1 Installation Instructions

The EzCoding software package can be used directly after decompression. If an antivirus software reports an error, you need to add the program to the trusted list. For first-time use, systems below Windows 8 are required to install the driver setup program in the Driver directory of the software.

Due to driver signature issues existing in CodingBox devices manufactured before October 2019, the newly released USB driver has been revised. The EzCoding V1011 version no longer supports the old version driver. After connecting the CodingBox to the USB port, refer to the number displayed on the nixie tube, and use the FWUpdateTools on the official website (www.sfpcodingbox.com) to upgrade the firmware.

Table 1 Nixie Tube Status Display When CodingBox Starts

Nixie Tube Display	Firmware Upgrade Recommendation	Firmware Area	Applicable Driver
'0'	No upgrade required	Application Area	WinUSB
'8'	Upgrade with FWUpdateTools	Application Area	Box Driver
Blinking '1'	Get support at support@sfpcodingbox.com	Bootloader Area	WinUSB
Blinking '-'	Get support at support@sfpcodingbox.com	Bootloader Area	Box Driver
Off	Firmware may be damaged; return for repair	BL update failure / Software shutdown	WinUSB

1.1 Driver Installation

Open EzCoding-Winusb-Driver.exe in the Driver directory. If the driver installation wizard is not automatically triggered, right-click the driver file and select Run as administrator.

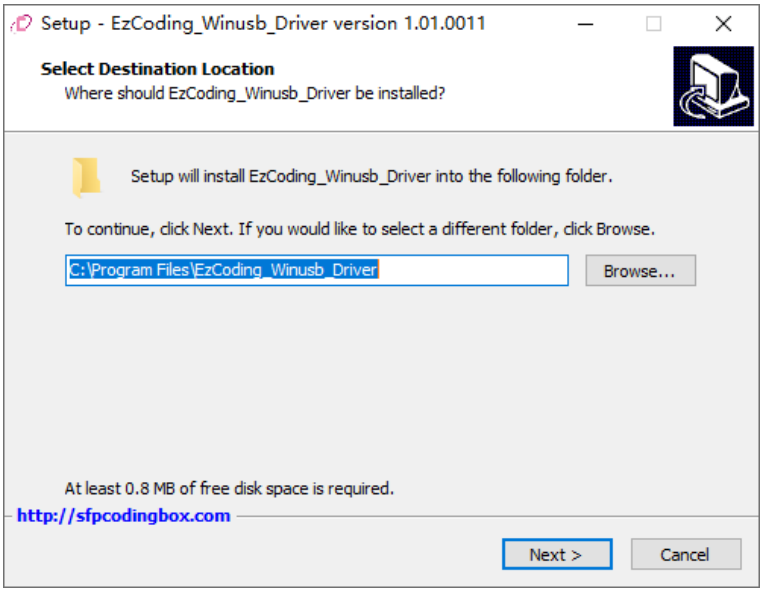

Figure 1 Driver Installation Step 1

Select Next to proceed with the installation.

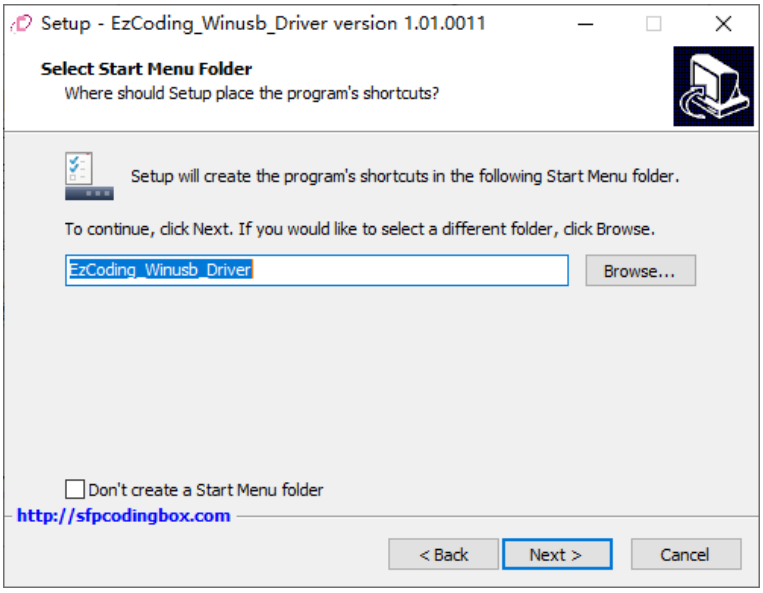

Figure 2 Driver Installation Step 2

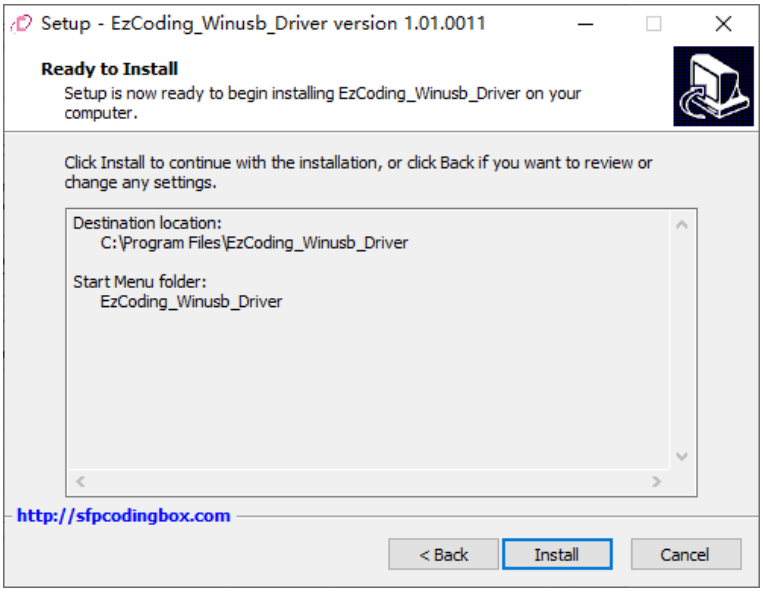

Figure 3 Driver Installation Step 3

Select Install to complete the installation, and choose Always install this driver software when prompted.

Figure 4 Driver Installation Signature Prompt

Open Device Manager, connect the device, and check if the driver is loaded successfully.

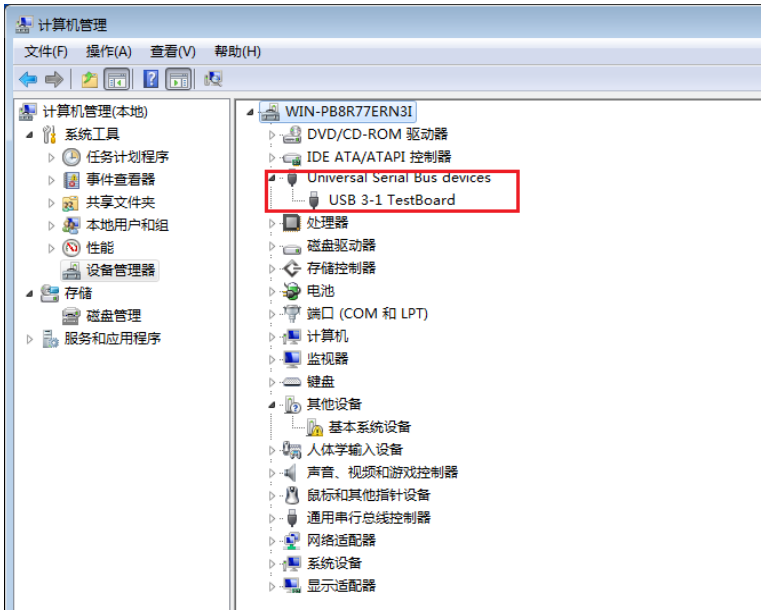

Figure 5 Device Manager

2 Basic Functions

2.1 Interface Functional Areas

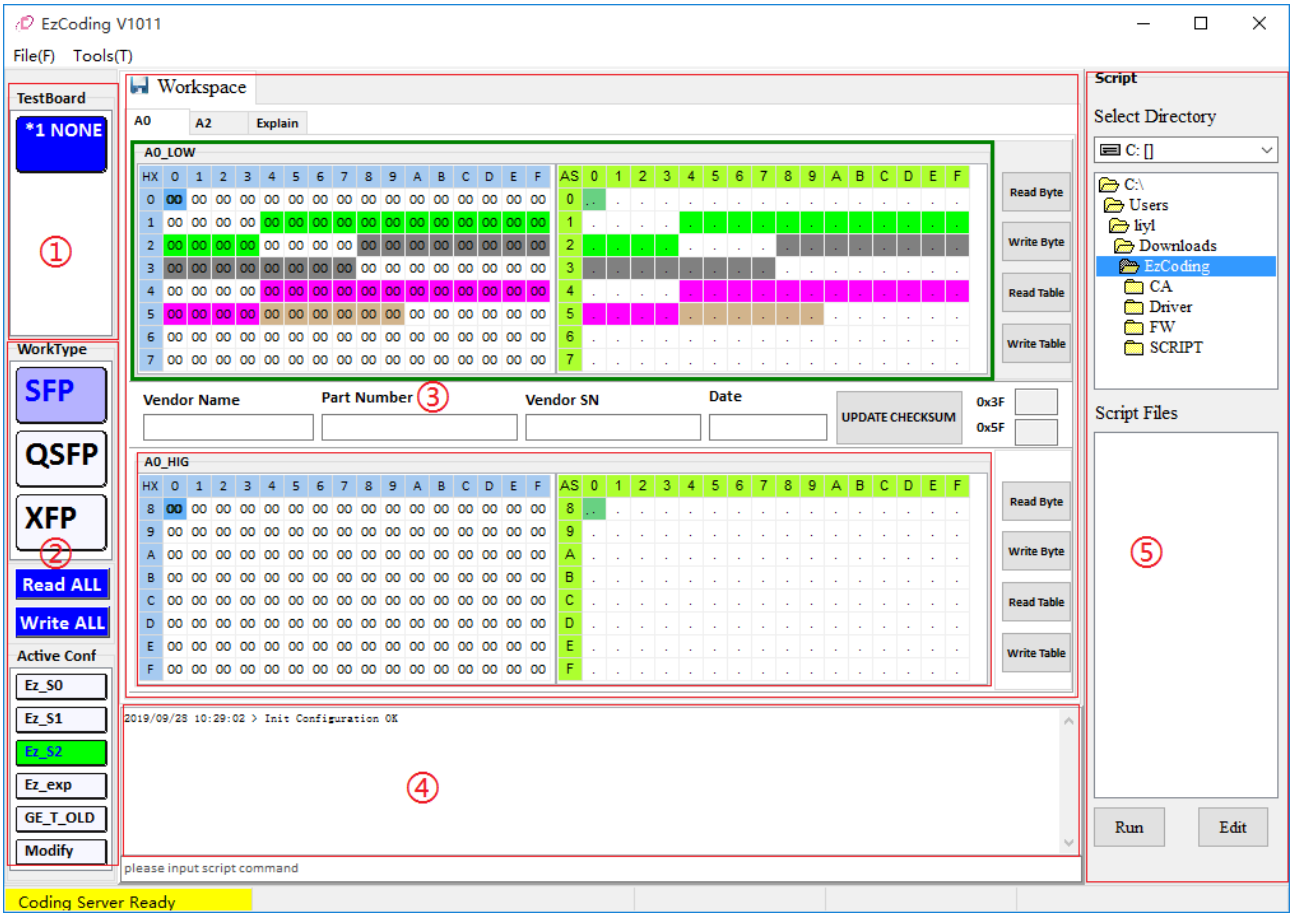

Figure 6 Interface Functional Areas

(1) CodingBox Selection Detection Area: Displays the number of currently connected CodingBox devices and the modules inserted into the selected board.

(2) Workspace Configuration Area: Select and configure different module types. When writing to or reading from a module, the selected type must match the module type detected inArea (1). The configurations for Activeconf, readALL, and WriteALL need to be set in Tools → Options.

(3) Workspace Display Area: A table area defined according to the module type. Each table stores 128 bytes of data, including the module's Seriesid definition and table selection area. The buttons on the right can perform read/write operations on specific bytes or the entire table data.

(4) Log Area: Records operation prompts and data.

(5) Script Functional Area: Default scripts are stored in the SCRIPT directory of the program. Users can select and run saved scripts.

2.2 EzCoding Data Structure Model

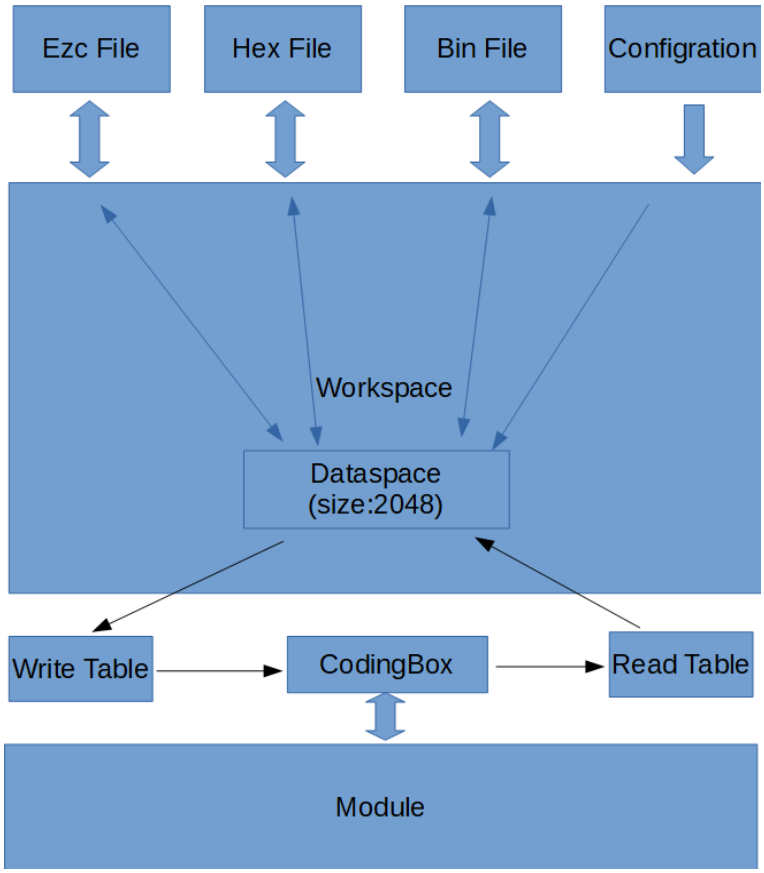

Figure 7 EzCoding Data Structure Model

(1) Data processed through file operations is stored in the Dataspace of the Workspace.

(2) When executing WriteTable orReadTable, the data in the Workspace interacts with the internal data tables of the module via the CodingBox.

2.3 Opening and Saving Encoding Files

The File menu provides options for opening and saving .bin, .hex, and .ezc files. For encoding files with 128 bytes of data, you must select the target data area for saving or reading in Area (3).

2.3.1 Opening and Saving 128-Byte .bin Files

The File menu options are available.

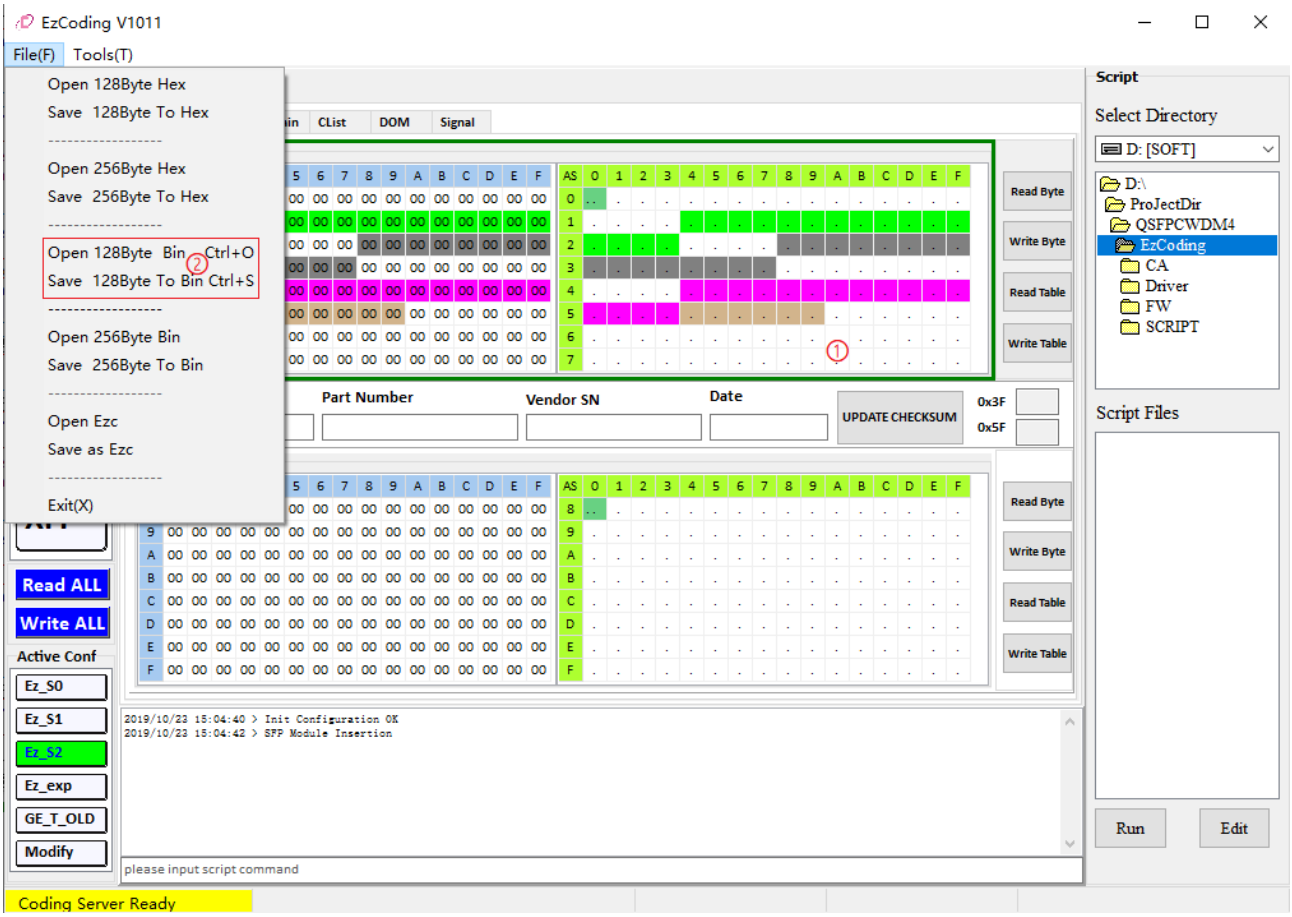

Figure 8 128-Byte .bin File List

(1) Select the Workspace area and specify the 128-byte table data to be saved or opened.

(2) Select the target .bin data file from the menu.

2.3.2 Opening and Saving 128-Byte .hex Files

The File menu options are available.

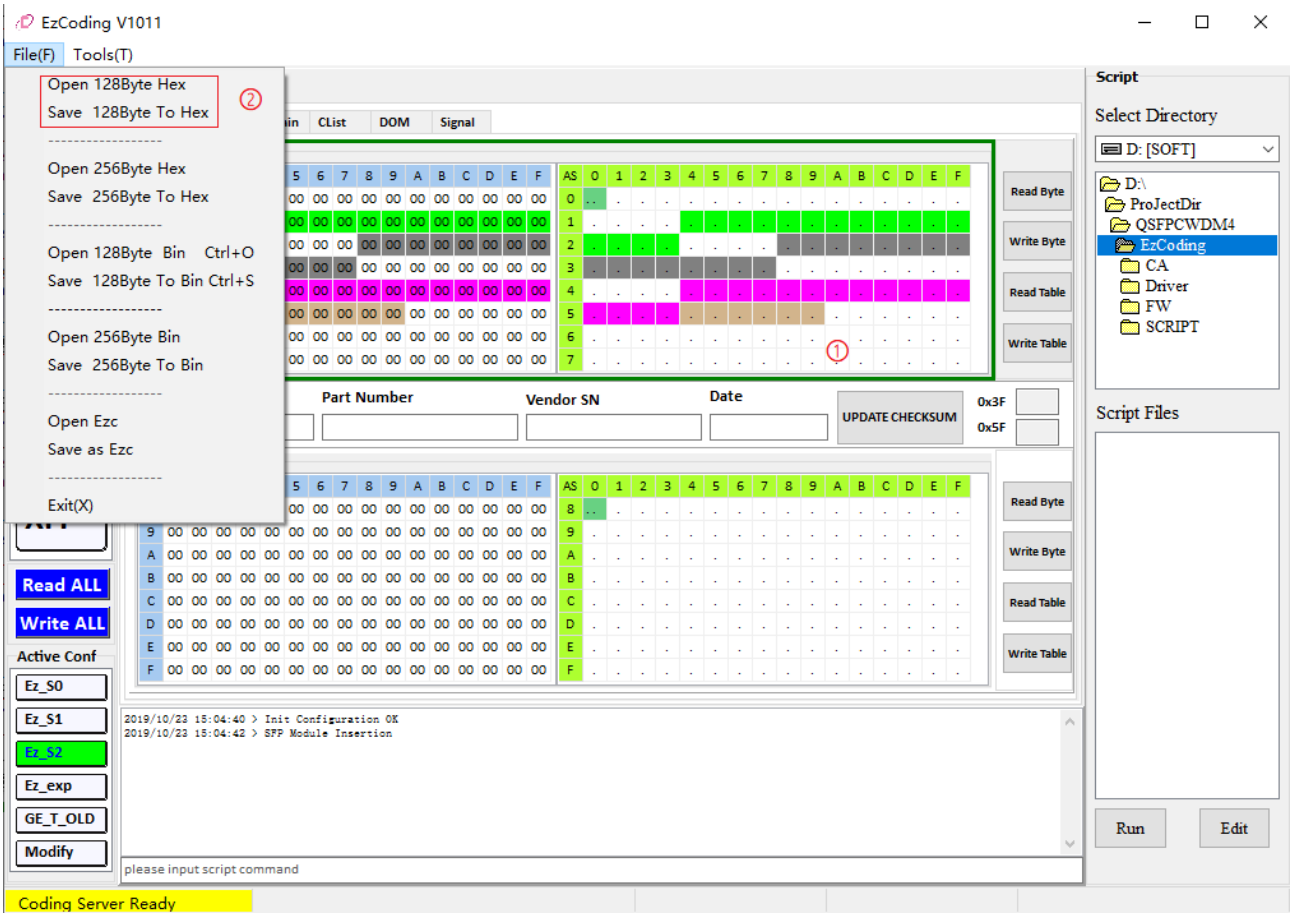

Figure 9 128-Byte .hex File List

(1) Select the Workspace area and specify the 128-byte table data to be saved or opened.

(2) Select the target .hex data file from the menu.

2.3.3 Opening and Saving 256-Byte .bin Files

The File menu options are available.

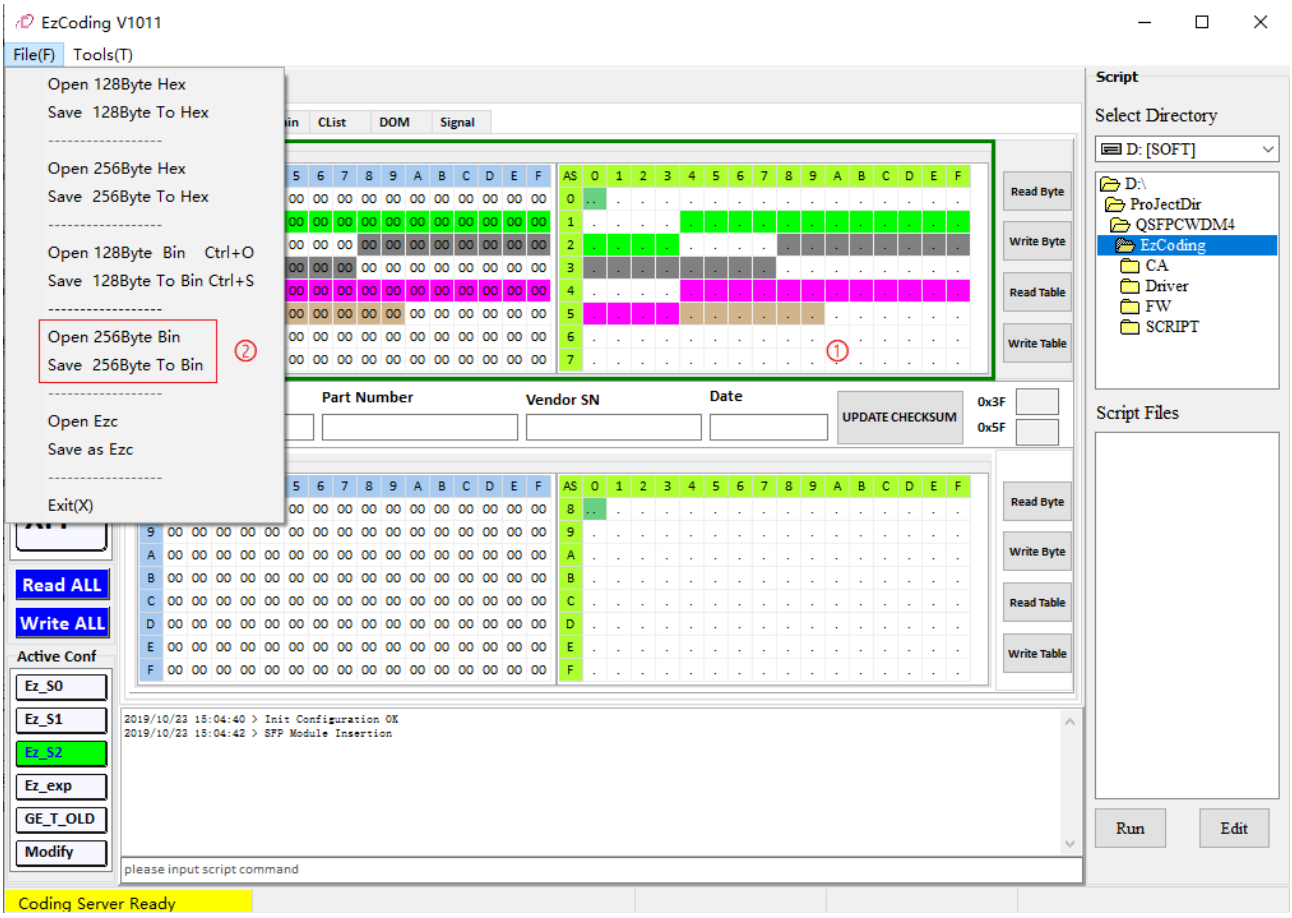

Figure 10 256-Byte .bin File List

(1) Select the target .bin data file from the menu.

(2) A 256-byte data file is saved as two data tables on the current page. This save option applies to data storage in Professional Mode.

2.3.4 Opening and Saving 256-Byte .hex Files

The File menu options are available.

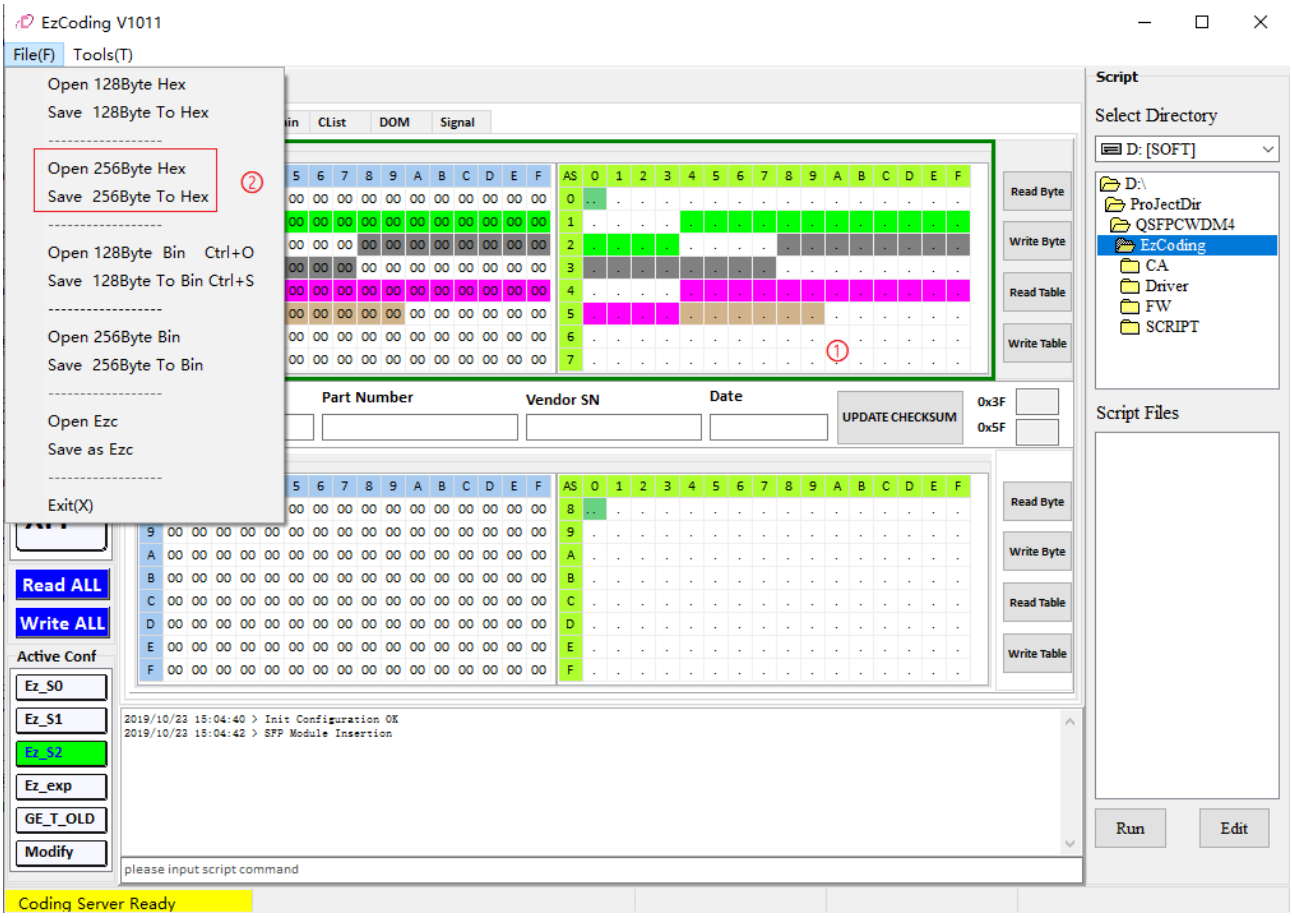

Figure 11 256-Byte .hex File List

(1) Select the target .hex data file from the menu.

(2) A 256-byte data file is saved as two data tables on the current page. This save option applies to data storage in Professional Mode.

2.3.5 Opening and Saving .ezc File Data

To open or save .ezc files, you need to configure the save properties of each data table in Tools → Options under Normal Mode.(Note: A data table, displayed in tabular form, contains 128 bytes of data and is referred to as a data table hereinafter.)

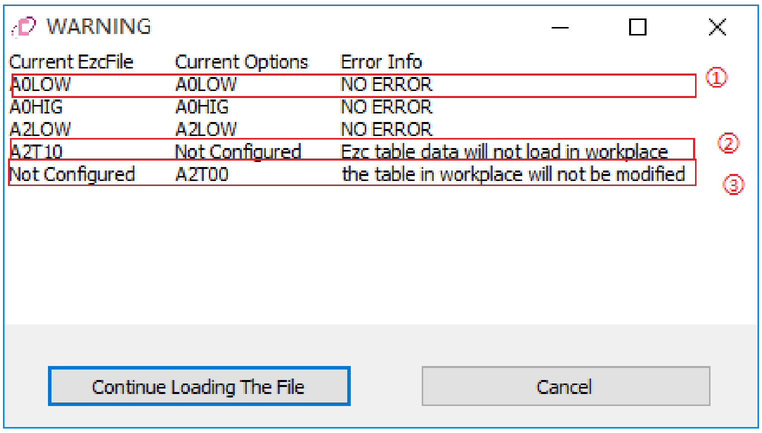

Figure 12 Prompt Dialog for Mismatched Table Selection Between Configuration and File

(1) Select the target .ezc data file from the menu to save or open.

(2) The configuration file will check and compare the number and names of the currently configured data tables with those in the target file. Only tables configured as NO ERROR can be loaded into the Workspace.

(3) When manually modifying the table names of an .ezc data table, set each table name to 5 characters in length and ensure it matches the table names in the configuration file.

```
[common]
version = 1.0.0
copyright = sfpcodingbox.com
module_type = SFP

[A0LOW]
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF

[A0HIG]
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
```

Figure 13 Schematic Diagram of .ezc Data File Content

2.4 Configuration Files

Open Tools → Options to bring up the following dialog box. It is used to configure functions such as module passwords, continuous write mode, display mode, automatic reading, opening/saving of .ezc file data tables, and data table settings. After modifying the configuration, click Save and restart the software to make the settings take effect.

2.4.1 Basic Functions of Configuration Files

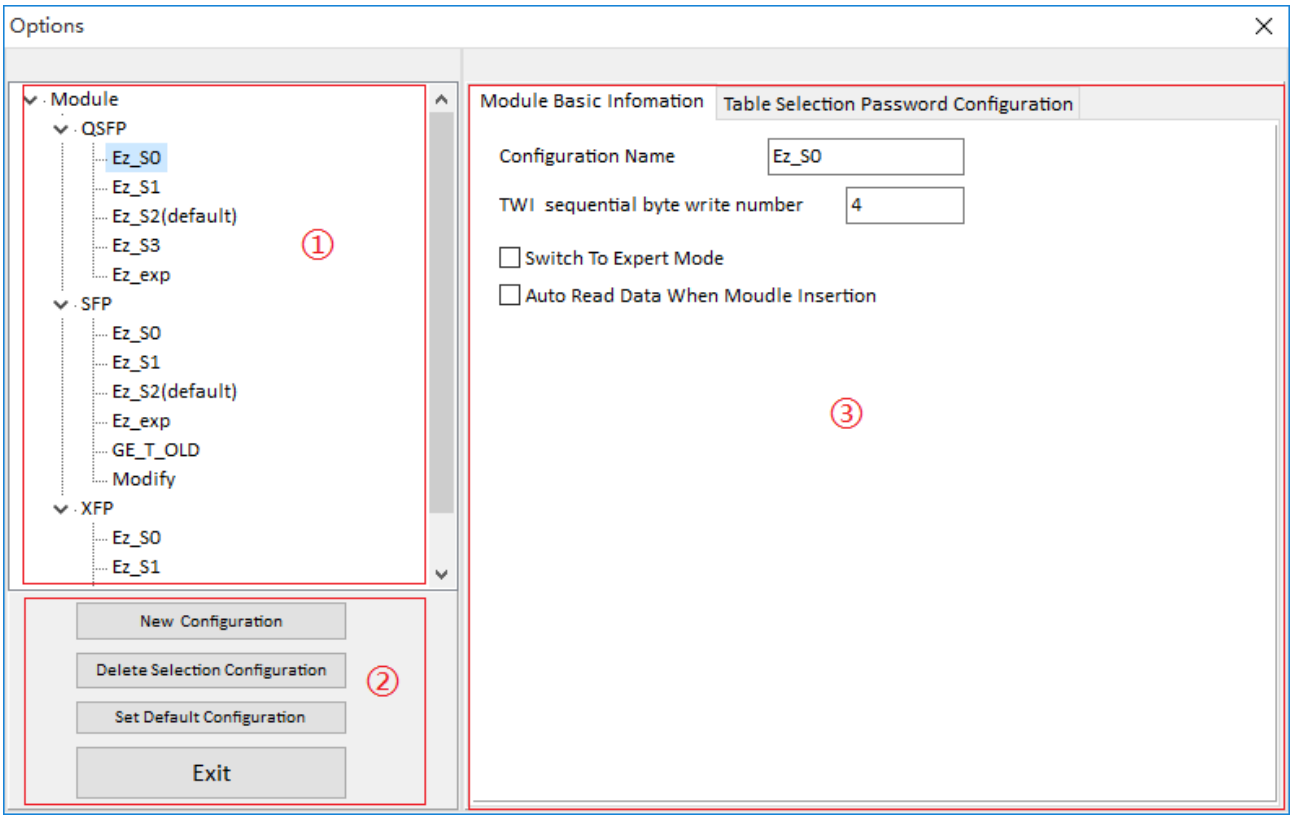

Figure 14 Basic Interface of the Options Panel

(1) Configuration File List: Displays the status of current configuration files. The label (default) indicates that the configuration file is the default one, which is automatically loaded when switching displays on the main interface.

(2) Functions for adding, deleting, and setting default configuration files. Before selecting a button, you need to first select the corresponding module type or configuration.

(3) Basic Information of the Selected Configuration File: Includes the configuration file name, number of continuous write bytes, switching between professional module mode and normal mode, and automatic reading of module data when a module is inserted.

2.4.2 Modifying Configurations

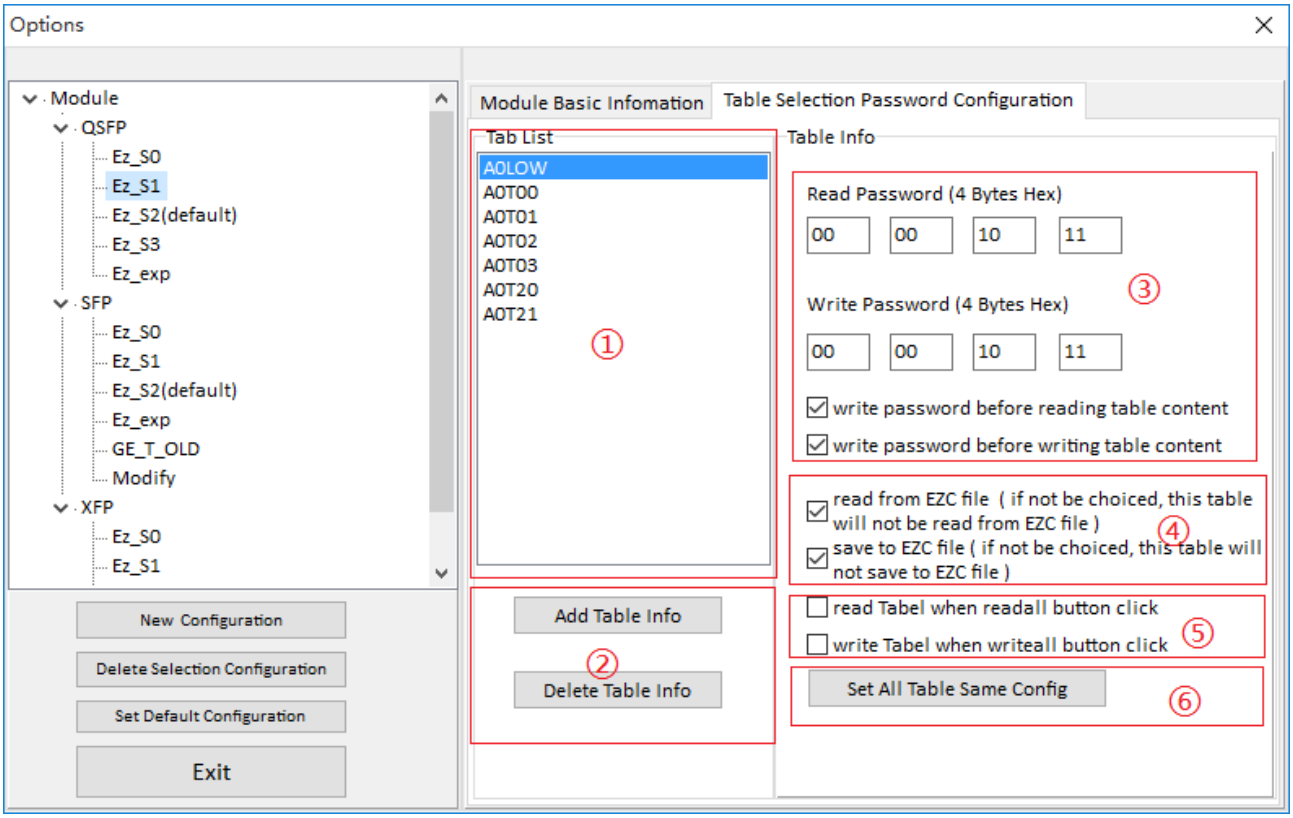

Figure 15 Password Interface of the Options Panel

(1) Data Table Configuration Area: Each configuration item consists of an Address and a Table Selection, with a length limit of 5 characters (the first 2 characters represent the address bits, the last 2 represent the table selection bits, and the middle 1 character is reserved). It is not recommended to set the maximum number of data tables to exceed 10.

(2) Data Table Addition and Deletion Area: TheTWI (Two-Wire Interface) -Table-address can be manually modified within the range of T00-TFF.

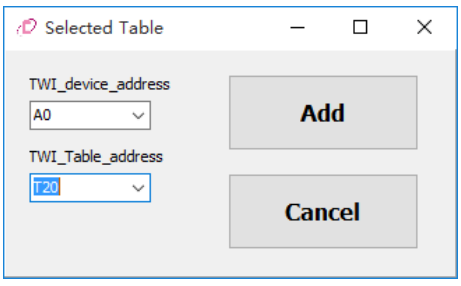

Figure 16 Table Selection Addition Interface

(3) Basic Information of the Selected Configuration File: Includes the configuration file name, number of continuous write bytes, switching between professional module mode and normal mode, and automatic reading of module data when a module is inserted.

(4) Configure the save and open properties of data tables for .ezc files.

(5) Configure the properties of the readall and write buttons in Figure 1 for data tables. It is recommended not to check A0Low andA2Low.

(6) Set the properties of all data tables in the current configuration to be consistent.

2.4.3 Selecting Configurations

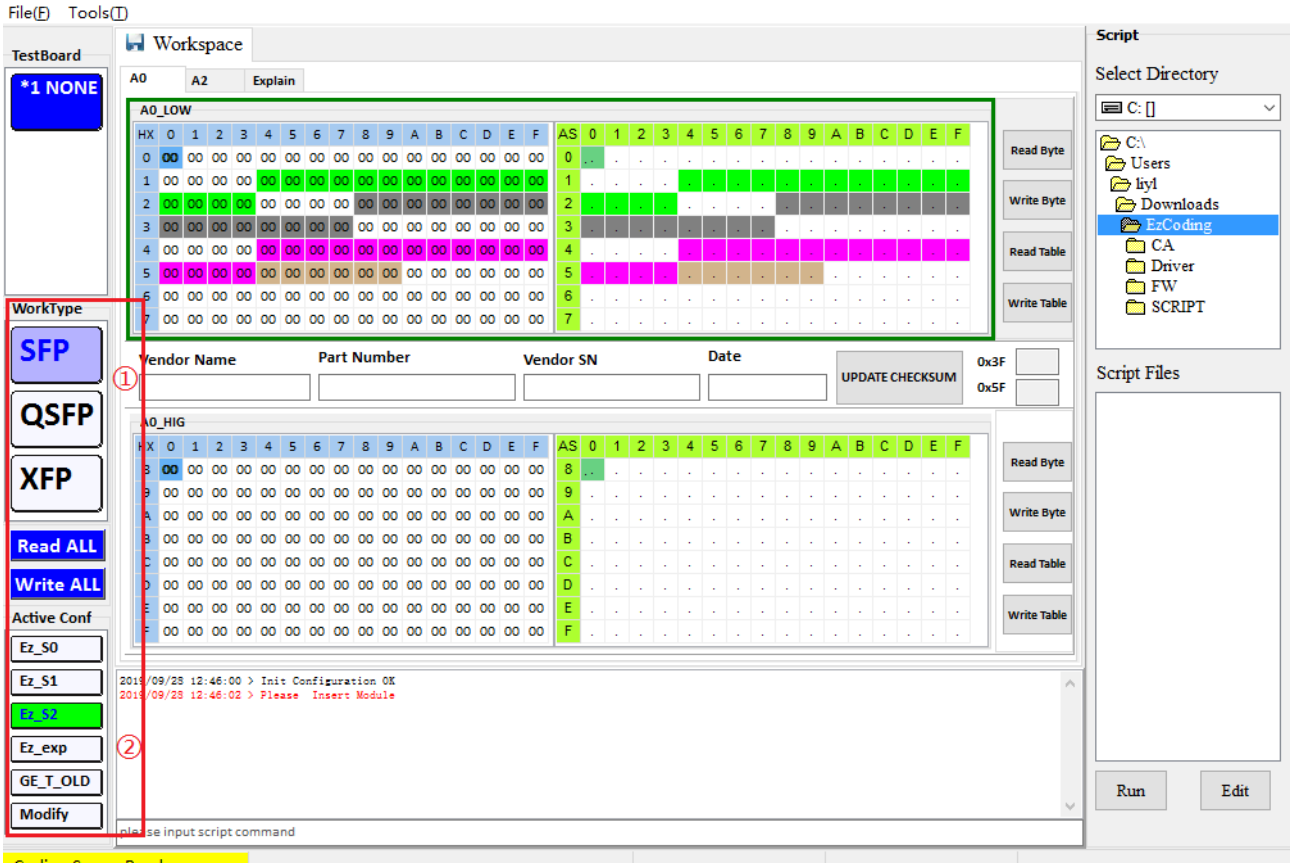

Figure 17 Configuration Selection Interface of the Options Panel

(1) Module Type Selection (Manual): Each module type corresponds to a differentWorkspace interface.

(2) Configuration File Selection Under Module Type (Manual): After selecting a configuration, the corresponding password and read/write configuration data are immediately activated on the interface, and the interface data will not be refreshed. However, the interface will switch displays when toggling between Normal Mode and Professional Mode.

2.5 Professional Mode

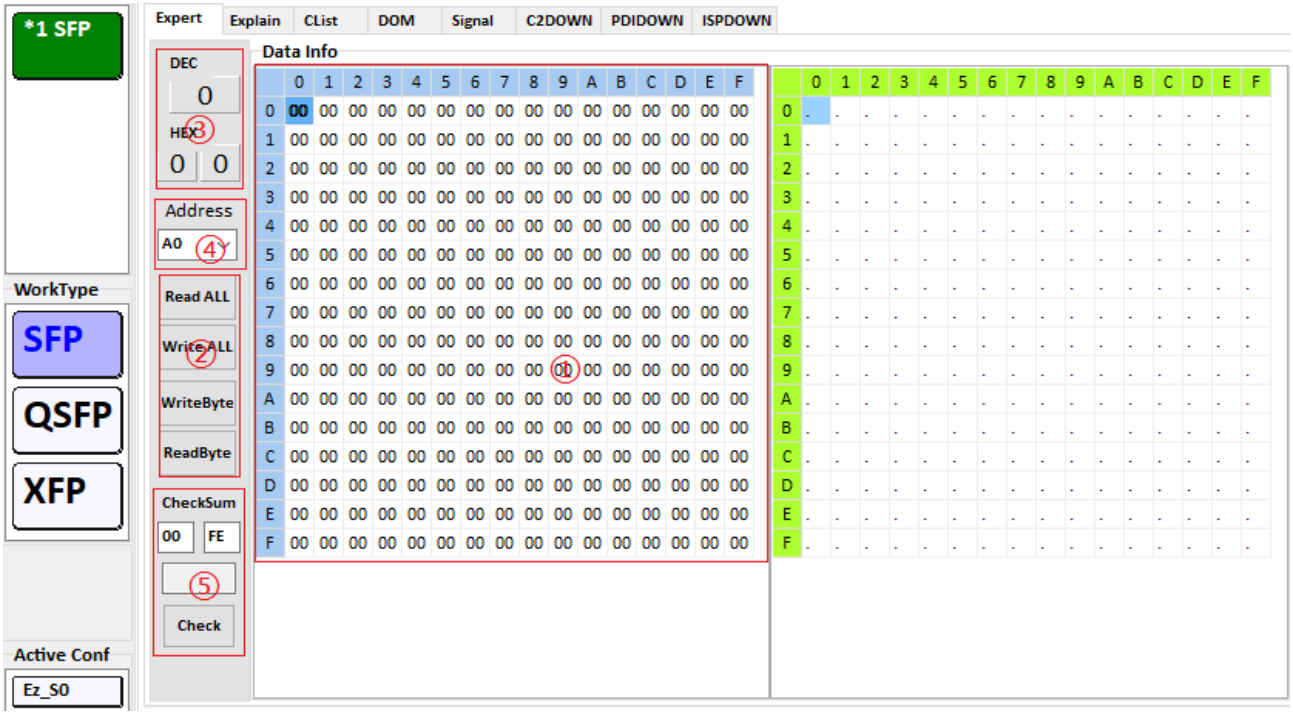

Figure 18 Professional Mode Interface

(1) Basic Data Editing Area: A 256-byte area representing the data range of TWI (Two-Wire Interface) readable/writable device addresses.

(2) Read/Write Operation Functional Area: Provides functions for single-byte write, single-byte read, and full-table read/write. Important Note: Full-table write is only supported for specific modules; refer to the module datasheet before operation. When using the full-table write function, ensure that the data table can be completely saved in the module; otherwise, a checksum error may occur or the module may be damaged. For example, full-table writing will definitely cause errors for QSFP and XFP modules.

(3) Selected Cell Display Area: Displays the position of the data currently being edited or selected. During single-byte write or read operations, the TWI will perform the operation based on the current value.

(4) Module Internal Memory Address: Defined by the module manufacturer. The drop-down list only lists commonly used addresses, and manual input (00-FF) is supported.

(5) Checksum Calculation Tool: Helps calculate the sum of byte data in a specific area in hexadecimal format.

2.6 Script Functions

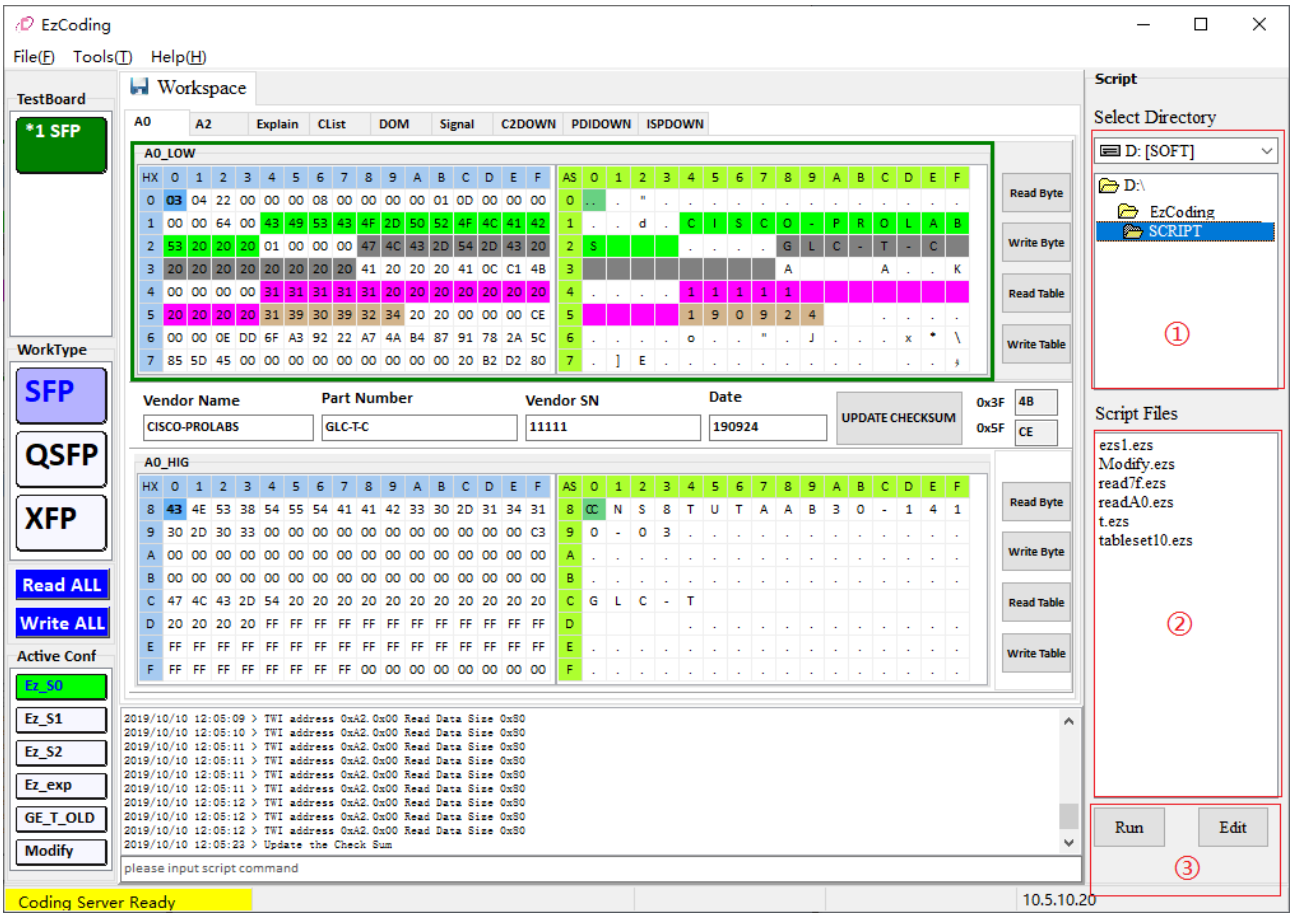

Figure 19 Script Area Interface

(1) Script Directory Selection: The default script directory is the SCRIPT folder under the program directory.

(2) Script File Format: .ezs.

(3) Script Execution and Editing: Script files are text files and must be saved in ANSI format. Close other programs that may occupy the CodingBox USB port before executing scripts.

2.6.1 read Command

The read command parses the TWI (Two-Wire Interface) read function. It mainly includes the device address, sub-address data length, followed by an optional buffer identifier parameter.

Command Format:

read [Module Device Address] [(Optional) Module Sub-address] [(Optional) Data Length] [(Optional) Buffer Identifier]

Parameter Description:

Module Device Address: Defaults to 0xA0 (hexadecimal), e.g., A0

Module Sub-address: Defaults to 0x00 (hexadecimal), e.g., 00

Data Length: Defaults to 128 (decimal), e.g., 128

Buffer Identifier: An identifier starting with ':', e.g., " :Databuf".

Application Example:

read A0 80 128 :databuf –Uses the I2C interface to read 128 bytes of data from address 0xA0, sub-address 0x80, and store it in the buffer labeled " databuf".

2.6.2 write Command

The write command parses the TWI (Two-Wire Interface) write function. It mainly includes the device address, sub-address data length, followed by an optional buffer identifier or data content parameter.

Command Format:

write [Module Device Address] [Module Sub-address] [Data Length] [Buffer Identifier/Data Content]

Parameter Description:

Module Device Address: Defaults to 0xA0 (hexadecimal), e.g., A0

Module Sub-address: Defaults to 0x00 (hexadecimal), e.g., 00

Data Length: Defaults to 128 (decimal), e.g., 128

Buffer Identifier/Data Content: An identifier starting with ':', e.g., " :Databuf".

Application Example:

write A0 80 128 :databuf –Uses the I2C interface to write 128 bytes of data from the buffer labeled " databuf" to address 0xA0, sub-address 0x80.

2.6.3 open Command

The open command is used to load a local file into the buffer.

Command Format:

open [Buffer Identifier] [File Path]

Parameter Description:

Buffer Identifier: An identifier starting with ':', e.g., " :Databuf".

File Path: Absolute path of the file

Application Example:

open :bff code.hex

2.6.4 save Command

The save command is used to save the buffer data to a local file.

Command Format:

save [Buffer Identifier] [File Path]

Parameter Description:

Buffer Identifier: An identifier starting with ':', e.g., " :Databuf".

File Path: Absolute path of the file.

Application Example:

save :bff code.hex

2.6.5 show Command

The show command is used to display the buffer data in the Log interface.
Command Format:
show [Buffer Identifier]
Parameter Description:
Buffer Identifier: An identifier starting with ':', e.g., " :Databuf" .
Application Example:
show :bff

2.7 Digital Diagnostics Function

The software determines whether to display DOM (Digital Optical Monitoring) information based on the selected Workspace type and by comparing it with the inserted module type.

2.7.1 Prerequisites for Enabling the Digital Diagnostics Function

(1) A CodingBox is connected to the PC.

(2) A module is inserted into the CodingBox, and the correct Workspace type is selected.

2.7.2 SFP Module Digital Diagnostics

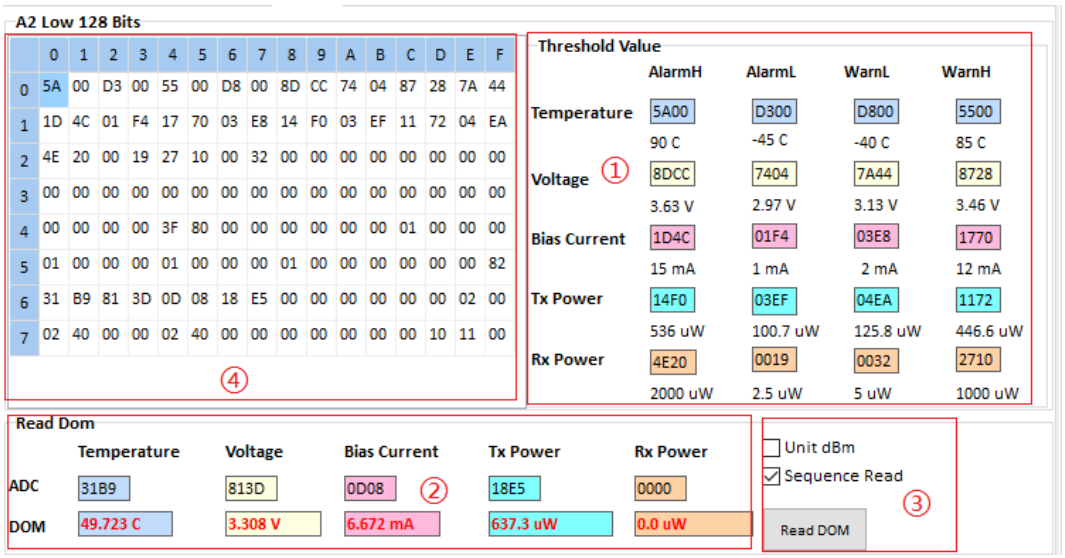

Figure 20 SFP Digital Diagnostics Interface

(1) Threshold Display Area: Includes temperature, voltage, bias current, transmit optical power, and receive optical power.

(2) Real-time Digital Diagnostics Information: Displays the sampled values of the module.

(3) Control Area: Provides unit conversion, continuous reading, and single reading functions. Pressing ReadDom switches the module to single reading mode.

(4) A2Low Byte Area Related to Digital Diagnostics.

2.7.3 QSFP Module Digital Diagnostics

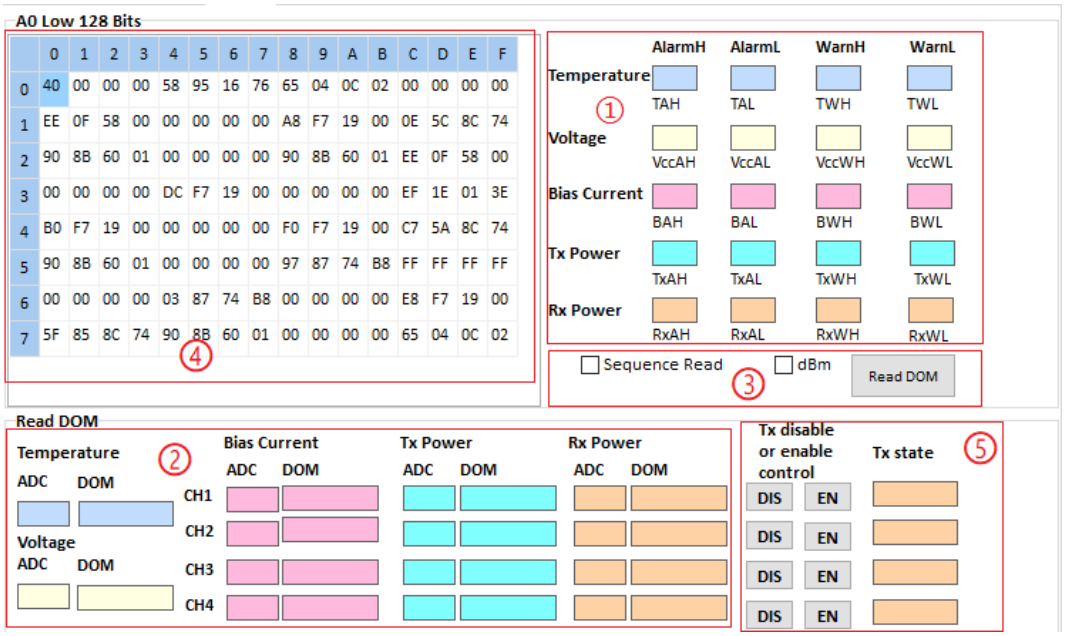

Figure 21 QSFP Digital Diagnostics Interface

(1) Threshold Display Area: Includes temperature, voltage, bias current, transmit optical power, and receive optical power.

(2) Real-time Digital Diagnostics Information: Displays the sampled values of the module.

(3) A0Low Byte Area Related to Digital Diagnostics.

(4) The software controls the on/off state of the transmit optical power by modifying the value of byte 0x56h in A0Low.

(5) Control Area: Provides unit conversion, continuous reading, and single reading functions. Pressing ReadDom switches the module to single reading mode.

2.7.4 XFP Module Digital Diagnostics

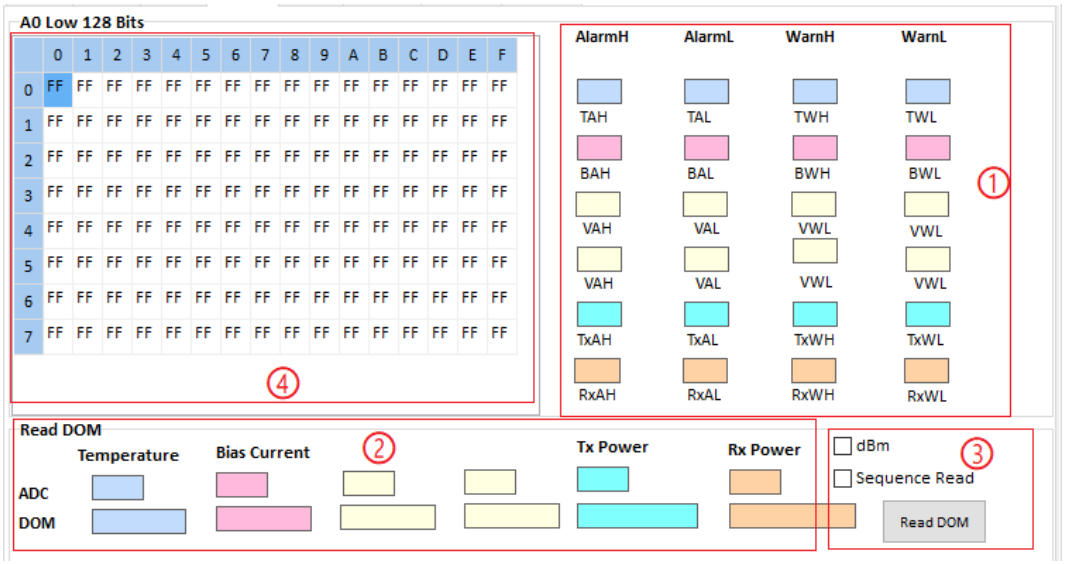

Figure 22 XFP Digital Diagnostics Interface

(1) Threshold Display Area: Includes temperature, voltage, bias current, transmit optical power, and receive optical power.

(2) Real-time Digital Diagnostics Information: Displays the sampled values of the module.

(3) Control Area: Provides unit conversion, continuous reading, and single reading functions. Pressing ReadDom switches the module to single reading mode.

(4) A0Low Byte Area Related to Digital Diagnostics.

2.8 Low-Speed Signal Function

The low-speed signal function is used to control the low-speed pins of the module, enabling the host to control functions of the optical module such as transmit optical power and LOS (Loss of Signal).

2.8.1 SFP Module Low-Speed Signal Function

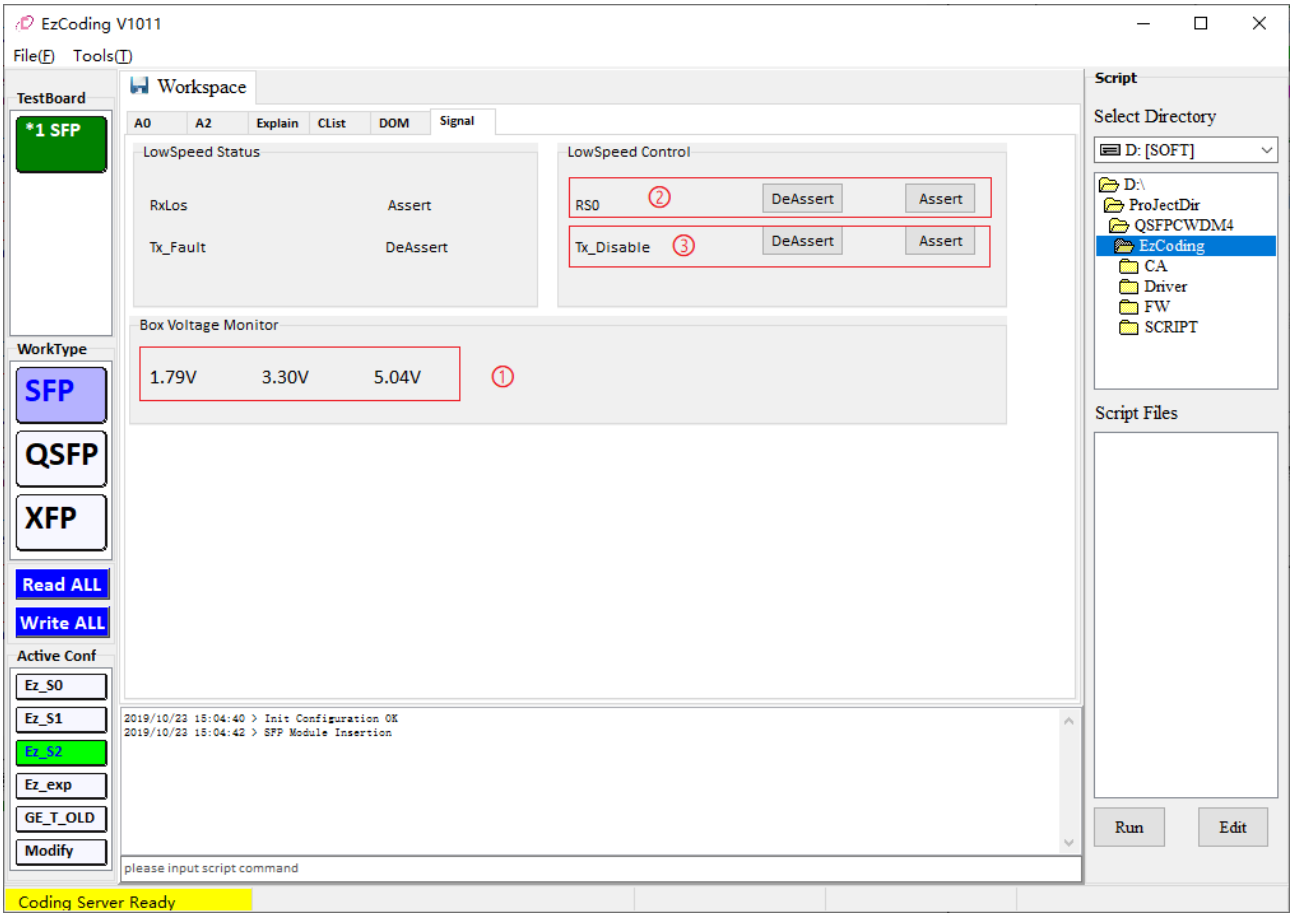

Figure 23 SFP Low-Speed Signal Interface

(1) Three channels of power supply voltage provided by the test board.

(2) SFP module rate selection function pins.

(3) Host-controlled module transmit optical power pins.

2.8.2 QSFP Module Low-Speed Signal Function

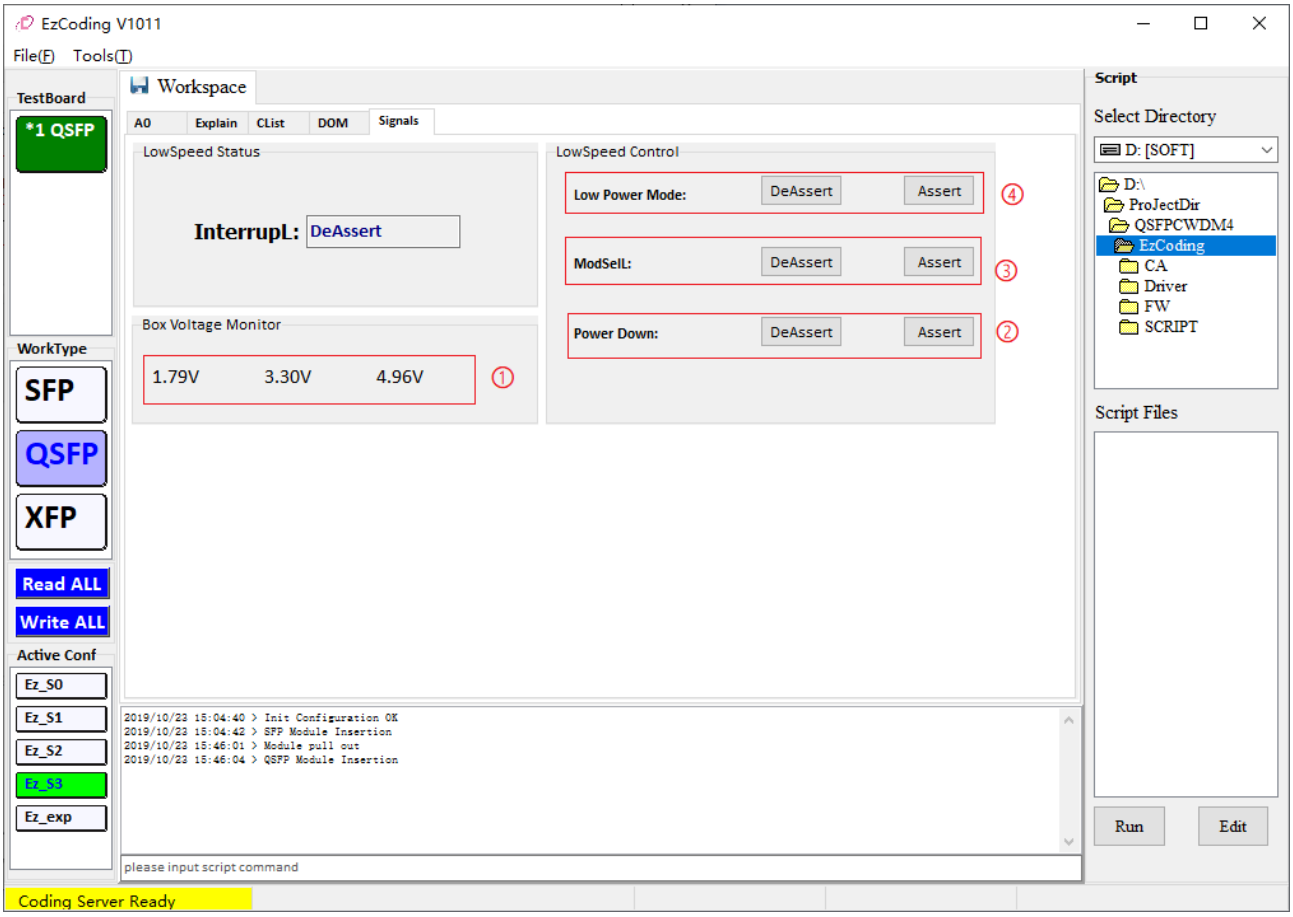

Figure 24 QSFP Low-Speed Signal Interface

(1) Three channels of power supply voltage provided by the test board.

(2) QSFP module power control switch (requires module support for this function).

(3) QSFP module mode selection function.

(4) QSFP module low-power consumption mode.

2.8.3 XFP Module Low-Speed Signal Function

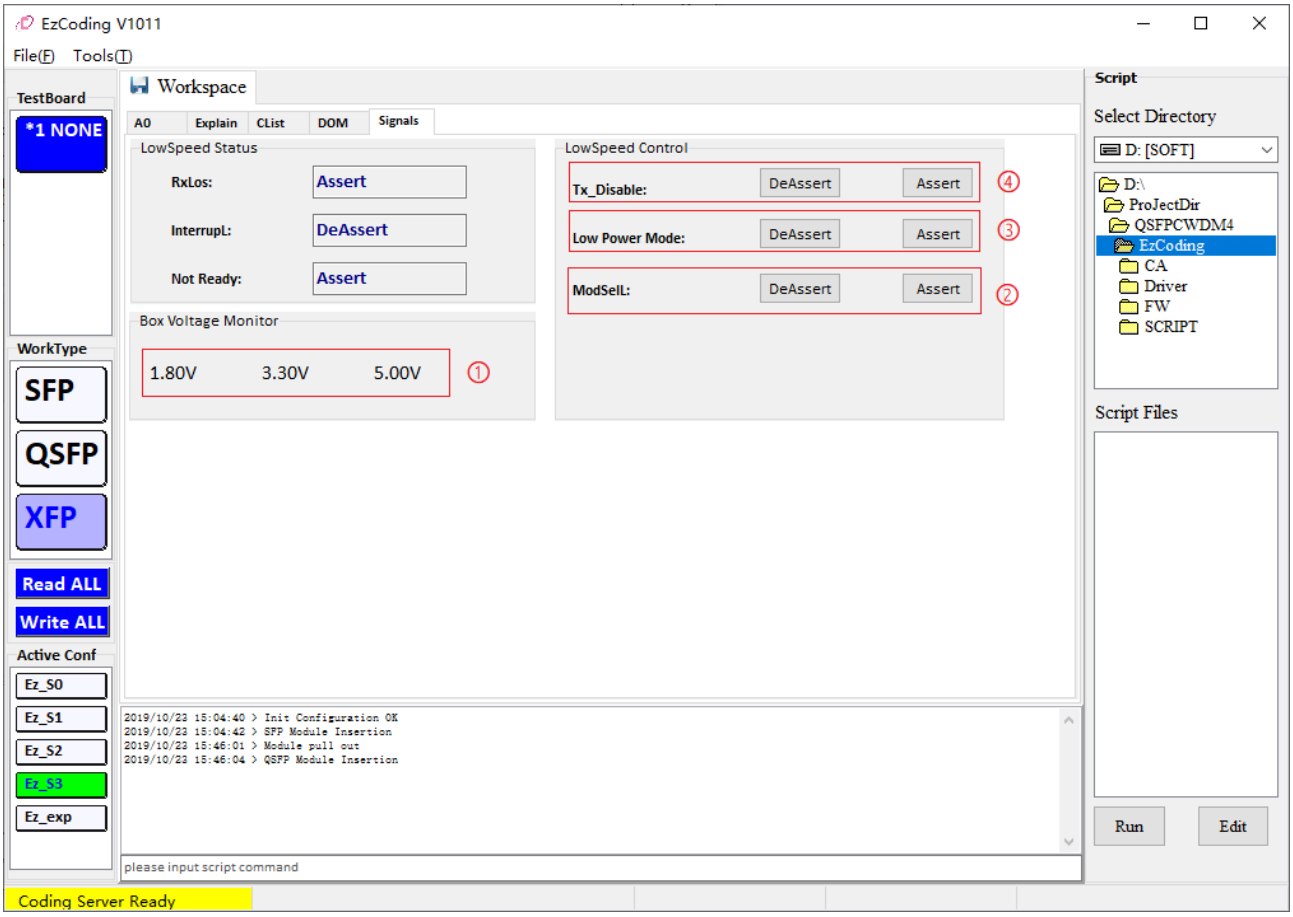

Figure 25 XFP Module Low-Speed Signal Function

(1) Three channels of power supply voltage provided by the test board.

(2) XFP module power control switch (requires module support for this function).

(3) XFP module mode selection function.

(4) XFP module low-power consumption mode.

3 FAQ

3.1 The software cannot detect the test board

(1) Confirm that the CodingBox device exists in Device Manager (see Figure 5).

(2) Verify that the USB port, CodingBox, and USB cable are in good condition. You can try disabling USB power management.

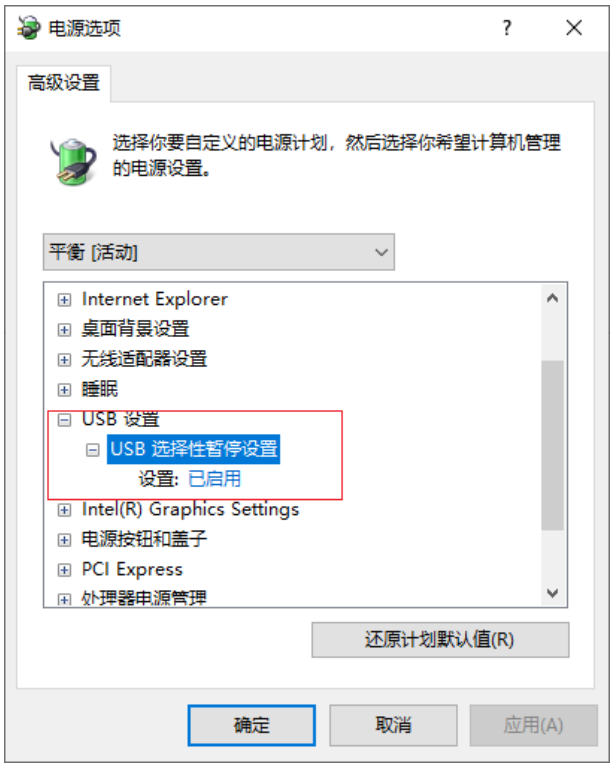

Figure 26 USB Power Management Settings

(1) Check if the test board displays '0' after being connected and powered on. If it displays '8', a firmware upgrade is required.

(2) For Windows 7 and earlier systems, you need to install the driver file. Open the driver file in the software's Driver directory to install it.

(3) For Windows 10 systems, if different devices with the same PID (Product ID) / VID (Vendor ID) have been used before, you need to delete the PID and VID data in the registry.

3.2 Configuration file data is lost after saving and reopening

(1) Configuration file data is stored in the EzCode.db file in the program directory. Insufficient user permissions will prevent data from being saved to the file.

(2) Move the software folder to the desktop or a directory other than the C drive.