



Jasper TC570 1185GRE WIN10 GPIO

User Manual with API Specification

Version 1.0

Date	Comment
12/12/2025	Initial Release

FOR TECHNICAL SUPPORT
CONTACT: support@diamondsystems.com

© Copyright 2025
Diamond Systems Corporation
www.diamondsystems.com

CONTENTS

OVERVIEW	3
INSTALLATION	3
STEPS TO RUN DIO Application	3
API Description	8
1. Functional Description.....	8
2. Function Specifications	8
3. Error Handling.....	12

OVERVIEW

This software is for the Jasper Congatec TC570 1185GRE GPIO capability. It allows for controlling and monitoring digital I/O through bit-level and byte-level reading and writing operations.

INSTALLATION

To access the GPIO, users need to fulfill the following requirements

1. Use a Windows 10/11 operating system.
2. Download the package from the official website.

STEPS TO RUN DIO Application

Step 1:

- 1 Download the ZIP package containing the DIO application and API libraries.
- 2 Right-click the ZIP file → **Extract All...** or use a tool like WinRAR/7-Zip.
- 3 Use the cd command to change into the extracted folder.
 - JasperGPIO.exe (the executable application)
 - Supporting Lib Files
 - Documentation and sample programs

Step 2: Run the Application

- 1 Start the application by command: **JaperGPIO.exe**
- 2 This launches the GPIO control program.
- 3 The program will initialize the **CGOS library** and open the GPIO board handle.
- 4 The user can see a menu or prompt for GPIO operations.

```
JASPER 1185GRE GPIO FUNCTIONS DEMO

GPIO INPUT AND OUTPUT: Main Menu
#####
Please Select an option from below!
1) Set GPIO Configuration
2) Set GPIO BIT Level Configuration
3) Read Byte from port
4) Read all bits from port
5) Write Byte to port
6) Write Bit to a port
7) Port loopback 0 1 255 read/write repetitive
q) Quit Program
#####
```

Step 3: Set GPIO Byte Configuration

1. Allow the user to configure all 8 GPIO pins at once as input or output.
 - Example:
 - 0xFF → all pins set as **output**
 - 0x00 → all pins set as **input**
2. Internally, this calls Jasper1185GREGPIOConfig().

```
JASPER 1185GRE GPIO FUNCTIONS DEMO

GPIO INPUT AND OUTPUT: Main Menu
#####
Please Select an option from below!
1) Set GPIO Configuration
2) Set GPIO BIT Level Configuration
3) Read Byte from port
4) Read all bits from port
5) Write Byte to port
6) Write Bit to a port
7) Port loopback 0 1 255 read/write repetitive
q) Quit Program
#####
1
Enter the GPIO Byte Configuration (1=input, 0=output):
```

Step 4: Set GPIO Bit Configuration

1. Configure individual pins (bit-level).
 - Example:
 - Bit 0 = Input
 - Bit 1 = Output
2. Internally, this calls Jasper1185GREGPIOConfig().

```
JASPER 1185GRE GPIO FUNCTIONS DEMO

GPIO INPUT AND OUTPUT: Main Menu
#####
Please Select an option from below!
1) Set GPIO Configuration
2) Set GPIO BIT Level Configuration
3) Read Byte from port
4) Read all bits from port
5) Write Byte to port
6) Write Bit to a port
7) Port loopback 0 1 255 read/write repetitive
q) Quit Program
#####
2

Info Mapping: Configure GPIO (0 = Output, 1 = Input)
-----
Enter direction for Bit 0 (0/1): 0
Enter direction for Bit 1 (0/1): 0
Enter direction for Bit 2 (0/1): 0
Enter direction for Bit 3 (0/1): 0
Enter direction for Bit 4 (0/1): 1
Enter direction for Bit 5 (0/1): 1
Enter direction for Bit 6 (0/1): 1
Enter direction for Bit 7 (0/1): 1
#####
```

Step 5: Read a Byte from the port

- 1 Reads the current state of all 8 pins at once.
 Example output: 0x3C → binary 00111100 means pins 2–5 are high, others low.
- 2 Internally, this calls Jasper1185GREGPIOInputByte().

```
JASPER 1185GRE GPIO FUNCTIONS DEMO

GPIO INPUT AND OUTPUT: Main Menu
#####
Please Select an option from below!
1) Set GPIO Configuration
2) Set GPIO BIT Level Configuration
3) Read Byte from port
4) Read all bits from port
5) Write Byte to port
6) Write Bit to a port
7) Port loopback 0 1Çô 255 read/write repetitive
q) Quit Program
#####
3
Byte value received from GPIO = 0x0
#####
```

Step 6: Read all bits from the port

Reads the state of each pin separately.

Example:

- Bit 0 = High
- Bit 1 = Low

Internally, this uses Jasper1185GREGPIOInputBit().

```
JASPER 1185GRE GPIO FUNCTIONS DEMO

GPIO INPUT AND OUTPUT: Main Menu
#####
Please Select an option from below!
1) Set GPIO Configuration
2) Set GPIO BIT Level Configuration
3) Read Byte from port
4) Read all bits from port
5) Write Byte to port
6) Write Bit to a port
7) Port loopback 0 1Çô 255 read/write repetitive
q) Quit Program
#####
4
The GPIO each bit values are
-----
BitNo: 7      6      5      4      3      2      1      0
-----
Value: 0      0      0      0      0      0      0      0
-----
#####
```

Step 7: Write a Byte to the port.

Allow the user to write a byte value to a GPIO. This effectively sets the state of 8 digital lines simultaneously.

```
JASPER 1185GRE GPIO FUNCTIONS DEMO

GPIO INPUT AND OUTPUT: Main Menu
#####
Please Select an option from below!
1) Set GPIO Configuration
2) Set GPIO BIT Level Configuration
3) Read Byte from port
4) Read all bits from port
5) Write Byte to port
6) Write Bit to a port
7) Port loopback 0-255 read/write repetitive
q) Quit Program
#####
5
Enter value 0-255 or q to quit:85
The Byte value 85 is sent to GPIO
```

Step 8: Write a Bit to a port

1. Sets a single pin high or low.
Example:
 - Bit 3 = High → turns on a connected LED.
 - Bit 4 = Low → turns off a relay.
2. Internally, this uses Jasper1185GREGPIOOutputBit().

```
JASPER 1185GRE GPIO FUNCTIONS DEMO

GPIO INPUT AND OUTPUT: Main Menu
#####
Please Select an option from below!
1) Set GPIO Configuration
2) Set GPIO BIT Level Configuration
3) Read Byte from port
4) Read all bits from port
5) Write Byte to port
6) Write Bit to a port
7) Port loopback 0-255 read/write repetitive
q) Quit Program
#####
6
Enter Bit (0-7):or q to quit :5
Enter Bit value (0-1):1
```


Step 9: Port loopback

1. Writes a known pattern to output pins and reads it back from input pins.
2. Confirms wiring and GPIO functionality.

Example: Write 0x55 → expect to read 0x55 if loopback is correct.

```
JASPER 1185GRE GPIO FUNCTIONS DEMO

GPIO INPUT AND OUTPUT: Main Menu
#####
Please Select an option from below!
1) Set GPIO Configuration
2) Set GPIO BIT Level Configuration
3) Read Byte from port
4) Read all bits from port
5) Write Byte to port
6) Write Bit to a port
7) Port loopback 0 ΓÇô 255 read/write repetitive
q) Quit Program
#####
7
Do you want to enable the Verbose Mode 0-Enable 1-Disable
1
Enter GPIO Bits to Write
0 - BIT 0 to BIT 3 [Output] BIT 4 to BIT 7 [Input];
1 - BIT 0 to BIT 3 [Input] & BIT 4 to BIT 7 [Output]
-1 go to Main menu: 0

Values written : 16 and Errors: 0
```

Step 10: Exit the Application

Allow the user to exit the application.

```
JASPER 1185GRE GPIO FUNCTIONS DEMO

GPIO INPUT AND OUTPUT: Main Menu
#####
Please Select an option from below!
1) Set GPIO Configuration
2) Set GPIO BIT Level Configuration
3) Read Byte from port
4) Read all bits from port
5) Write Byte to port
6) Write Bit to a port
7) Port loopback 0 ΓÇô 255 read/write repetitive
q) Quit Program
#####
```

API Description

1. Functional Description

Function Name	Description
Jasper1185GREGPIOInitBoard()	Initializes CGOS library and opens the CGOS board handle.
Jasper1185GREGPIOFreeBoard()	Releases CGOS board resources and uninitializes the library.
Jasper1185GREGPIOConfig()	Configures the direction (Input/Output) of GPIO port.
Jasper1185GREGPIOOutputBit()	Writes logic value (High/Low) to a specific GPIO bit.
Jasper1185GREGPIOInputBit()	Reads the logic state (High/Low) from a specific GPIO bit.
Jasper1185GREGPIOOutputByte()	Writes a value to a GPIO pins.
Jasper1185GREGPIOInputByte()	Reads the value from a GPIO pins.

2. Function Specifications

2.1 Jasper1185GREGPIOConfig

Prototype: BYTE Jasper1185GREGPIOConfig(UINT dir);

Description: Configures the direction of the specified GPIO port as Input or Output.

1. Valid dir range: 0x00 to 0xFF
2. Function internally calls CgosIOSetDirection().

Parameters:

Name	Type	Description
dir	UINT	8-bit direction mask (0 = Input, 1 = Output).

Return Values:

Value	Description
DE_NONE	Operation successful.
DE_INVALID_PARM	Invalid parameter value.
HW_FAILURE	Failed to set the GPIO direction.

2.2 Jasper1185GREGPIOOutputBit

Prototype: BYTE Jasper1185GREGPIOOutputBit(UINT bit, UINT value);

Description: Sets a specified GPIO bit to High (1) or Low (0).

1. Reads the current GPIO port state using CgosIORead().
2. Modifies only the selected bit according to value.
3. Writes the new port state using CgosIOWrite()

Parameters:

Name	Type	Description
bit	UINT	Bit index within the port (0–7).
value	UINT	Value to write: 0 = Low 1 = High

Return Values:

Value	Description
DE_NONE	Operation successful.
DE_INVALID_PARM	Invalid parameter value.
HW_FAILURE	Failed to set the GPIO direction.

2.3 Jasper1185GREGPIOInputBit

Prototype: BYTE Jasper1185GREGPIOInputBit(UINT bit, UINT* value);

Description: Reads the current state (High or Low) of a specified GPIO bit.

1. The function reads the entire 8-bit port and masks the required bit.
2. The returned bit value constants:
 - Jasper_GPIO_BIT_HIGH = 1
 - Jasper_GPIO_BIT_LOW = 0

Parameters:

Name	Type	Description
bit	UINT	Bit index within the port (0–7).
value	UINT	Store the read bit value (0 = Low, 1 = High).

Return Values:

Value	Description
DE_NONE	Operation successful.
DE_INVALID_PARM	Invalid parameter value.
HW_FAILURE	Failed to set the GPIO direction.

2.4 Jasper1185GREGPIOOutputByte

Prototype: BYTE Jasper1185GREGPIOOutputByte(UINT value);

Description: Sets a specified GPIO byte to a value.

1. Reads the current GPIO port state using CgosIORead().
2. Modifies only the selected bit according to value.
3. Writes the new port state using CgosIOWrite()

Parameters:

Name	Type	Description
value	UINT	Value to write: 0 – 255.

Return Values:

Value	Description
DE_NONE	Operation successful.
DE_INVALID_PARM	Invalid parameter value.
HW_FAILURE	Failed to set the GPIO direction.

2.5 Jasper1185GREGPIOInputByte

Prototype: BYTE Jasper1185GREGPIOInputByte(UINT* value);

Description: Reads the current state byte value from the GPIO pins.

1. The function reads the entire 8-bit port and masks the required bit.
2. The returned bit value constants:
 - Jasper_GPIO_BIT_HIGH = 1
 - Jasper_GPIO_BIT_LOW = 0

Parameters:

Name	Type	Description
value	UINT	Store the read byte value.

Return Values:

Value	Description
DE_NONE	Operation successful.
DE_INVALID_PARM	Invalid parameter value.
HW_FAILURE	Failed to set the GPIO direction.

2.6 Jasper1185GREGPIOInitBoard

Prototype: BYTE Jasper1185GREGPIOInitBoard(void);

Description: Initializes the CGOS library and opens the first available CGOS board for GPIO access.

1. Initialize the CGOS library using CgosLibInitialize().
2. Verify the library's availability using CgosLibIsAvailable().
3. Open the CGOS board with index 0 using CgosBoardOpen().

Return Values:

Value	Description
DE_NONE	Initialization successful.
HW_FAILURE	Failed to initialize or open the CGOS board.

2.7 Jasper1185GREGPIOFreeBoard

Prototype: BYTE Jasper1185GREGPIOFreeBoard(void);

Description: Closes the CGOS board and releases all library resources.

1. Close it using CgosBoardClose().
2. Uninitialize the CGOS library using CgosLibUninitialize().

Return Values:

Value	Description
DE_NONE	Board successfully released

3. Error Handling

Error Code	Description
DE_NONE	No error. Function executed successfully.
DE_INVALID_PARM	Invalid function argument.
HW_FAILURE	Hardware access or configuration error.

More Information

For more information, please visit <https://www.diamondsystems.com/>

For Technical Support requests can be directed to:

www.diamondsystems.com/support/request or
support@diamondsystems.com

Thank you for choosing Diamond systems.