



# Lynxtester



**GenerationRFID**  
EOL Test & Embedded Electronics



## LYNX53

**CAN controlled Solid State Relay Test  
Boards for Functional EOL testers**

## Table of contents

---

<b>1. GENERAL DESCRIPTION .....</b>	<b>2</b>	8.4.1. Enable Relay.....	6
<b>2. FEATURES.....</b>	<b>2</b>	8.4.2. Disable Relay .....	6
<b>4. TERMINOLOGY.....</b>	<b>1</b>	8.4.3. Disable All Relays .....	7
<b>5. APPLICATION SETUPS .....</b>	<b>1</b>	8.4.4. Enable/Disable Watchdog.....	7
5.1. External Stimulus to DUT .....	1	8.4.5. Refresh Watchdog.....	7
<b>6. ELECTRICAL SPECIFICATIONS.....</b>	<b>1</b>	8.4.6. Set Serial Number.....	8
6.1. Recommended operating conditions	1	8.4.7. Set Product Name .....	8
6.2. Absolute Maximum Ratings .....	1	8.4.8. Set Hardware Version .....	9
<b>7. PINOUT .....</b>	<b>1</b>	8.4.9. Can Enable Configuration Of The	9
7.1. DUT connector Interface .....	1	Serial	9
7.2. Backplane Connector.....	2	8.4.10. Can Configuration Of The Serial .....	9
<b>8. DID LIST.....</b>	<b>2</b>	<b>9. OTHER RELEVANT CHARACTERISTICS .</b>	<b>10</b>
8.1. Configuration .....	2	9.1. Watchdog functionality.....	10
8.2. Read Data Diagnostic List.....	2	9.1.1. Fall-safe behavior .....	10
8.3. GET RELAYS STATE .....	3	9.2. Control via CAN (UDS service 0x2E). 10	
8.3.1. Read Software Version .....	3	9.2.1. Operation summary.....	10
8.3.2. Read Hardware Version .....	4	9.3. LEDs Manage.....	10
8.3.3. Read Product Id .....	4	9.4. Relay Command Behavior .....	11
8.3.4. Read Product Name .....	5	9.5. Can Identifier Configuration .....	11
8.3.5. Read Serial Number .....	5		
8.4. Write Data Diagnostic List .....	6		

# Lynx53

## CAN controlled Solid State Relay Test Boards for Functional EOL testers

### 1. GENERAL DESCRIPTION

The **Lynx53** is a solid-state relay-based test board designed for EOL testing of electronic products.

It provides **24 independent test channels** to connect DUT pins to both common lines (BK+ and BK-) and **external stimulus**. Each DUT pin is routed through two solid-state relays, enabling flexible switching as illustrated in Figure 1.

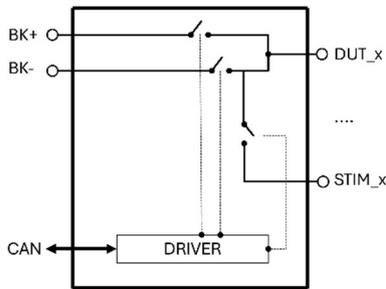


Figure 1. Lynx53 functional Block Diagram

When connected to DMM measuring lines, the platform allows monitoring of multiple physical variables, such as voltages, resistances, capacitances, inductances, or frequencies, depending on the DMM capabilities. It is also designed to switch external stimulus into DUT pins.

The design ensures reliable and isolated distribution of signals and power rails, making it suitable for demanding testing and automation environments.

Each channel can be individually activated or deactivated, providing maximum flexibility for configuring test benches, switching external devices, or integrating into industrial processes.

The Lynx53 is controlled via CAN bus, or via Ethernet when integrated within the Lynx Chassis

### 2. FEATURES

**Board size:** 293 × 100 mm (stackable board for 3U, 19" subracks).

**Scalability:** expandable when combined with Lynx backplanes

**Compatibility:** fully integrated with Lynx Tester Scheduler software

**Number of channels:** 50 independent test channels, each implemented with two solid-state relays.

**Connector Type:** female DB50.

**Communications:**

- CAN bus when used standalone.
- Ethernet when connected through a Lynx Chassis.

**Status indicators (LEDs):**

- **Blue:** General status (Power/OK).
- **Green:** CAN activity.
- **Red:** Error or fault indication red.

**Power supply:** 12 V DC.

**Connectors:**

- DB50 connector for the 50 relay outputs.
- DIN connector (Backplane) for power supply, CAN bus, and synchronization (SYNC).

**Application highlights:**

- Industrial automation – remote control and distribution of power or signals in manufacturing environments.
- Signal multiplexing – selective routing of digital or analog signals to measurement instruments.
- System integration – scalable control solution when combined with Lynx Chassis and LynxTester software.
- Remote device management – activation or deactivation of external circuits via CAN commands

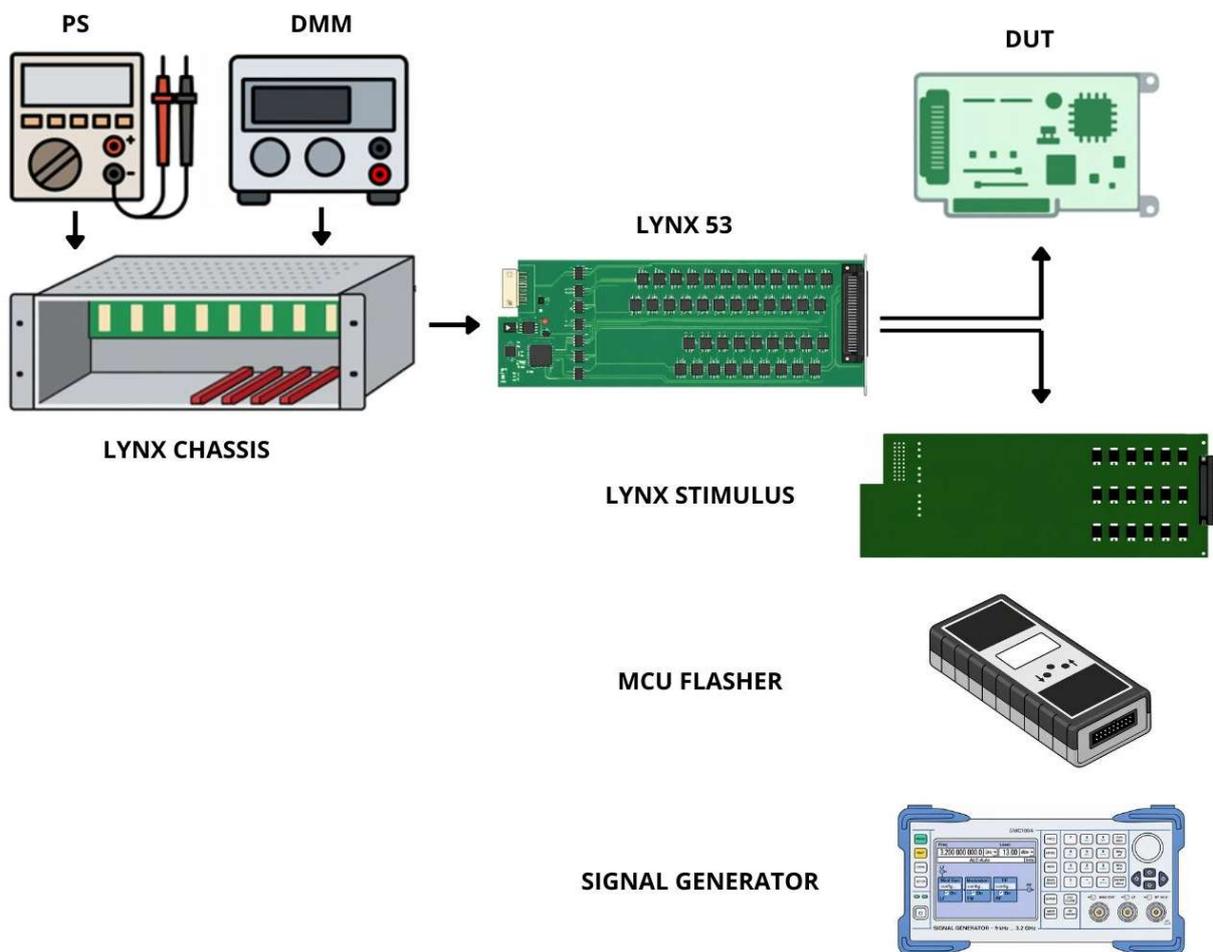
## 4. TERMINOLOGY

- EOL** → End of Line
- DUT** → Device Under Test
- DMM** → Digital Multimeter
- MCU** → Microcontroller Unit
- CAN** → Controller Area Network
- MIPI-10** → Mobile Industry Processor Interface (10-pin debug interface)
- SSR** → Solid-State Relay
- DB50** → 50-pin D-sub connector
- BK** → Backline common rails (BK+ positive, BK - negative)
- SYNC** → Synchronization line between boards
- GND** → Ground reference
- ICT** => In Circuit Test
- TP** => test point

## 5. APPLICATION SETUPS

### 5.1. External Stimulus to DUT

Every electronic DUT requires a functional validation test to ensure that all the components are properly assembled and the functions are correct. A key point on a functional test is to stimulate the DUT with the proper signal. For instance, pull-down resistors for the digital inputs, specific loads for the outputs or even digital signals to flash the MCU or generate a specific wave pattern to apply into an analog input. Lynx53, combined with Lynx Stimulus PCBA can easily do this with the minimum external components.



## 6. ELECTRICAL SPECIFICATIONS

### 6.1. Recommended operating conditions

PARAMETER	NOTES	SYMBOL	MIN	TYP	MAX	UNIT
Supply voltage (nominal)	Input via backplane or external	$V_{IN}$	8	12	24	V
Current consumption	All relays OFF, 12 V supplied	$I_{IDLE}$		380		mA
On-resistance (DC only)	$I_F = 10 \text{ mA}$ , $I_L = 2 \text{ A}$	$R_{ON}$		0.05	0.07	$\Omega$
Turn-on time	$I_F = 10 \text{ mA}$ , $V_L = 5 \text{ V}$ , $I_L = 1 \text{ A}$	$t_{ON}$	370		550	$\mu\text{s}$
Turn-off time	$I_F = 10 \text{ mA}$ , $V_L = 5 \text{ V}$ , $I_L = 1 \text{ A}$	$t_{OFF}$	18		50	$\mu\text{s}$
Relay activation latency (UDS → ON)	Command reception to actuation	$t_{UDS}$			750	$\mu\text{s}$
Operating temperature	Industrial grade	$T_{AMB}$	0		+70	$^{\circ}\text{C}$

Table 1. Electrical characteristics

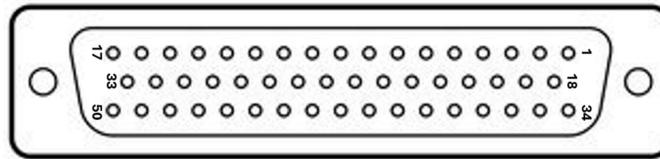
### 6.2. Absolute Maximum Ratings

PARAMETER	SYMBOL	VALUE	UNIT
DC or peak AC load voltage*	$V_L$	45	V
Load current (DC only)*	$I_L$	1000	mA

Table 2. Absolute Maximum Ratings

## 7. PINOUT

### 7.1. DUT connector Interface



DUT		TO BK+		TO BK-		STIM			
PIN	NAME	Re- lay	Code	Re- lay	Code	PIN	NAME	Re- lay	Code
6	DUT_1	U1	0x01h	U19	0x19h	30	STIM_1	U38	0x31h
22	DUT_2	U2	0x02h	U20	0x1Ah	47	STIM_2	U39	0x32h
21	DUT_3	U3	0x03h	U23	0x1Bh	46	STIM_3	U40	0x33h
38	DUT_4	U4	0x04h	U24	0x1Ch	13	STIM_4	U41	0x34h
37	DUT_5	U5	0x05h	U25	0x1Dh	45	STIM_5	U42	0x35h
5	DUT_6	U6	0x06h	U26	0x1Eh	29	STIM_6	U43	0x36h
4	DUT_7	U7	0x07h	U28	0x1Fh	12	STIM_7	U44	0x37h
20	DUT_8	U8	0x08h	U29	0x20h	28	STIM_8	U45	0x38h
19	DUT_9	U9	0x09h	U30	0x21h	27	STIM_9	U46	0x39h
36	DUT_10	U10	0x0Ah	U31	0x22h	44	STIM_10	U47	0x3Ah
34	DUT_11	U11	0x0Bh	U32	0x23h	10	STIM_11	U65	0x3Bh
18	DUT_12	U15	0x0Ch	U34	0x24h	11	STIM_12	U66	0x3Ch
3	DUT_13	U16	0x0Dh	U35	0x25h	26	STIM_13	U67	0x3Dh
35	DUT_14	U17	0x0Eh	U36	0x26h	43	STIM_14	U68	0x3Eh
25	DUT_15	U18	0x0Fh	U37	0x27h	15	STIM_15	U69	0x3Fh
42	DUT_16	U48	0x10h	U60	0x28h	14	STIM_16	U70	0x40h
41	DUT_17	U49	0x11h	U61	0x29h	48	STIM_17	U71	0x41h
9	DUT_18	U50	0x12h	U62	0x2Ah	31	STIM_18	U72	0x42h
24	DUT_19	U58	0x13h	U63	0x2Bh	33	STIM_19	U73	0x43h
8	DUT_20	U59	0x14h	U64	0x2Ch	50	STIM_20	U74	0x44h
7	DUT_21	U75	0x15h	U80	0x2Dh	49	STIM_21	U85	0x45h
39	DUT_22	U76	0x16h	U81	0x2Eh	17	STIM_22	U86	0x46h
40	DUT_23	U77	0x17h	U82	0x2Fh	16	STIM_23	U87	0x47h
23	DUT_24	U78	0x18h	U83	0x30h	32	STIM_24	U88	0x48h

## 7.2. Backplane Connector

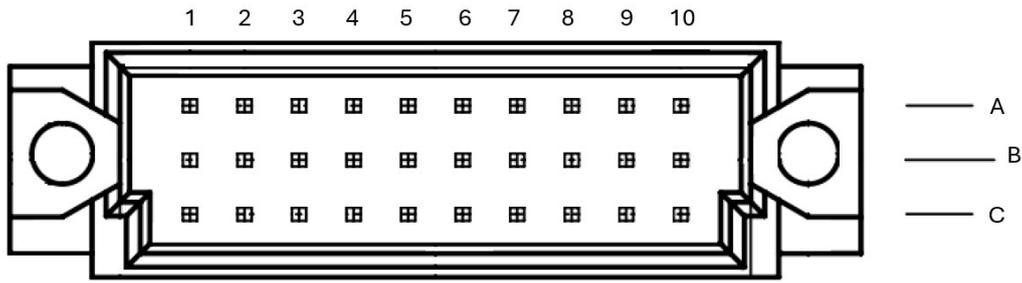


Figure 2. Backplane Connector Pinout

PINS	NAME	DESCRIPTION
A1	CAN_L	CAN Lo bus line
B1	CAN_H	CAN Hi bus line
C1, A2, B2, A9, B9, C9, A10, B10, C10	-	Not connected
C2	SYNC	Reserved for future use
A3, B3, C3	GND_control	Ground reference for the PS of the Lynx control boards
A4, B4, C4	PS_control	Power supply for the Lynx control boards
A5, B5, C5	GND_DUT	Ground reference for the PS of the DUT
A6, B6, C6	PS_DUT	Power supply for the DUT
A7, B7, C7	BK-	Connection to the measurement equipment, positive
A8, B8, C8	BK+	Connection to the measurement equipment, negative

## 8. DID LIST

### 8.1. Configuration

For all the frames, these parameters must be considered:

- **CAN Baudrate** = 500 Kbps
- **CAN-ID**
  - TX Frame = 1000 + (SERIAL % 100)
  - RX Frame = 1100 + (SERIAL % 100)
- **CAN FRAME LENGTH** = 0x08

### 8.2. Read Data Diagnostic List

DID	Name
0xF702	Get Relays State
0xF195	Read Software Version
0xF193	Read Hardware Version
0xF182	Read Product ID
0xF181	Read Product Name
0xF18C	Read Serial Number

### 8.3. GET RELAYS STATE

#### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
03	22	F7	02	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x22 = Read Data Service
- DID (2 bytes): 0xF702 = Get Relays State

#### RECEIVED FRAME

SF INDEX	LENGTH	SERVICE	DID		BUFOUT		
10	1C	62	F7	02	RS (0-3)	RS (4-7)	RS (8-11)
SF INDEX	BUFOUT						
21	RS (12-15)	RS (16-19)	RS (20-23)	RS (24-27)	RS (28-31)	RS (32-35)	RS (36-39)
SF INDEX	BUFOUT						
22	RS (40-43)	RS (44-47)	RS (48-51)	RS (52-55)	RS (56-59)	RS (60-63)	RS (64-67)
SF INDEX	BUFOUT						
23	RS (68-71)	RS (72-75)	RS (76-79)	RS (80-83)	RS (84-87)	RS (88-91)	RS (92-95)
SF INDEX	BUFOUT						
24	RS (96-99)	-	-	-	-	-	-

- SF INDEX (4 bytes): Sequential Frames Index
  - 0x10 = First frame index
  - 0x21 = Second frame index
  - 0x22 = Third frame index
  - 0x23 = Fourth frame index
  - 0x24 = Fifth frame index
- LENGTH (1 byte): 0x1C = 28 bytes frame length
- SERVICE (1 byte): 0x62 = Read Data Service OK
- DID (2 bytes): 0xF702 = Get Relays State
- BUFOUT (25 bytes): Relay State (RS, 2 bits per relay)
  - 0b11 = OFF
  - 0b10 = ON

#### 8.3.1. Read Software Version

#### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
03	22	F1	95	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x22 = Read Data Service
- DID (2 bytes): 0xF195 = Read Software Version

**RECEIVED FRAME**

LENGTH	SERVICE	DID		BUFOUT			
07	62	F1	95	SwVer [0]	SwVer [1]	SwVer [2]	SwVer [3]

- LENGTH (1 byte): 0x07 = 7 bytes frame length
- SERVICE (1 byte): 0x62 = Read Data Service OK
- DID (2 bytes): 0xF195 = Read Software Version
- 0 (4 bytes): Software Version

**8.3.2. Read Hardware Version**
**SENT FRAME**

LENGTH	SERVICE	DID		BUFIN			
03	22	F1	93	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x22 = Read Data Service
- DID (2 bytes): 0xF193 = Read Hardware Version

**RECEIVED FRAME**

LENGTH	SERVICE	DID		BUFOUT			
07	62	F1	93	HwVer [0]	HwVer [1]	HwVer [2]	HwVer [3]

- LENGTH (1 byte): 0x07 = 7 bytes frame length
- SERVICE (1 byte): 0x62 = Read Data Service OK
- DID (2 bytes): 0xF193 = Read Hardware Version
- BUFOUT (4 bytes): Hardware Version

**8.3.3. Read Product Id**
**SENT FRAME**

LENGTH	SERVICE	DID		BUFIN			
03	22	F1	82	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x22 = Read Data Service
- DID (2 bytes): 0xF182 = Read Product ID

**RECEIVED FRAME**

LENGTH	SERVICE	DID		BUFOUT			
07	62	F1	82	ProdID[0]	ProdID[1]	ProdID[2]	ProdID[3]

- LENGTH (1 byte): 0x07 = 7 bytes frame length
- SERVICE (1 byte): 0x62 = Read Data Service OK
- DID (2 bytes): 0xF182 = Read Product ID
- BUFOUT (4 bytes): Product ID

### 8.3.4. Read Product Name

#### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
03	22	F1	81	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x22 = Read Data Service
- DID (2 bytes): 0xF181 = Read Product Name

#### RECEIVED FRAME

SF INDEX	LENGTH	SERVICE	DID		BUFOUT		
10	17	62	F1	81	char1	char2	char3
SF INDEX	BUFOUT						
21	char4	char5	char6	char7	char8	char9	char10
SF INDEX	BUFOUT						
22	char11	char12	char13	char14	char15	char16	char17
SF INDEX	BUFOUT						
23	char18	char19	char20	-	-	-	-

- SF INDEX (4 bytes): Sequential Frames Index
  - 0x10 = First frame index
  - 0x21 = Second frame index
  - 0x22 = Third frame index
  - 0x23 = Fourth frame index
- LENGTH (1 byte): 0x17 = 23 bytes frame length
- SERVICE (1 byte): 0x62 = Read Data Service OK
- DID (2 bytes): 0xF181 = Read Product Name
- BUFOUT (25 bytes): Product Name

### 8.3.5. Read Serial Number

#### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
03	22	F1	8C	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x22 = Read Data Service
- DID (2 bytes): 0xF18C = Read Serial Number

#### RECEIVED FRAME

LENGTH	SERVICE	DID		BUFOUT			
07	62	F1	8C	SerNum[0]	SerNum[1]	SerNum[2]	SerNum[3]

- LENGTH (1 byte): 0x07 = 7 bytes frame length
- SERVICE (1 byte): 0x62 = Read Data Service OK
- DID (2 bytes): 0xF18C = Read Serial Number
- BUFOUT (4 bytes): Serial Number

#### 8.4. Write Data Diagnostic List

DID	NAME
0xF701	Enable Relay
0xF700	Disable Relay
0xF703	Disable All Relays
0xF709	Enable / Disable Watchdog
0xF70A	Refresh Watchdog
0xF705	Set Serial Number
0xF606	Set Product Name
0xF704	Set Hardware Version
0xF601	CAN enable configuration of the serial
0xF603	CAN configuration of the serial

Table 3. Write Data DID List

##### 8.4.1. Enable Relay

###### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
04	2E	F7	01	RelayIndex	-	-	-

- LENGTH (1 byte): 0x04 = 4 bytes frame length
- SERVICE (1 byte): 0x2E = Write Data Service
- Did (2 BYTES): 0xF701 = Enable Relay
- BUFIN (1 bytes): 0xF18C = RelayIndex in Hex Format (from 0x01-0x64)

###### RECEIVED FRAME

LENGTH	SERVICE	DID		BUFOUT			
03	6E	F7	01	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x6E = Write Data Service OK
- DID (2 bytes): 0xF701 = Enable Relay

##### 8.4.2. Disable Relay

###### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
04	2E	F7	00	RelayIndex	-	-	-

- LENGTH (1 byte): 0x04 = 4 bytes frame length
- SERVICE (1 byte): 0x2E = Write Data Service
- DID (2 bytes): 0xF700 = Disable Relay
- BUFIN (1 byte): RelayIndex in Hex Format (from 0x01-0x64)

###### RECEIVED FRAME

LENGTH	SERVICE	DID		BUFOUT			
03	6E	F7	00	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x6E = Write Data Service OK
- DID (2 bytes): 0xF700 = Disable Relay

### 8.4.3. Disable All Relays

#### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
03	2E	F7	03	RelayIndex	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x2E = Write Data Service
- DID (2 bytes): 0xF703 = Disable All Relays

#### RECEIVED FRAME

LENGTH	SERVICE	DID		BUFOUT			
03	6E	F7	03	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x6E = Write Data Service OK
- DID (2 bytes): 0xF703 = Disable All Relays

### 8.4.4. Enable/Disable Watchdog

#### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
04	2E	F7	09	En/Dis	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x2E = Write Data Service
- DID (2 bytes): 0xF709 = Enable/Disable Watchdog
- BUFIN (1 byte):
  - 0x00 = Disable Watchdog
  - 0x01 = Enable Watchdog

#### RECEIVED FRAME

LENGTH	SERVICE	DID		BUFOUT			
03	6E	F7	09	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x6E = Write Data Service OK
- DID (2 bytes): 0xF709 = Enable/Disable Watchdog

### 8.4.5. Refresh Watchdog

#### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
04	2E	F7	0A	AA	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x2E = Write Data Service
- DID (2 bytes): 0xF70A = Refresh Watchdog
- BUFIN (1 byte): 0xAA = Key

**RECEIVED FRAME**

LENGTH	SERVICE	DID		BUFOUT			
03	6E	F7	0A	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x6E = Write Data Service OK
- DID (2 bytes): 0xF70A = Refresh Watchdog

**8.4.6. Set Serial Number**
**SENT FRAME**

SF INDEX	LENGTH	SERVICE	DID		BUFIN		
10	0B	2E	F7	05	Pw[31:24]	Pw[23:16]	Pw[15:8]
SF INDEX	BUFIN						
21	Pw[7:0]	Ser[31:24]	Ser[23:16]	Ser[15:8]	Ser[7:0]	-	-

- SF INDEX (2 bytes): Sequential Frames Index
  - 0x10 = First frame index
  - 0x21 = Second frame index
- LENGTH (1 byte): 0x0B = 11 bytes frame length
- SERVICE (1 byte): 0x2E = Write Data Service
- DID (2 bytes): 0xF705 = Set Serial Number
- BUFIN (8 bytes):
  - Password (4 bytes) (always 0xAA 0x55 0x99 0x66 )
  - Serial Number (4 bytes)

**RECEIVED FRAME**

LENGTH	SERVICE	DID		BUFOUT			
03	6E	F7	05	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x6E = Write Data Service OK
- DID (2 bytes): 0xF705 = Set Serial Number

**8.4.7. Set Product Name**
**SENT FRAME**

SF INDEX	LENGTH	SERVICE	DID		BUFIN		
10	17	2E	F6	06	PN1	PN2	PN3
SF INDEX	BUFIN						
21	PN4	PN5	PN6	PN7	PN8	PN9	PN10
SF INDEX	BUFIN						
22	PN11	PN12	PN13	PN14	PN15	PN16	PN17
SF INDEX	BUFIN						
23	PN18	PN19	PN20	-	-	-	-

- SF INDEX (2 bytes): Sequential Frames Index
  - 0x10 = First frame index
  - 0x21 = Second frame index
  - 0x22 = Third frame index
  - 0x23 = Fourth frame index
- LENGTH (1 byte): 0x17 = 23 bytes frame length
- SERVICE (1 byte): 0x2E = Write Data Service

- DID (2 bytes): 0xF606 = Set Product Name
- BUFIN (20 bytes): Product Name (20 characters)

#### RECEIVED FRAME

LENGTH	SERVICE	DID		BUFOUT			
03	6E	F6	06	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x6E = Write Data Service OK
- DID (2 bytes): 0xF606 = Set Product Name

#### 8.4.8. Set Hardware Version

##### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
04	2E	F7	04	Hw[31:24]	Hw[23:16]	Hw[15:8]	Hw[7:0]

- LENGTH (1 byte): 0x04 = 4 bytes frame length
- SERVICE (1 byte): 0x2E = Write Data Service
- DID (2 bytes): 0xF704 = Set Hardware Version
- BUFIN (4 byte): Hardware Version

#### RECEIVED FRAME

LENGTH	SERVICE	DID		BUFOUT			
03	6E	F7	04	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x6E = Write Data Service OK
- DID (2 bytes): 0xF704 = Set Hardware Version

#### 8.4.9. Can Enable Configuration Of The Serial

##### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
03	2E	F6	01	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x2E = Write Data Service
- DID (2 bytes): 0xF601 = CAN enable configuration of the serial

#### RECEIVED FRAME

LENGTH	SERVICE	DID		BUFOUT			
03	6E	F6	01	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x6E = Write Data Service OK
- DID (2 bytes): 0xF601 = CAN enable configuration of the serial

#### 8.4.10. Can Configuration Of The Serial

##### SENT FRAME

LENGTH	SERVICE	DID		BUFIN			
03	2E	F6	03	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length

- SERVICE (1 byte): 0x2E = Write Data Service
- DID (2 bytes): 0xF603 = CAN configuration of the serial

## RECEIVED FRAME

LENGTH	SERVICE	DID		BUFOUT			
03	6E	F6	03	-	-	-	-

- LENGTH (1 byte): 0x03 = 3 bytes frame length
- SERVICE (1 byte): 0x6E = Write Data Service OK
- DID (2 bytes): 0xF603 = CAN enable of the serial

## 9. OTHER RELEVANT CHARACTERISTICS

### 9.1. Watchdog functionality

The LynxICT includes a configurable watchdog mechanism to ensure safe operation. When enabled, the watchdog requires the user to send refresh frames within the configured timeout period.

#### 9.1.1. Fall-safe behavior

- If no refresh frame is received within 10 seconds, all relays are automatically deactivated.
- The blue status LED turns off, and the red error LED turns on.
- A diagnostic flag (Timeout Expired) is raised.

### 9.2. Control via CAN (UDS service 0x2E)

DID (hex)	Command	Description	Example
0xF709	Enable / Disable Watchdog	Activates or deactivates the watchdog function.	04 2E F7 09 01 → Enable 04 2E F7 09 00 → Disable
0xF70A	Refresh Watchdog	Must be sent periodically (≤ 10 s).	04 2E F7 0A AA

#### 9.2.1. Operation summary

- **Enable** the watchdog (DID 0xF709).
- Send **refresh frames** (DID 0xF70A) at intervals shorter than 10 seconds.
- If refresh is not received within the timeout, the system enters the fail-safe state.

This mechanism prevents uncontrolled relay activation in case of communication loss, software malfunction, or operator inaction.

### 9.3. LEDs Manage

LynxICT includes three status LEDs that provide quick visual feedback on system operation:

COLOR	INDICATES
Blue	Power ON (board ready)
Green	Reception of CAN frames (blinks 3 times per frame)
Red	Watchdog timeout (fail-safe state)

#### 9.4. Relay Command Behavior

After receiving relay control commands, the board executes the following sequence:

1. Load commands
  - The requested relay switching operations are performed.
2. Reset watchdog
  - The watchdog timer is refreshed to avoid timeout.
3. Update LEDs
  - Current status is indicated via LEDs:
    - Red OFF
    - Blue ON

#### 9.5. Can Identifier Configuration

Each LynxICT device generates its communication identifiers (CAN IDs) based on the assigned Serial Number. ID generation rule:

- **Transmit ID (Tx):**  $1000 + (\text{Serial Number} \% 100)$
- **Receive ID (Rx):**  $1000 + (\text{Serial Number} \% 100)$

This mechanism guarantees that each device has a unique CAN identifier set, avoiding collisions when multiple boards are used in the same system.

Whenever the **Serial Number** is changed, the CAN ID configuration must be updated accordingly and stored in non-volatile memory (NVM).

---

#### Rev. A — Document

Information furnished by **Generation RFID** is believed to be accurate and reliable. However, **Generation RFID** assumes no responsibility for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications are subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of **Generation RFID**.

Trademarks and registered trademarks are the property of their respective owners.

#### Generation RFID, S.L.

Carrer Camí de Valls, 43204 Reus, Catalonia, Spain

© [2025] Generation RFID. All rights reserved.

Technical Support: [www.generationrfid.com](http://www.generationrfid.com)