GND
2	TRIGGER IN (5V)
3	GND
4	BUS B (-)
5	BUS B (+)
6	GND
7	GND
8	GND
9	BUS A (+)
10	BUS A (-)
11	GND
12	GND

13	GND
14	GND
15	GND
16	GND
17	GND
18	GND
19	GND
20	GND
21	GND
22	GND
23	GND
24	GND
25	GND

## 5. Setup and Configuration

### 5.1. Hardware Setup

1. Choose operating mode if used it as stand alone or in CMX CH8 chassis.
2. If used in stand-alone mode, connect 24VDC power supply to DB15 connector. (Refer to [pinout table](#))
3. If used in stand-alone mode, identify communication port the product will be used. (Refer to technical specifications)
4. If used with ethernet port, the default IP address is 192.168.1.30. Set the IP address of the computer on the same network.
5. If used with ethernet port, energize the board and then run PING to verify Ethernet connection.
6. If used with USB port, energize the board then check “Device Manager” on computer if it identifies any com port.
7. If used in CMX CH8, insert the module into the slot which module will be used. Chassis must be powered OFF during mounting.

### 5.2. Software Setup

1. If used in stand-alone mode, CMXAPI driver software is being used to communicate with the board.
2. Download and install CMXAPI driver software from manufacturer website [www.konaka.com.tr](http://www.konaka.com.tr) or contact [support@konaka.com.tr](mailto:support@konaka.com.tr) CMXAPI driver software is supported by Windows OS only. Both 32- and 64-Bit distributions are available.
3. If the Linux® operating system is being used, contact the manufacturer for driver request.
4. If the board will be used with RS485 interface under any OS, contact the manufacturer for ICD documentation. If OS is Windows, CMXAPI driver is used.
5. If used in CMX CH8 chassis with CMX MB2 processor, CmxWinSDK32 or CmxWinSDK64 driver software is being used to communicate with the board. CmxWinSDK32 is 32 Bit distribution and CMXWinSDK64 is 64 Bit distribution. Download and install CmxWinSDK driver software from manufacturer website [www.konaka.com.tr](http://www.konaka.com.tr) or contact [support@konaka.com.tr](mailto:support@konaka.com.tr). CmxWinSDK driver software is supported by Windows OS only.

6. If the Linux® operating system is being used, contact the manufacturer for driver request.
  7. All relevant software is being installed on C:\Konaka Defence\ directory.
  8. Sample applications are available on the manufacturer's website. Contact the manufacturer for any sample applications request.
  9. For any customization requests on driver software, contact the manufacturer.
  10. Both CMXAPI and CmxWinSDK driver software, NI LabVIEW SDK library is available. NI LabVIEW sample applications are available on manufacturer's website.
- ✓ Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
  - ✓ Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries.
  - ✓ LabVIEW is a registered trademark of National Instruments (NI) Corporation in the United States and/or other countries.

### 5.3. Configuration

When the driver's software is installed, the device is ready to operate. If IP addresses need to be updated, follow the instructions below.

1. If the device will be used in chassis, download the CMX IP configurator utility from the manufacturer's website. It should be installed on C:\Konaka Defence\Utility\CMX IP Configurator. When CMX 1553 is used in chassis, it doesn't communicate with its' own interface. CMX MB2 processor manages CMX 1553 module over SPI bus interface. CMX IP Configurator tool updates CMX MB2 processor IP address.

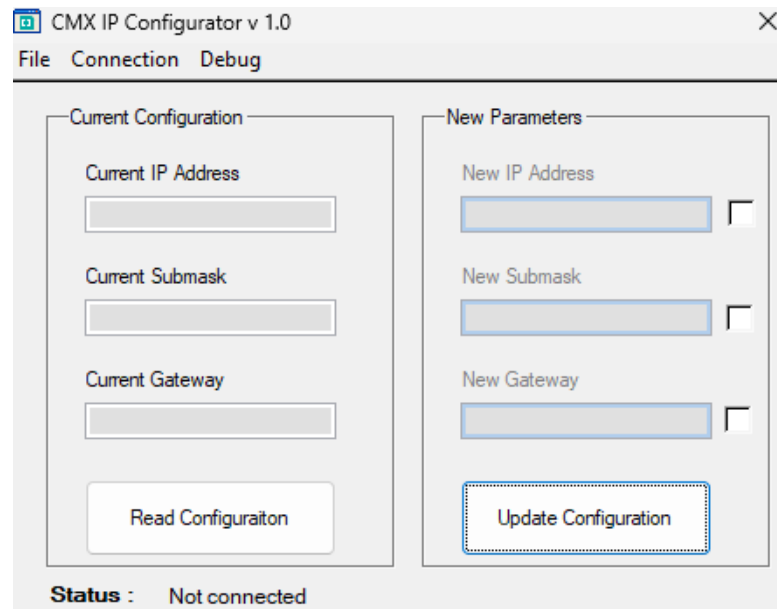


Figure 2 - CMX IP Configurator

2. If the device will be used stand-alone, CMX 1553 communicates over Ethernet, USB or RS485. IP

address can be updated through CMXAPI driver.

## 6. Data Acquisition and Usage Scenarios

### BC, RT, and MT Modes

The CMX 1553 can be configured as Bus Controller (BC), Remote Terminal (RT), or Bus Monitor (MT). concurrently. The user should determine the mode in which the device will operate before proceeding with the configuration.

- **Bus Controller (BC) Mode:**

The BC controls all communication on the bus and manages message sequencing. A Bus List can be created with 32 RX, 32 TX, 4 Broadcast, and 2 Mode Code messages, allowing a total of 70 messages to be handled simultaneously. Message intervals and the transition to the next cycle after completing the bus list can be adjusted by setting the period time. The bus list can operate in either single-pass or loop mode. Additionally, message transmission can be triggered using the trigger method. In BC mode, messages can be updated at the time of transmission.

- **Remote Terminal (RT) Mode:**

RTs respond to commands or send messages on the bus. Each RT supports 32 different sub addresses for communication. In RT mode, received and sent messages can be logged.

- **Bus Monitor (MT) Mode:**

MT listens to and records all messages on the bus or selected messages. This mode is typically used for analysis or as a backup.

Time Tag settings can be configured in all modes. This setup provides a flexible and robust bus communication solution compliant with the MIL-STD-1553 protocol.

## 7. CmxWinSDK Driver Software

CmxWinSDK is a driver software which provides access to all CMX hardware modules. In this section, CMX 1553 functions are explained in detail. For the latest version of CmxWinSDK driver software, refer to the manufacturer's website. These functions are the same for both 32- and 64-Bit distribution.

### 7.1 Cmx1553OpenBoard

This function connects to CMX 1553 board.

```
unsigned int Cmx1553OpenBoard (char* IPAddress, unsigned int BoardID, unsigned int *Error);
```

## Input parameters:

- IP Address: IP address of the CMX MB2 processor board
- BoardID: Slot number of the module. Should be between 1...8

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

**Note:** This function returns "BoardHandle". This BoardHandle will be used as input parameter of other functions.

## 7.2 Cmx1553BoardSetMode

This function is used to select the modes in which the hardware will operate.

```
unsigned int Cmx1553BoardSetMode (unsigned int Handle, unsigned int BCMMode, unsigned int RT1Mode, unsigned int RT2Mode, unsigned int MTMode, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.
- BCMMode: Activate Bus Controller.
- RT1Mode: Activate Remote Terminal 1.
- RT2Mode: Activate Remote Terminal 2.
- Mt\_mode: Activate Monitor Terminal.
  - o ENABLE
  - o DISABLE

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

## 7.3 Cmx1553BoardGetMode

This function returns information about which modes of the 1553 module are active.

```
unsigned int Cmx1553BoardGetMode (unsigned int Handle, unsigned int *Mode, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

Mode: Information about the active modes on the hardware.

- 0x01: BC Mode Enable
- 0x02:RT1 Mode Enable
- 0x04:RT2 Mode Enable
- 0x08:MT Mode Enable
- Error: Error code. Refer to "[Error Codes](#)"

### 7.4 Cmx1553Reset

This function resets the 1553 chip to prepare the hardware for operation.

```
unsigned int Cmx1553Reset (unsigned int Handle, unsigned int *Error);
```

#### Input parameters:

- Handle: Boardhandle of the module.

#### Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.5 Cmx1553SharedInit

This function is used to perform the initialization processes of the active modes on the hardware.

```
unsigned int Cmx1553SharedInit (unsigned int Handle, unsigned int *Error);
```

#### Input parameters:

- Handle: Boardhandle of the module.

#### Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.6 Cmx1553BcInitMsg

This function is used to create a Bus Controller message.

```
unsigned int Cmx1553BcInitMsg (unsigned int Handle, unsigned int Message_Type, unsigned int Index,
unsigned int Data_Count, unsigned short *data, unsigned int Control_Word, unsigned int RT_Addr,
unsigned int RT_Subaddr, unsigned int Mcode, unsigned int *Error);
```

#### Input parameters:

- Handle: Boardhandle of the module.
- Message\_Type: Message type selection.



- 0x00: RX Subbaddress Message
- 0x01: TX Subbaddress Message
- 0x02: Broadcast Message
- 0x03: TX Mode Code Message
- 0x04: RX Mode Code Message
- Index: Message Indexes
  - RX Subbaddress Message : 0x01-0x20
  - TX Subbaddress Message : 0x01-0x20
  - Broadcast Message : 0x01-0x04
  - TX Mode Code Message : 0x01
  - RX Mode Code Message : 0x01
- Data\_Count: The number of data words the user wants to send or receive.
  - 0x00-0x20
- Data: Message Datas.
  - 0x0000-0xFFFF
- Control\_Word: Message Control Word
  - BCST : 0x0002
  - MCODE: 0x0004
  - EOMINT: 0x0010
  - MSKBCR: 0x0020
  - SFTEST:0x0040
  - USEBUSB:0x0000
  - USEBUSA:0x0080
  - RTRYENA:0x0100
  - RSVMASK:0x0200
  - TFMASK:0x0400
  - SSFMASK:0x0800
  - BSYMASK:0x1000
  - SRQMASK:0x2000
  - MEMASK:0x4000
  - TXTMC17:0x8000
- RT\_Addr: Remote terminal address to which the message will be sent.
  - 0x00-0x1F
- RT\_Subaddr: Remote terminal sub-address to which the message will be sent.
  - 0x01-0x1E
- Mcode: Mode Code Number.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

**NOTE:** The Cmx1553BcInitMsg function is used to create messages. The user can create 32 different RX, 32 different TX, 4 Broadcast, 1 TX Mode Code, and 1 RX Mode Code message.

**INDEX:** The Index value is a unique identifier assigned to each message.

For example, if the user creates 6 messages:

- 1. RX Subaddress message - INDEX = 1
- 2. RX Subaddress message - INDEX = 2
- 1. Broadcast message - INDEX = 1
- 1. TX Subaddress message - INDEX = 1
- 2. TX Subaddress message - INDEX = 2
- 3. RX Subaddress message - INDEX = 3

The Index parameter should be entered in this manner.

**Data:** If the user selects 0x00 or 0x02 as Message\_Type, these fields are used to enter the data. If other message types are selected, these parameters will not be processed.

**Control Word:** While creating message blocks, the user can manually select the control words. The necessary control words should be combined using the OR operation and entered as a parameter.

- If it is a Broadcast message, BCST: 0x0002 must be added.
- If it is a Mode Code message, MCODE: 0x0004 must be added.

**Mcode:** This parameter is ignored if Message\_Type is not equal to 0x03 or 0x04.

## 7.7 Cmx1553BcUpdateMsg

This function is used to update a Bus Controller message.

`unsigned int Cmx1553BcUpdateMsg (unsigned int Handle, unsigned int Message_Type, unsigned int Index, unsigned int Data_Count, unsigned short *data , unsigned int Control_Word , unsigned int RT_Addr , unsigned int RT_Subaddr, unsigned int Mcode, unsigned int *Error);`

## Input parameters:

- Handle: Boardhandle of the module.
- Message\_Type: Message type selection.
  - 0x00: RX Subbaddress Message
  - 0x01: TX Subbaddress Message
  - 0x02: Broadcast Message

- 0x03: TX Mode Code Message
- 0x04: RX Mode Code Message
- Index: Index of the message to be updated
  - RX Subbaddress Message : 0x01-0x20
  - TX Subbaddress Message : 0x01-0x20
  - Broadcast Message : 0x01-0x04
  - TX Mode Code Message : 0x01
  - RX Mode Code Message : 0x01
- Data\_Count: The number of data words the user wants to send or receive.
  - 0x00-0x20
- Data: Message Datas.
  - 0x0000-0xFFFF
- Control\_Word: Message Control Word
  - BCST : 0x0002
  - MCODE: 0x0004
  - EOMINT: 0x0010
  - MSKBCR: 0x0020
  - SFTEST:0x0040
  - USEBUSB:0x0000
  - USEBUSA:0x0080
  - RTRYENA:0x0100
  - RSVMASK:0x0200
  - TFMASK:0x0400
  - SSFMASK:0x0800
  - BSYMASK:0x1000
  - SRQMASK:0x2000
  - MEMASK:0x4000
  - TXTMC17:0x8000
- RT\_Addr: Remote terminal address to which the message will be sent.
  - 0x00-0x1F
- RT\_Subaddr: Remote terminal sub-address to which the message will be sent.
  - 0x01-0x1E
- Mcode: Mode Code Number.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

**NOTE:** By using this command, values such as data contents and control word contents of the messages can be modified. For example, previously created messages could be as follows:

1. RX Subbaddress message - INDEX = 1
2. RX Subbaddress message - INDEX = 2
3. Broadcast message - INDEX = 1
4. RX Subbaddress message - INDEX = 2

If the 2nd message needs to be updated, Message\_Type parameter should be set to 0x00, and Index parameter should be set to 0x02. The remaining parameters are the ones that can be updated. In this case, the 2nd RX Subbaddress message will be updated. While messages in the bus list are being transmitted, the update operation can be performed.

## 7.8 Cmx1553BcInitMsgList

This function is used to create the message list of the Bus Controller.

```
unsigned int Cmx1553BcInitMsgList (unsigned int Handle, unsigned int Mode, unsigned int Loop,
unsigned int Msg_Gap_Time, unsigned int Period, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.
- Mode: Mode selection
  - o 0x00: Timestamp Mode
  - o 0x01: Trigger Mode
- Loop: Loop Enable
  - o 0x00: Loop Disable
  - o 0x01: Loop Enable
- Msg\_Gap\_Time: Time interval between messages (us).
  - o A value between a minimum of 100 µs and a maximum of 6,553,500 µs can be entered.
- Period: Bus list period duration (us)
  - o A value between a minimum of 100 µs and a maximum of 6,553,500 µs can be entered.

### Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

## 7.9 Cmx1553BcSetTTAG

This function is used to set the Bus Controller time tag resolution and value.

```
unsigned int Cmx1553BcSetTTAG (unsigned int Handle, unsigned int HiLo, unsigned int BcTTAG,
unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.
- HiLo: Bus Controller Time Tag Resolution.
  - o 0x00: 16 bit Low Resolution
  - o 0x01: 32 bit High Resolution
- BcTTAG: Bus Controller Time Tag Value.
  - o 0x02: TTAG\_2U
  - o 0x03: TTAG\_4U
  - o 0x04: TTAG\_8U
  - o 0x05: TTAG\_16U
  - o 0x06: TTAG\_32U
  - o 0x07: TTAG\_64U

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.10 Cmx1553BcGetMsgInProgress

This function provides information on whether a message being processed in the Bus Controller is present or not.

```
unsigned int Cmx1553BcGetMsgInProgress (unsigned int Handle, unsigned int * MsgProgress, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"
- MsgProgress: Information about the processed message
  - o 0x00: No processed message.
  - o 0x01: Processed message exists.

### 7.11 Cmx1553BcGetMsgBlock

This function returns the processed message block in the Bus Controller. It returns values such as Control Word, Command Word, Data Addr, Time to Next Msg, Time Tag Word, Block Status, Loopback Word, and RT Status.

`unsigned int Cmx1553BcGetMsgBlock (unsigned int Handle, unsigned short * MsgBlock, unsigned int *Error);`

### Input parameters:

- Handle: Boardhandle of the module.

### Output parameters:

- Error: Error code. Refer to “Error Codes”
- MsgBlock: Information about the processed message

## 7.12 Cmx1553BcGetMsgData

This function returns the processed message data in the Bus Controller.

`unsigned int Cmx1553BcGetMsgData (unsigned int Handle, unsigned short * MsgData, unsigned int *Error);`

### Input parameters:

- Handle: Boardhandle of the module.

### Output parameters:

- Error: Error code. Refer to “Error Codes”
- MsgData: Bus Controller Message Datas.

## 7.13 Cmx1553BcEnable

This function is used to enable the Bus Controller terminal.

`unsigned int Cmx1553BcEnable (unsigned int Handle, unsigned int *Error);`

### Input parameters:

- Handle: Boardhandle of the module.

### Output parameters:

- Error: Error code. Refer to “[Error Codes](#)”

## 7.14 Cmx1553BcStop

This function is used to stop the Bus Controller terminal.

`unsigned int Cmx1553BcStop (unsigned int Handle, unsigned int *Error);`

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.15 Cmx1553BcTrigger

This function generates a trigger pulse for the Bus Controller. It is used when the messages are created with trigger mode for sending during the Bus Controller Message List creation.

`unsigned int Cmx1553BcTrigger (unsigned int Handle, unsigned int *Error);`

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.16 Cmx1553BcStart

This function is used to start the Bus Controller terminal.

`unsigned int Cmx1553BcStart (unsigned int Handle, unsigned int *Error);`

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.17 Cmx1553MtSetTTAG

This function is used to set the Bus Monitor time tag resolution and value.

`unsigned int Cmx1553MtSetTTAG (unsigned int Handle, unsigned int HiLo, unsigned int MTTAG, unsigned int *Error);`

## Input parameters:

- Handle: Boardhandle of the module.
- HiLo: Bus Monitor Time Tag Resolution.

- 0x00: 16 bit Low Resolution
- 0x01: 32 bit High Resolution
- MTTAG: Bus Monitor Time Tag Value.
  - 0x00: MTTAG OFF
  - 0x02: MTTAG\_2U
  - 0x03: MTTAG\_4U
  - 0x04: MTTAG\_8U
  - 0x05: MTTAG\_16U
  - 0x06: MTTAG\_32U
  - 0x07: MTTAG\_64U
  - 0x80: MTTAG\_100N

#### Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

#### 7.18 Cmx1553MtSetFilter

This function initializes the Bus Monitor address list. It sets up the channels to be monitored.

```
unsigned int Cmx1553MtSetFilter (unsigned int Handle, unsigned int *Error);
```

#### Input parameters:

- Handle: Boardhandle of the module.

#### Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

#### 7.19 Cmx1553MtGetLastCommand

This function returns the last command word sent over the Bus.

```
unsigned int Cmx1553MtGetLastCommand (unsigned int Handle, unsigned short *LastCommand, unsigned int *Error);
```

#### Input parameters:

- Handle: Boardhandle of the module.

#### Output parameters:

- Error: Error code. Refer to "Error Codes"
- LastCommand: Last Command Word.



## 7.20 Cmx1553MtGetLastMessage

This function returns the last message sequence sent over the Bus.

```
unsigned int Cmx1553MtGetLastMessage (unsigned int Handle, unsigned short * LastMessage, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.

### Output parameters:

- Error: Error code. Refer to “Error Codes”
- LastMessage: Last Message.

## 7.21 Cmx1553MtEnable

This function is used to start the Bus Monitor terminal.

```
unsigned int Cmx1553MtEnable (unsigned int Handle, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.

### Output parameters:

- Error: Error code. Refer to “[Error Codes](#)”

## 7.22 Cmx1553MtStop

This function is used to stop the Bus Monitor terminal.

```
unsigned int Cmx1553MtStop (unsigned int Handle, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.

### Output parameters:

- Error: Error code. Refer to “Error Codes”

## 7.23 Cmx1553Rt1Init

This function is used for the initialization of Remote Terminal 1.

`unsigned int Cmx1553Rt1Init (unsigned int Handle, unsigned int Support_Broadcast, unsigned int Undef_Mcodes_Valid , unsigned int Use_Smcp, unsigned int *Error);`

## Input parameters:

- Handle: Boardhandle of the module.
- Support\_broadcast: A parameter used to enable/disable RT1/RT2 broadcast commands according to MIL-STD-1553.
  - o 0x00: No Support Broadcast Messages
  - o 0x01: Support Broadcast Messages
- Undef\_Mcodes\_Valid : A parameter used to enable/disable undefined mode code commands.
  - o 0x00: Undefined Mode Code Commands Not Valid
  - o 0x01: Undefined Mode Code Commands Valid
- Use\_Smcp: A parameter used to enable/disable simplified mode command processing for RT1/RT2.
  - o 0x00: Don't use SMCP
  - o 0x01: use SMCP

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.24 Cmx1553Rt1Init1

This function is used for the initialization of Remote Terminal 1.

`unsigned int Cmx1553Rt1Init1 (unsigned int Handle,, unsigned int Use_Smcp, unsigned int *Error);`

## Input parameters:

- Handle: Boardhandle of the module.
- Use\_Smcp: A parameter used to enable/disable simplified mode command processing for RT1/RT2.
  - o 0x00: Don't use SMCP
  - o 0x01: Use SMCP

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.25 Cmx1553Rt1Init2

This function is used for the initialization of Remote Terminal 1 (RT1). To complete the RT1 initialization process, the functions CM\_RT1\_INIT, CM\_RT1\_INIT1, and CM\_RT1\_INIT2 must be called sequentially.

```
unsigned int Cmx1553Rt1Init2 (unsigned int Handle, unsigned int IllegalCmd_Detect, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.
- IllegalCmd\_Detect: This parameter is used to enable/disable "invalid command detection" for RT1/RT2 according to MIL-STD-1553.
  - o 0x00: Illegal Comand Not Detect
  - o 0x00: Illegal Command Detect

### Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.26 Cmx1553Rt1SetAddr

This function is used to select the Remote Terminal 1 address.

```
unsigned int Cmx1553Rt1SetAddr (unsigned int Handle, unsigned int Addr, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.
- Addr: This is the RT1 address to be assigned.
  - o 0x00-0x1F

### Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.27 Cmx1553Rt1SetTTAG

This function is used to select the time tag clock for Remote Terminal 1.

```
unsigned int Cmx1553Rt1SetTTAG (unsigned int Handle, unsigned int RT1Ttag, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.
- RT1Ttag: Remote Terminal 1 time tag value.
  - o 0x02: TTAG\_2U (2us clock)
  - o 0X03: TTAG\_4U (4us clock)
  - o 0X04: TTAG\_8U (8us clock)
  - o 0X05: TTAG\_16U (16us clock)

- 0X06: TTAG\_32U (32us clock)
- 0X07: TTAG\_64U (64us clock)

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

**NOTE:** Remote Terminal 1 and Remote Terminal 2 always use 16-bit time tag resolution. If the Bus Controller is enabled, RT1 and RT2 share the Bus Controller's time tag value. If it is not enabled, the value entered by the user in the CmxRT1SetTTAG and CmxRT2SetTTAG functions is processed as the RT1 and RT2 time tag value.

## 7.28 Cmx1553Rt1SetTXData

This function is used to load data into the transmit buffer of Remote Terminal 1.

`unsigned int Cmx1553Rt1SetTXData (unsigned int Handle, unsigned short *RT1TxData, unsigned int Data_Count, unsigned int *Error);`

## Input parameters:

- Handle: Boardhandle of the module.
- RT1TxData: The data array to be loaded into the Remote Terminal 1 transmit buffer.
- Data\_Count: The length of the data to be loaded (in bytes), with a maximum of 32.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

## 7.29 Cmx1553Rt1Enable

This function is used to enable Remote Terminal 1.

`unsigned int Cmx1553Rt1Enable (unsigned int Handle, unsigned int *Error);`

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

## 7.30 Cmx1553Rt1Stop

This function is used to stop the Remote Terminal 1.

```
unsigned int Cmx1553Rt1Stop (unsigned int Handle, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.

### Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.31 Cmx1553Rt1Start

This function is used to start Remote Terminal 1.

```
unsigned int Cmx1553Rt1Start (unsigned int Handle, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.

### Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.32 Cmx1553Rt2Init

This function is used for the initialization of Remote Terminal 2.

```
unsigned int Cmx1553Rt2Init (unsigned int Handle, unsigned int Support_Broadcast, unsigned int Undef_Mcodes_Valid, unsigned int Use_Smcp, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.
- Support\_broadcast: A parameter used to enable/disable RT1/RT2 broadcast commands according to MIL-STD-1553.
  - o 0x00: No Support Broadcast Messages
  - o 0x01: Support Broadcast Messages
- Undef\_Mcodes\_Valid : A parameter used to enable/disable undefined mode code commands.
  - o 0x00: Undefined Mode Code Commands Not Valid
  - o 0x01: Undefined Mode Code Commands Valid
- Use\_Smcp: A parameter used to enable/disable simplified mode command processing for RT1/RT2.
  - o 0x00: Don't use SMCP
  - o 0x01: use SMCP

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.33 Cmx1553Rt2Init1

This function is used for the initialization of Remote Terminal 2.

`unsigned int Cmx1553Rt2Init1 (unsigned int Handle, unsigned int Use_Smcp, unsigned int *Error);`

## Input parameters:

- Handle: Boardhandle of the module.
- Use\_Smcp: A parameter used to enable/disable simplified mode command processing for RT1/RT2.
  - o 0x00: Don't use SMCP
  - o 0x00: Use SMCP

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.34 Cmx1553Rt2Init2

This function is used for the initialization of Remote Terminal 1 (RT1). To complete the RT1 initialization process, the functions CM\_RT1\_INIT, CM\_RT1\_INIT1, and CM\_RT1\_INIT2 must be called sequentially.

`unsigned int Cmx1553Rt2Init2 (unsigned int Handle, unsigned int IllegalCmd_Detect, unsigned int *Error);`

## Input parameters:

- Handle: Boardhandle of the module.
- IllegalCmd\_Detect: This parameter is used to enable/disable "invalid command detection" for RT1/RT2 according to MIL-STD-1553.
  - o 0x00: Illegal Command Not Detect
  - o 0x00: Illegal Command Detect

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.35 Cmx1553Rt2SetAddr

This function is used to select the Remote Terminal 2 address.

```
unsigned int Cmx1553Rt2SetAddr (unsigned int Handle, unsigned int Addr, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.
- Addr: This is the RT2 address to be assigned.
  - o 0x00-0x1F

### Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.36 Cmx1553Rt2SetTTAG

This function is used to select the time tag clock for Remote Terminal 2.

```
unsigned int Cmx1553Rt2SetTTAG (unsigned int Handle, unsigned int RT2Ttag, unsigned int *Error);
```

### Input parameters:

- Handle: Boardhandle of the module.
- RT2Ttag: Remote Terminal 2 time tag value.
  - o 0x02: TTAG\_2U (2us clock)
  - o 0x03: TTAG\_4U (4us clock)
  - o 0x04: TTAG\_8U (8us clock)
  - o 0x05: TTAG\_16U (16us clock)
  - o 0x06: TTAG\_32U (32us clock)
  - o 0x07: TTAG\_64U (64us clock)

### Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

**NOTE:** Remote Terminal 1 and Remote Terminal 2 always use 16-bit time tag resolution. If the Bus Controller is enabled, RT1 and RT2 share the Bus Controller's time tag value. If it is not enabled, the value entered by the user in the CmxRT1SetTTAG and CmxRT2SetTTAG functions is processed as the RT1 and RT2 time tag value.

### 7.37 Cmx1553Rt2SetTXData

This function is used to load data into the transmit buffer of Remote Terminal 2.

```
unsigned int Cmx1553Rt2SetTXData (unsigned int Handle, unsigned short *RT2TxData, unsigned int Data_Count, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.
- RT2TxData: The data array to be loaded into the Remote Terminal 2 transmit buffer.
- Data\_Count: The length of the data to be loaded (in bytes), with a maximum of 32.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.38 Cmx1553Rt2Enable

This function is used to enable Remote Terminal 2.

```
unsigned int Cmx1553Rt2Enable (unsigned int Handle, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.39 Cmx1553Rt2Stop

This function is used to stop Remote Terminal 2.

```
unsigned int Cmx1553Rt2Stop (unsigned int Handle, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.40 Cmx1553Rt2Start

This function is used to start Remote Terminal 2.

```
unsigned int Cmx1553Rt2Start (unsigned int Handle, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.



## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.41 Cmx1553RtCheckAddr

This function performs the parity check for the address entered for Remote Terminal 1 or 2.

```
unsigned int Cmx1553RtCheckAddr (unsigned int Handle, unsigned int Rt_Num, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.
- Rt\_Num: Remote Terminal Number.
  - o 0x01: RT1
  - o 0x02: RT2

## Output parameters:

- Error: Error code. Refer to "Error Codes"

### 7.42 Cmx1553RtGetMessage

This function returns the messages received or transmitted by Remote Terminal 1 or 2.

```
unsigned int Cmx1553RtGetMessage (unsigned int Handle, unsigned short* Rtmmessage, unsigned int Rt_Num, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.
- Rt\_Num: Remote Terminal Number.
  - o 0x01: RT1
  - o 0x02: RT2

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"
- Rtmmessage : RT messages.

### 7.43 Cmx1553DiscardBoard

This function is used to disconnect the connection with the connected hardware.

```
unsigned int Cmx1553DiscardBoard (unsigned int Handle, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.44 Cmx1553SetIP

This function sets the IP address of the module. When it is used stand-alone, the module communicates over ethernet using this IP address.

```
unsigned int Cmx1553SetIP (unsigned int Handle, unsigned int NewIPAddress[4], unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.
- NewIPAddress[4]: Enter the new IP address as four bytes.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.45 Cmx1553GetIP

This function reads the IP address of the module.

```
unsigned int Cmx1553GetIP (unsigned int Handle, unsigned int *NewIPAddress[4], unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- NewIPAddress[4]: Read the new IP address as four bytes.
- Error: Error code. Refer to "[Error Codes](#)"

### 7.46 Cmx1553ResetIP

This function resets the module IP address to default 192.168.1.30.

```
unsigned int Cmx1553ResetIP (unsigned int Handle, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- Error: Error code. Refer to "[Error Codes](#)"

### 7.47 Cmx1553GetFirmware

This function gets the module firmware version.

```
unsigned int Cmx1553GetFirmware (unsigned int Handle, double *FirmwareVersion, unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- FirmwareVersion: Read the module firmware version as double type.
- Error: Error code. Refer to "[Error Codes](#)"

### 7.48 CmxDASetMAC

This function sets the MAC address of the module.

```
unsigned int Cmx1553SetMAC (unsigned int Handle, unsigned int *NewMAC[6], unsigned int *Error);
```

## Input parameters:

- Handle: Boardhandle of the module.

## Output parameters:

- NewMAC[6]: Read the new mac address as six bytes.
- Error: Error code. Refer to "[Error Codes](#)"

### 7.49 Cmx1553GetMAC

This function gets the MAC address of the module.

```
unsigned int Cmx1553GetMAC (unsigned int Handle, unsigned int *NewMAC[6], unsigned int *Error);
```

**Input parameters:**

- Handle: Boardhandle of the module.

**Output parameters:**

- NewMAC[6]: Read the new mac address as six bytes.
- Error: Error code. Refer to “Error Codes”

## **8. CMXAPI Driver Software**

CMXAPI is a driver software to operate CMX 1553 in stand-alone mode. This driver software allows users to operate the board through Ethernet, USB or RS485 interface.

All functions in CMXAPI driver software are the same as CmxWinSDK driver software.

## **9. CmxWinSDK LabVIEW API Library**

Konaka Defence released LabVIEW API library by using CmxWinSDK32 32bit DLL file. This library allows CMX 1553 to be used in LabVIEW. The library can be downloaded from the manufacturer's website.

## **10. Error Codes**

This section provides error codes and definitions. These error codes are applicable to both CmxWinSDK and CMXAPI driver software.

- 0x01: No Error. Command success.
- 0x02: Board ID Error.
- 0x03: CRC Error.
- 0x04: Timeout Error.
- 0x05: Interface Error. (This is applicable to CMXAPI driver only).
- 0x06: Invalid Parameter Error.
- 0x07: Unsupported Function Error
- 0x08: No Connection Error
- 0x09: Command Send Error

## **11. Safety Precautions**

To ensure safe operation and prevent damage to the device or harm to personnel, follow these precautions:

- Only qualified personnel should install and operate the device.
- Ensure that all power connections and wiring are performed according to the manufacturer's

guidelines.

- Avoid exposure to excessive heat, moisture, or mechanical stress.
- Do not disassemble or modify the device, as this may void the warranty and cause malfunction.

## 12. Usage in Hazardous Locations

CMX 1553 is not certified for use in hazardous or explosive environments. If the device must be used in such conditions, proper protective measures and enclosures should be employed to prevent risk.

- Follow all local regulations and industry standards when operating in environments with flammable gases, vapors, or dust.
- If required, use explosion-proof enclosures or isolators.
- Regularly inspect the device and its connections to ensure safety compliance

## 13. Compliance with Electronic Standards

CMX 1553 is manufactured in accordance with industry-standard electronic quality and safety regulations. These standards ensure that the device meets high reliability and performance requirements for industrial and defense applications.

## 14. Maintenance and Troubleshooting

### Periodic Maintenance Requirements

To ensure the long-term reliability and performance of the CMX 1553, the following maintenance procedures are recommended:

- **Visual Inspection:** Regularly check for physical damage, loose connectors, or signs of overheating.
- **Connection Check:** Ensure that all wiring and connectors are securely fastened and free of oxidation or dirt.
- **Firmware Updates:** Keep the firmware updated to the latest version provided by the manufacturer.
- **Environmental Conditions:** Ensure that the device operates within the recommended temperature, humidity, and vibration limits.

### Troubleshooting

Common issues and their possible solutions:

- **No Power:** Check power connections and ensure the correct voltage supply.
- **Incorrect Output/Input Signal:** Verify the configuration settings and calibration data.
- **Communication Failure:** Ensure proper connection of Ethernet, USB, SPI, or RS485 interfaces and check for driver/software issues.
- **Intermittent Operation:** Inspect for loose connectors, grounding issues, or environmental factors

affecting performance.

- **Firmware Malfunction:** Perform a firmware upgrade using the SWD interface.

### Firmware Upgrade Guideline

CMX 1553 supports firmware upgrades via the SWD interface. The upgrade process follows these steps:

1. Remove the module from the chassis.
2. Power on the module using DB15 connector before initiating the update. (Refer to the pinout table).
3. Use an ST-Link debugger to connect via the SWD interface (5 pin header J1 connector)
4. Download the latest firmware from the manufacturer's website
5. Download ST Link utility from STMicroelectronics website.
6. Import binary file into ST Link utility software tool then download and verify the firmware.

**Important Note:** Don't erase the EEPROM of the board during operation. Otherwise, all factory data will be erased and the device may not operate properly.

### Trademark Notice

CMX 1553 utilizes components from STMicroelectronics. STMicroelectronics, the ST logo, and other related marks are trademarks or registered trademarks of STMicroelectronics in various jurisdictions.

This product's use of STMicroelectronics components does not imply any affiliation, endorsement, or sponsorship by STMicroelectronics. All rights to STMicroelectronics' intellectual property remain with STMicroelectronics. The manufacturer of CMX 1553 assumes sole responsibility for the product's design, functionality, and compliance with applicable regulations.

### Trademark Notice

CMX 1553 utilizes components from Holt Integrated Circuits. Holt Integrated Circuits, the Holt logo, and other related marks are trademarks or registered trademarks of Holt Integrated Circuits in various jurisdictions.

**Holtic®** is a registered trademark of Holt Integrated Circuits, Inc. All other trademarks and trade names are the property of their respective owners. This product may include a technology licensed from Holt Integrated Circuits, Inc.

## 15. Manufacturer Warranty and Warranty Exclusions

CMX 1553 is covered under a one-year limited warranty from the date of purchase, ensuring protection against defects in materials and workmanship.

- **Warranty Coverage:**
  - Repair or replacement of defective components within the warranty period.
  - Technical support for troubleshooting issues covered under warranty terms.
- **Warranty Exclusions:**
  - Damage due to improper use, accidents, or unauthorized modifications.
  - Exposure to extreme environmental conditions beyond specified limits.
  - Use in hazardous locations without proper protective enclosures.
  - Failure to follow calibration and maintenance recommendations.
  - Electrical damage caused by incorrect wiring or overvoltage conditions.

## 16. Use in Critical Applications

CMX 1553 is **not designed for applications where human life is at risk or where it could directly impact human health**. The manufacturer is not legally responsible for the use of this device in such applications.

## 17. Legal Disclaimer

The CMX 1553 must be used strictly in accordance with the manual provided. Any modifications, adjustments, or alternative usage outside the manufacturer's recommendations are the sole responsibility of the user. The manufacturer assumes no liability for damages resulting from unauthorized changes or improper use of the device.

This product incorporates components from Holt Integrated Circuits, Inc. ("Holt"). Holt Integrated Circuits, the Holt logo, and related marks are trademarks or registered trademarks of Holt Integrated Circuits, Inc. in various jurisdictions.

Holt components are provided under their respective terms and conditions. The inclusion of Holt components in this product does not imply any endorsement, certification, or warranty by Holt Integrated Circuits, Inc. All specifications, features, and performance characteristics are subject to change without notice.

The manufacturer of this product assumes no liability for any direct, indirect, incidental, or consequential damage arising from the use of Holt components. Users are responsible for ensuring that the product complies with applicable regulations and safety requirements for their intended applications.

For detailed technical specifications and terms of use, please refer to Holt Integrated Circuits' official documentation.

---

## 18. Restricted and Unauthorized Use Disclaimer

This product incorporates components from Holt Integrated Circuits, Inc. ("Holt"). The manufacturer strictly prohibits the use of Holt components in any applications that violate local, national, or international laws and regulations. Additionally, Holt components must not be used in applications involving weapons, military systems, life-critical medical devices, or any other restricted or prohibited uses without explicit authorization from the relevant authorities.

The manufacturer assumes no responsibility or liability for any damage, losses, or legal consequences arising from the misuse of Holt components in unauthorized or illegal applications. The end user is solely responsible for ensuring compliance with all applicable laws and regulations regarding the use of this product.

For further details, please refer to Holt Integrated Circuits' official terms of use and licensing agreements.



### Support and Contact Information

For technical support, warranty claims, or general inquiries regarding CMX 1553, please contact the manufacturer through the following channels:

#### Technical Support

- **Email:** [support@konaka.com.tr](mailto:support@konaka.com.tr)
- **Phone:** +90 312 5024278
- **Website:** [www.konaka.com.tr](http://www.konaka.com.tr)

#### Warranty & Repair Services

If you encounter any issues requiring warranty service or repairs, please provide:

- A detailed description of the problem.
- The serial number of the device.
- Proof of purchase or warranty documentation.

#### Sales & Orders

For purchasing inquiries, bulk orders, or distributor information, contact:

- **Email:** [sales@konaka.com.tr](mailto:sales@konaka.com.tr)
- **Phone:** +90 312 5024278

For urgent support requests, please check the Maintenance & Troubleshooting section before reaching out.

### Ordering Guide

CMX 1553-1 has multiple configurations which must be defined prior to ordering.

The following table shows how to create PN.

CMX 1553 – 1 – R0

- CMX 1553-1: Base product code
- R0: Remote IO configuration (Without Enclosure)
- R1: Remote IO configuration (With Enclosure)

**Note:** Contact us for custom design manufacturing requests.

© 2025 Konaka Savunma Sanayi ve Ticaret Limited Şirketi. Her hakkı saklıdır. Diğer tüm markalar veya isimler, ilgili sahiplerinin mülkiyetindedir.

Bu belge, sistem entegrasyonu için özel ürün operasyon bilgileri sağlamak amacıyla Konaka Savunma müşterileri için hazırlanmıştır. Konaka'nın yazılı izni olmadan izinsiz kullanım yasaktır. Konaka, ürün güncellemelerini, düzeltmeleri ve açıklamaları içerecek şekilde bu belgeyi revize etme hakkını saklı tutar ve eski sürümlere uymayabilir. Bu belgede sağlanan bilgilerin doğru olduğuna inanılmaktadır ve tasarım garantileri, satılabilirlik veya belirli bir amaca uygunluk dahil ancak bunlarla sınırlı olmamak üzere açık veya zımni herhangi bir beyan veya garanti olmaksızın "olduğu gibi" sağlanmaktadır. Konaka Savunma, ürünlerin kullanımıyla ilgili herhangi bir sorumluluk kabul etmez ve burada yer alan herhangi bir bilgiye güvenmekten kaynaklanan hiçbir sorumluluktan sorumlu olmayacaktır. Dolaylı olarak veya bununla bağlantılı olarak hiçbir lisans veya hak verilmez.

© 2025 Konaka Savunma Sanayi ve Ticaret Limited Şirketi. All rights reserved. All other brands or names are property of their respective holders.

This document has been produced for the customers of Konaka Defence, with the intent and purpose of providing specific product operation information for systems integration. Unauthorized use or intent is prohibited without written permission from Konaka. Konaka reserves the right to revise this document to include product updates, corrections, and clarifications and may not conform in every aspect to former issues. The information provided in this document is believed to be accurate and is provided "as is" with no representations or warranties of any kind whether expressed or implied, including, but not limited to, warranties of design, merchantability or fitness for a particular purpose. Konaka Defence does not assume any responsibility for its use and shall not be responsible for any liability resulting from reliance upon any information contained herein. No licenses or rights are granted by implication or otherwise in connection therewith.