HiReg

User Guide

Issue 04

Date 2015-04-30

Trademarks and Permissions

 , HISILICON, and other HiSilicon icons are trademarks of HiSilicon Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

About This Document

Purpose

This document describes how to use the HiReg tool.

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3751	V800
Hi3716C	V100
Hi3716M	V100
Hi3716C	V2XX
Hi3719C	V1XX
Hi3718C	V1XX
Hi3719M	V1XX
Hi3718M	V1XX
Hi3716M	V4XX
Hi3716M	V3XX
Hi3798M	V1XX
Hi3796M	V1XX
Hi3110E	V5XX
Hi3798C	V2XX

Intended Audience

This document is intended for:

- Technical support engineers
- Board hardware development engineers

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 04 (2015-04-30)

This issue is the fourth official release, which incorporates the following changes:

Hi3798C V200, Hi3716M V420, and Hi3716M V410 are supported.

Issue 03 (2015-03-10)

This issue is the third official release, which incorporates the following changes:

Hi3110E V500 is supported.

Issue 02 (2014-10-31)

This issue is the second official release, which incorporates the following changes:

Hi3796M V100 and Hi3798M V100 are supported.

Issue 01 (2014-09-05)

This issue is the first official release, which incorporates the following changes:

Chapter 2 GUI and Basic Functions

Section 2.3 is modified.

Issue 00B01 (2014-05-15)

This issue is the first draft release.

Contents

Figures

1 Introduction to the HiReg

1.1 Overview

The HiReg is a tool for debugging HiSilicon chip registers, inter-integrated circuit (I^2C) registers, and DDR data. After the HiReg is connected to a board over the serial port or network port, the following functions are supported:

- Reading or writing a single data segment
- Reading or writing data segments in batches
- Reading data periodically; Recording data.

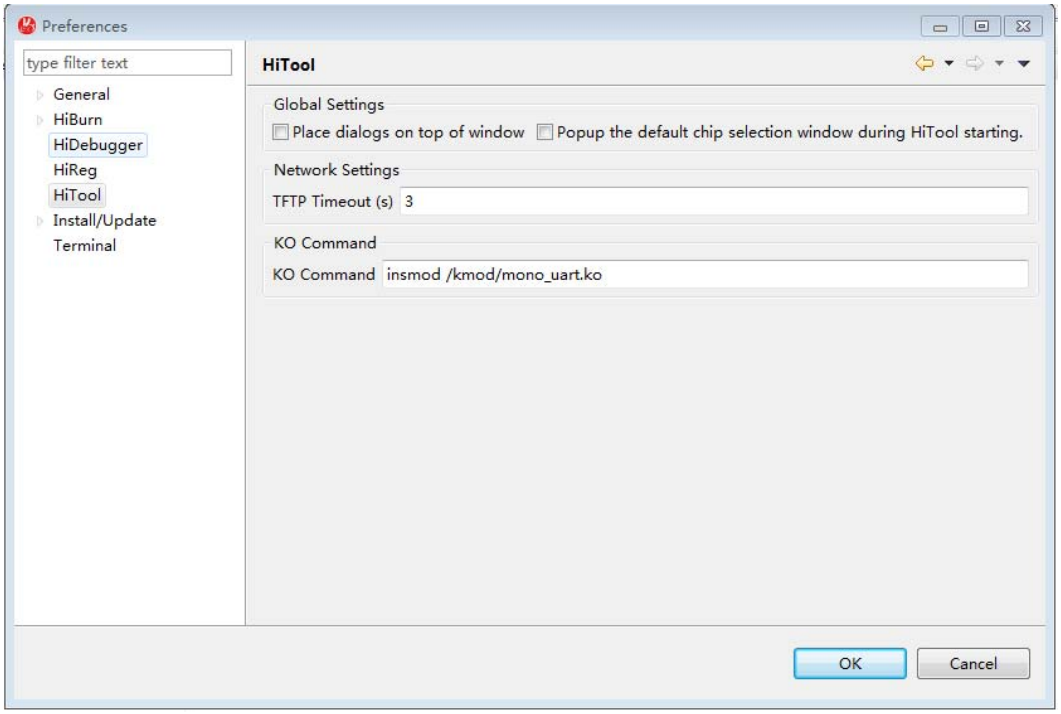
1.2 Environment Preparations

Before using the HiReg, do as follows:

- Install the regtool on the board. The latest version is reg-tools-1.0.0.
- Install the latest Btools on the board. The latest version is ver0.0.1_2013018.

The KO command is provided by default. You can configure the KO command in the **Preferences** dialog box shown in Figure 1-1.

Figure 1-1 Configuring the KO command



2 GUI and Basic Functions

This chapter describes the main GUI of the HiReg, HiReg perspective, HiReg toolbar, and basic functions.

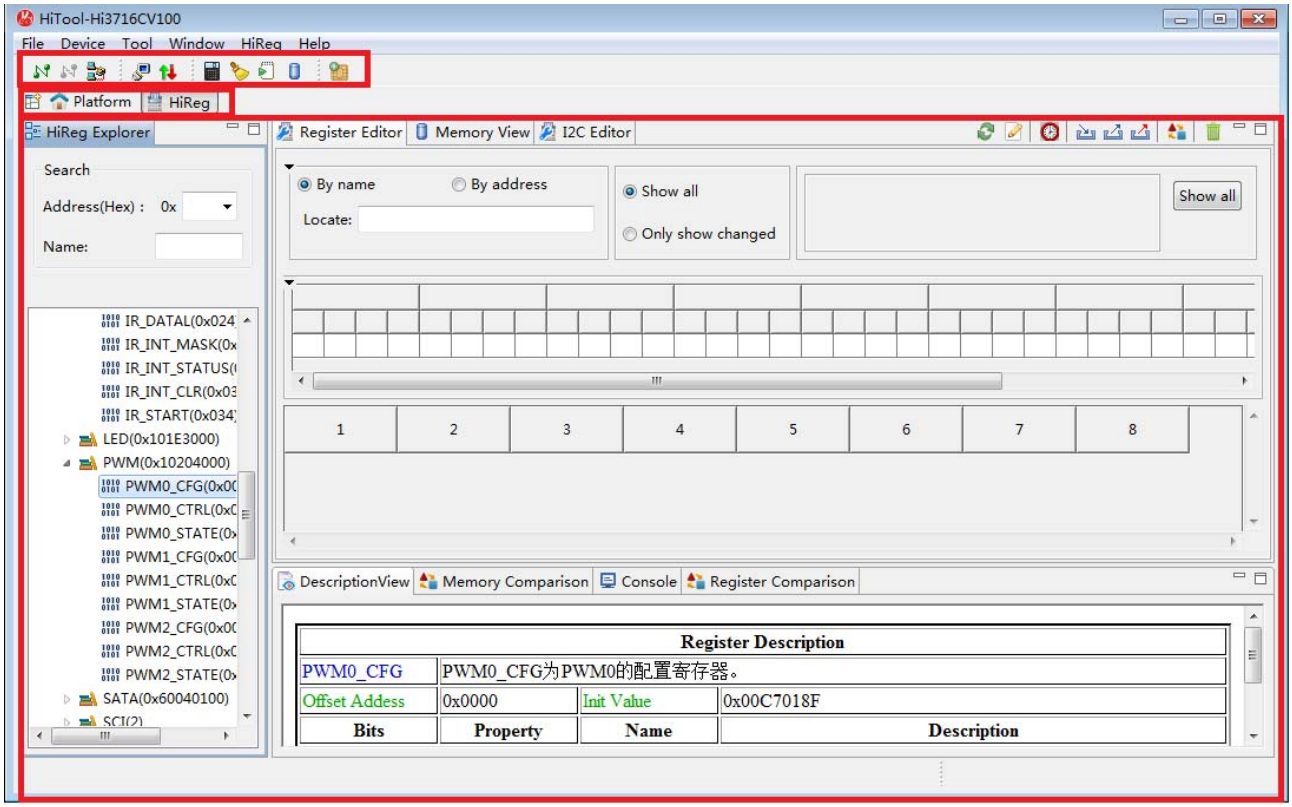
2.1 Main GUI

NOTE

This document uses the Hi3716C V100 board as an example. The description is similar for other chip boards.

Figure 2-1 shows the main GUI for debugging registers after the HiReg connects to the Hi3716C V100 board over the serial port.

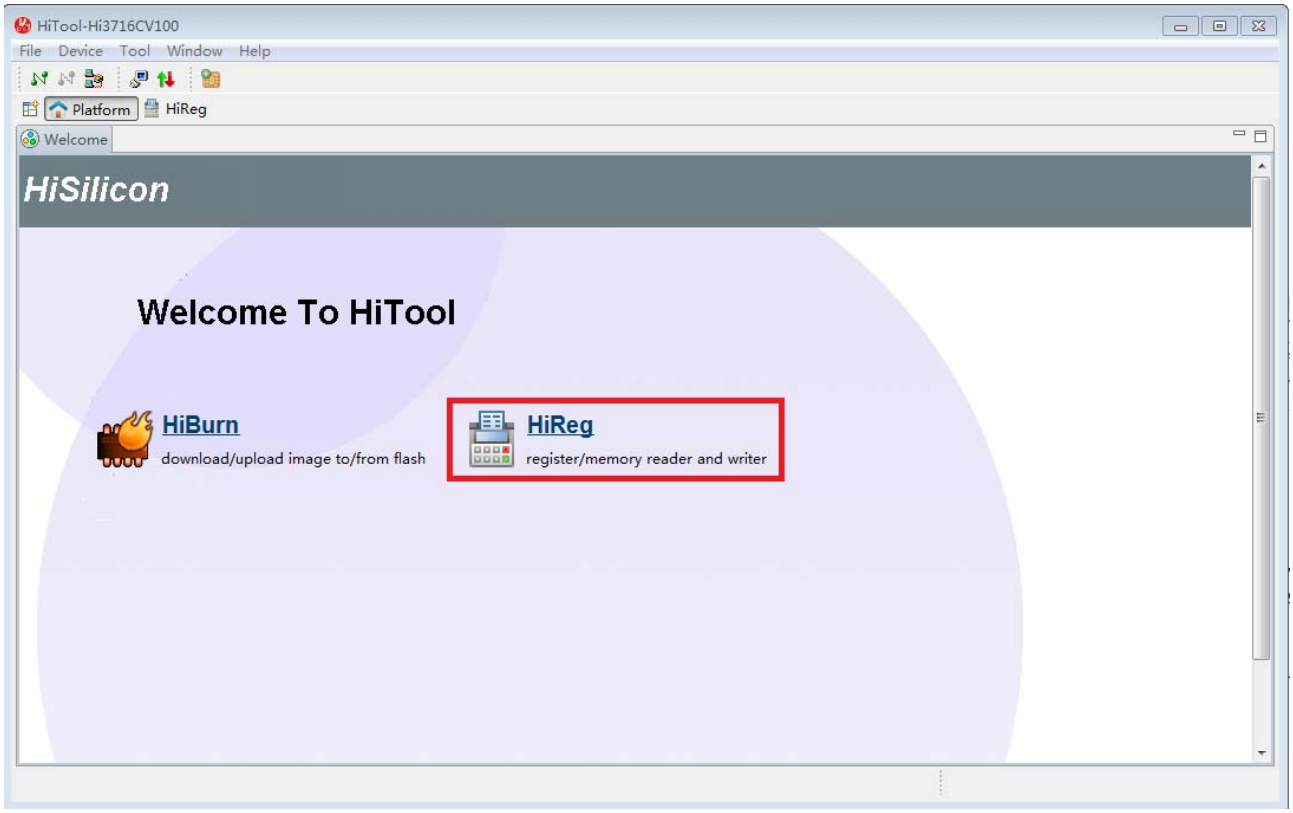
Figure 2-1 Main GUI for debugging registers



2.2 HiReg Perspective

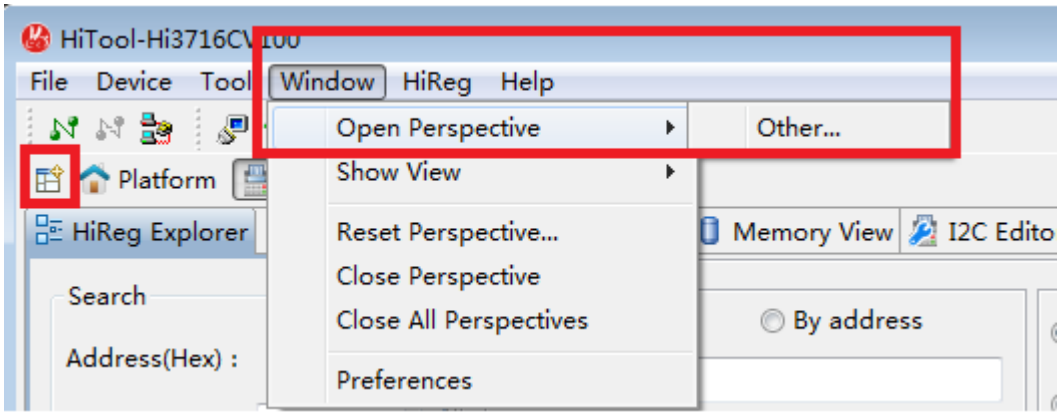
In the **Welcome** wizard of the HiTool, click **HiReg**, as shown in Figure 2-2.

Figure 2-2 Clicking the HiReg



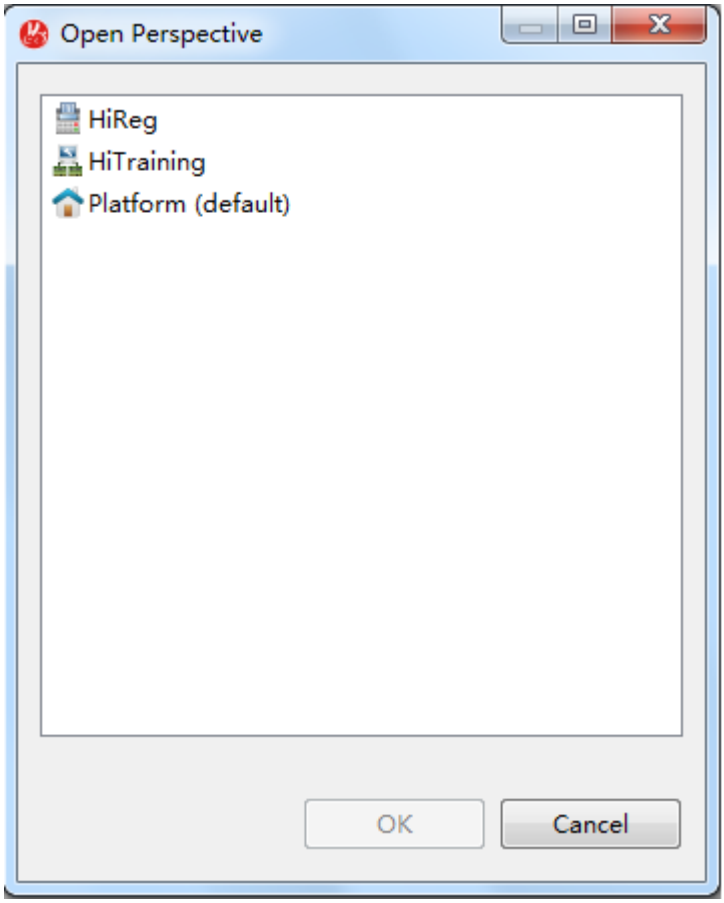
You can also click  or choose **Window** > **Open Perspective** > **Other**, as shown in Figure 2-3.

Figure 2-3 Opening the HiReg perspective



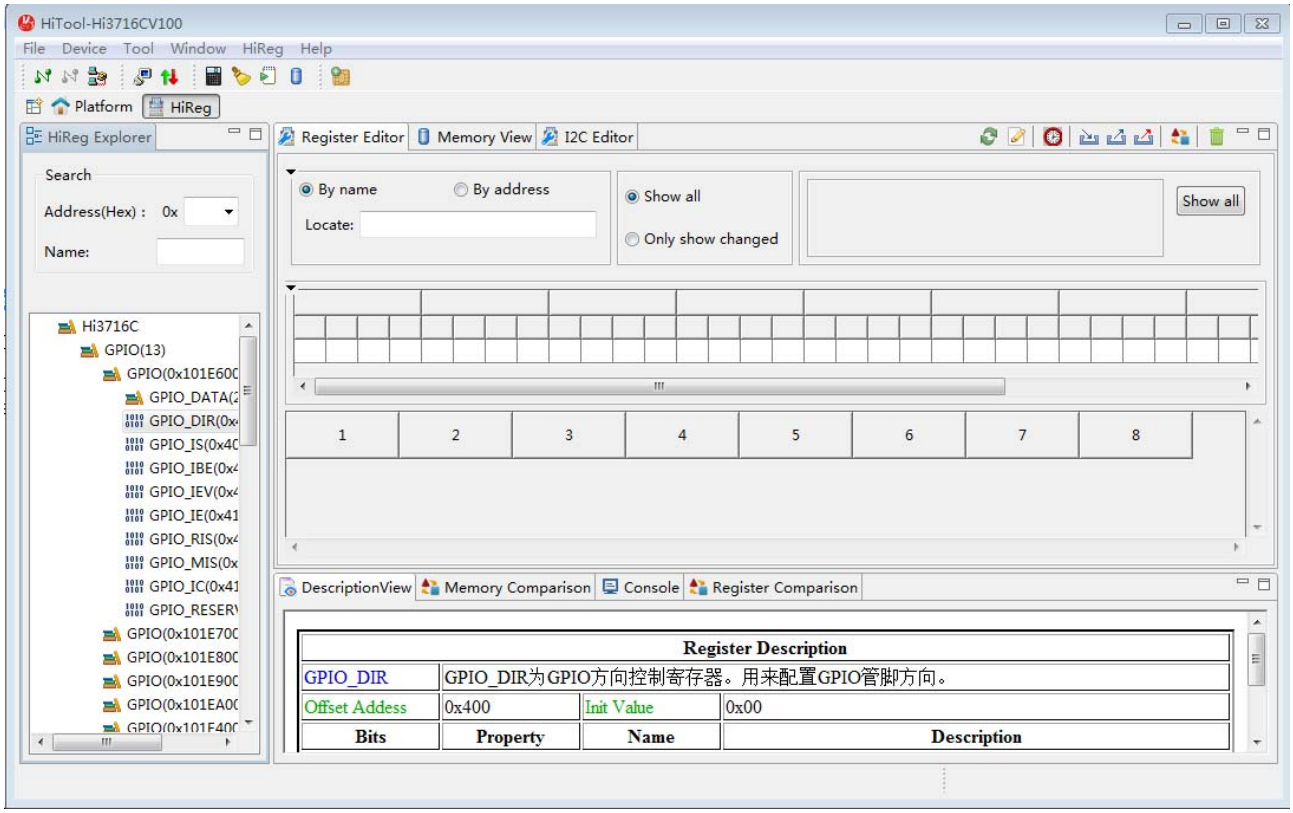
In the displayed **Open Perspective** dialog box, click **HiReg** and **OK**.

Figure 2-4 Clicking HiReg



Then the HiReg perspective shown in Figure 2-5 is displayed.

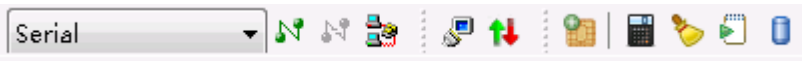
Figure 2-5 HiReg perspective



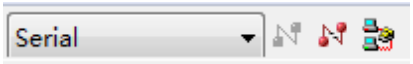
2.3 HiReg Toolbar

Figure 2-6 shows the HiReg toolbar.

Figure 2-6 HiReg toolbar



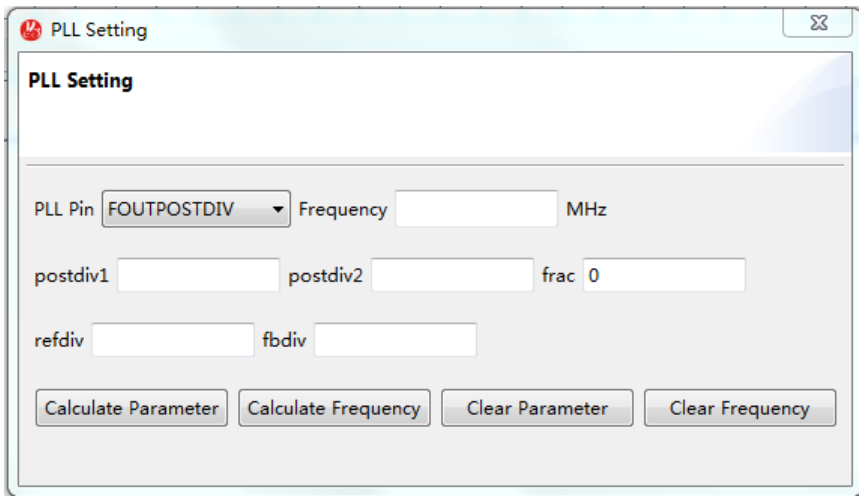
The following describes the icons on the HiReg toolbar:

- 

  - Function: connection manager
  - Description: The icons are used to configure connections.
- 
  - Function: connection icon
  - Description: If the icon becomes green, a connection is established.
- 
  - Function: disconnection icon

  - Function: If the icon becomes red, a connection is established; if the icon is dimmed, the connection is ended.
- 
  - Function: terminal tool
  - Description: Clicking this icon opens the **Terminal** window.
- 
  - Function: TFTP tool
  - Description: Clicking this icon opens the TFTP tool view.
- 
  - Function: importing data sheets
  - Description: The files to be imported must be in .zip format; otherwise, files fail to be imported.
- 
  - Function: calculating the PLL divider parameters
  - Description: Select a PLL pin, enter the expected output frequency, and click **Calculate Parameter**. The integral divider parameters for PLL configuration registers (including **postdiv1**, **postdiv2**, **refdiv**, and **fbdiv**) and the decimal divider parameter **frac** are obtained. If the entered frequency is an integer, **frac** is **0**. You can also enter a frequency and some divider parameters and click **Calculate Parameter** to obtain other divider parameters; or you can enter all divider parameters and click **Calculate Frequency** to obtain the output frequency of the PLL pin.

Figure 2-7 Calculating the PLL divider parameters



- 
  - Function: running the memory clear command that is similar to the **Himc** command
  - Description: Clicking this icon opens the window for running the memory clear command.
- 
  - Function: running a script

- Description: Clicking this icon parses and runs a script file.

A script file must meet the following requirements:

- The comment line starts with #.
- A line has only one statement.
- The executable statement in each line is the command supported by the board. The executable statement must start with himd, himd.l, himm, himc, hier, or hiew; otherwise, the statement is an invalid command.
- The HiReg sends a command to the board directly without parsing it. The extension name .data of the script file is recommended.

  See the following example:

```
#only for test
himd 0x80000000 0x4
himd.l 0x80000000 0x4
#end
```

- 
  - Function: memory view
  - Description: Clicking this icon opens the **Memory View** window.

3 Register Editor

3.1 Functions

The HiReg can be used to debug chip registers. It has the following functions:

- Supports two connection modes: serial port connection and network port connection.
- Detects the chip model.

  The function of debugging HiReg registers is available only when the model of the chip on the board that matches the chip model configured in the HiTool is connected.
- Displays the structure of chip registers.

  Multiple register views can be added for editing and comparing data blocks of various registers.
- Reads and writes to registers.

  One or more registers can be read or written at a time, and registers can be read periodically.
- Views, filters, and edits register data.
- Exports, imports, and compares register data.

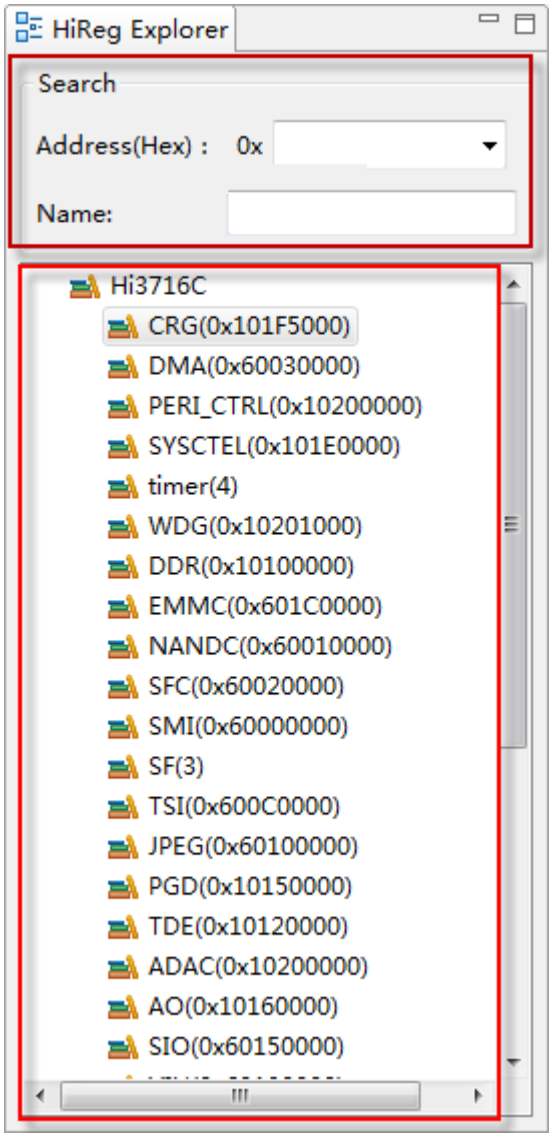
  Memory data can be imported, exported, or compared.
- Integrates register manuals.
- Supports the console output.

3.2 HiReg Explorer

As shown in Figure 3-1, the **HiReg Explorer** perspective is divided into two parts:

- Register query area
- Register list area

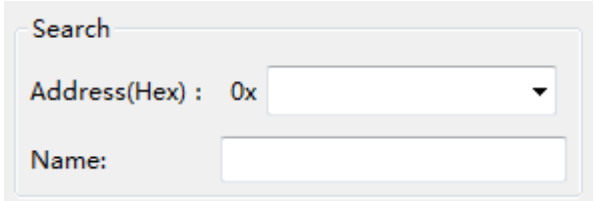
Figure 3-1 HiReg Explorer



3.2.1 Register Query Area

Figure 3-2 shows the **Search** group box for querying a register.

Figure 3-2 Querying a register



A register can be queried in either of the following two ways:

- By address

  Enter a hexadecimal address in **Address(Hex)** and press **Enter**. If the register with the entered address exists, it is added to the register list in **Register Editor**; if the register does not exist, an unknown register is constructed and then added to the register list in **Register Editor**.

- By name

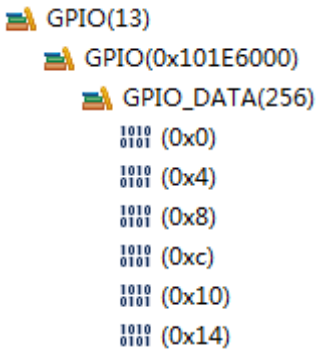
  Enter a register name in **Name** and press **Enter**. If the register with the entered name exists, it is added to the register list in **Register Editor**; if the register does not exist, the original register list is displayed.

3.2.2 Register List Area

The following describes the register list area:

- Register list

Figure 3-3 List node



The structure of a complete register list is as follows: Chip > Module Group > Module > Register Group > Register.

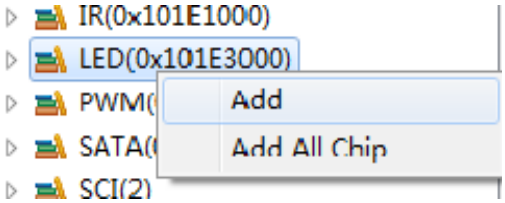
- Operations
  - Double-click operations

    Chip node: Double-clicking this node has no effect.

    Module Group, Module, or Register Group node: Double-clicking any node adds all registers under the node to **Register Editor**.

    Register node: Double-clicking this node adds the register to **Register Editor**.
  - Shortcut menu

Figure 3-4 Shortcut menu



CAUTION

No matter whether you add registers by choosing **Add** or **Add All Chip** from the shortcut menu, a register can be added only once. That is, registers are unique in the register list.

- **Add**: Adds all registers under the current node to **Register Editor**.
- **Add All Chip**: Adds all registers of the current chip to **Register Editor**.

3.2.3 Chip Model Detection

After a board is connected, the HiReg automatically detects whether the model of the chip on the board is consistent with the configured chip model.

- If yes, a tree structure is displayed in **HiReg Explorer**, and you can add registers to **Register Editor** to read or write to them.
- If no, a message is displayed in **HiReg Explorer**, indicating that the chip model mismatches, as shown in Figure 3-5. In this case, choose **Device** > **Switch Device**, select a correct chip model, and try again. See Figure 3-6.

Figure 3-5 Chip model detection

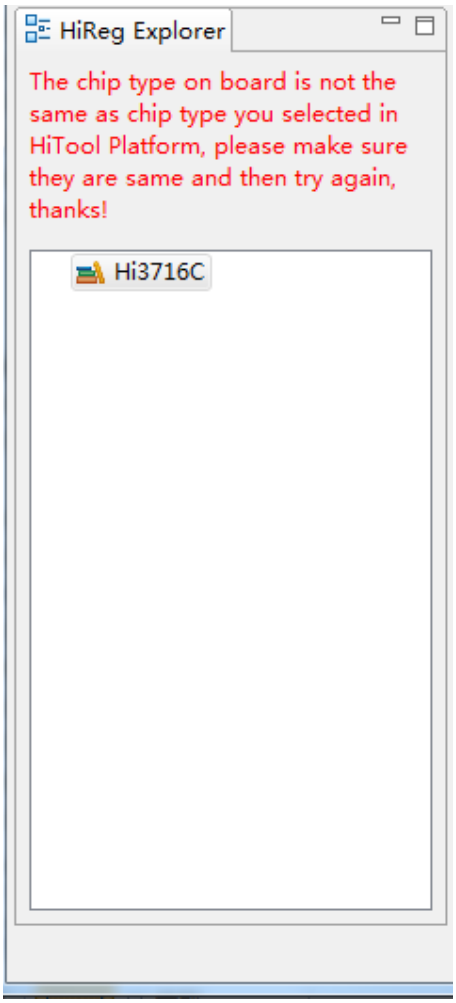
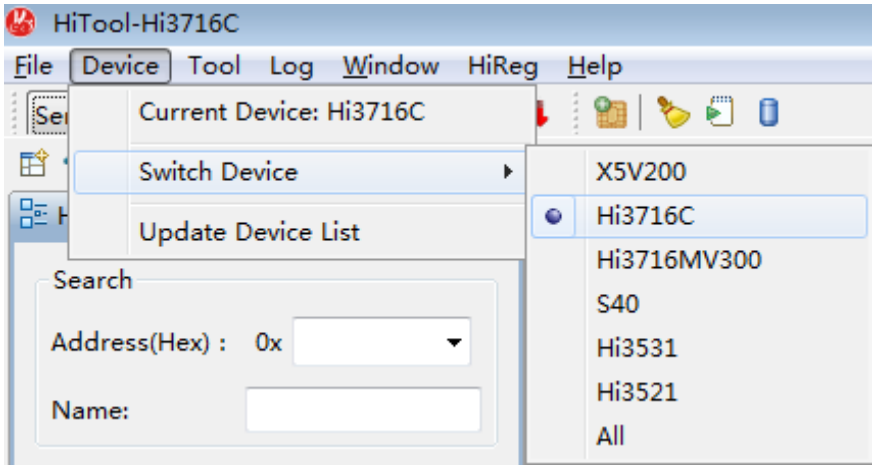


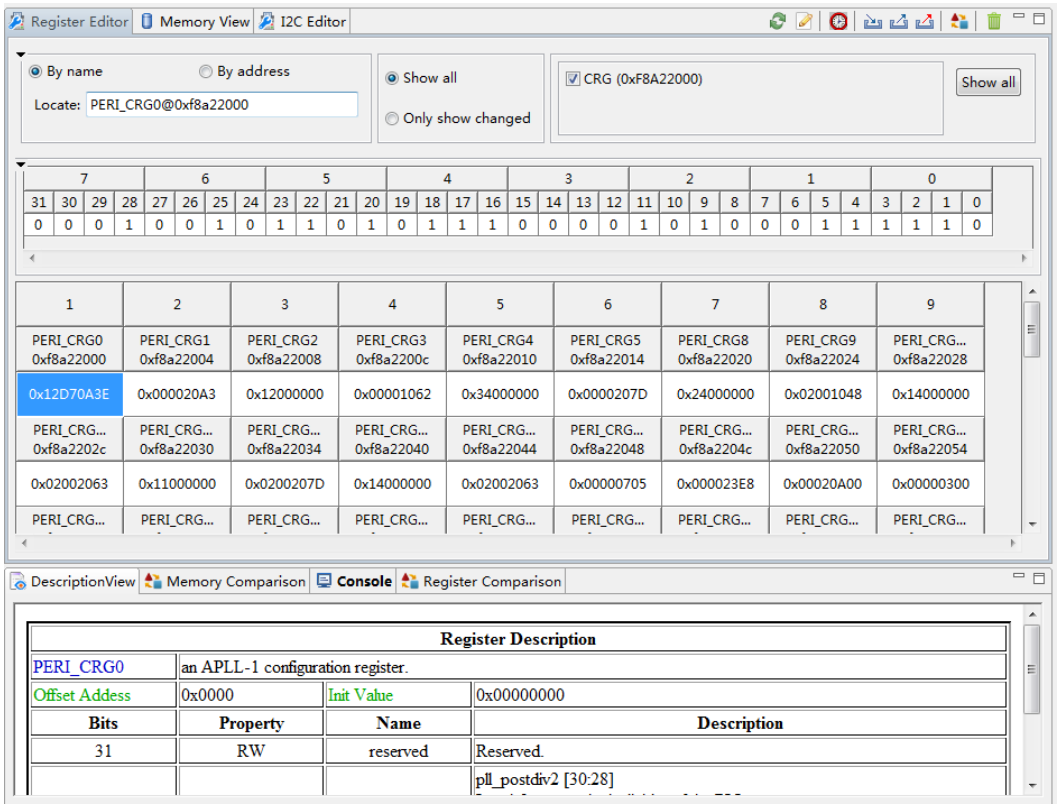
Figure 3-6 Switching a chip



3.3 Register Editor

Figure 3-7z shows **Register Editor**.

Figure 3-7 Register Editor



3.3.1 Toolbar

Figure 3-8 shows the toolbar of **Register Editor**.

Figure 3-8 Register Editor toolbar

The following describes the icons on the toolbar:

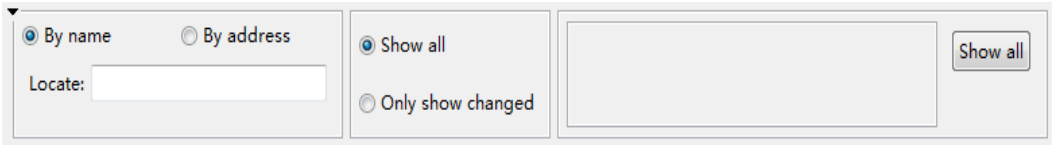
- 
  - Function: reading data
  - Description: Reads data of all registers in the editor from the device and updates the data to the GUI.
- 
  - Function: writing data
  - Description: Writes data of all registers with the write property in the editor to the device.
- 
  - Function: setting the automatic read task
  - Description: Starts/Stops the automatic read task.
- 
  - Function: importing data
  - Description: Imports register data from the register data file to the editor.

  
  - Function: exporting data
  - Description: Exports data of all registers in the editor to a file.
- 
  - Function: exporting modified registers
  - Description: Exports modified registers from the memory to a file.
- 
  - Function: register comparator
  - Description: Opens the file device comparator.
- 
  - Function: removing registers
  - Description: Removes all registers from the editor.

3.3.2 Search Area and Bit Editor

Figure 3-9 shows the search area of **Register Editor**.

Figure 3-9 Search area of Register Editor



- **By name**: Enters a name in the **Locate** text box.
- **By address**: Enters an address in the **Locate** text box.
- **Show all**: Displays data of all registers.
- **Only show changed**: Displays only modified registers.

Figure 3-10 Register bit editor

7				6				5				4				3				2				1				0			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

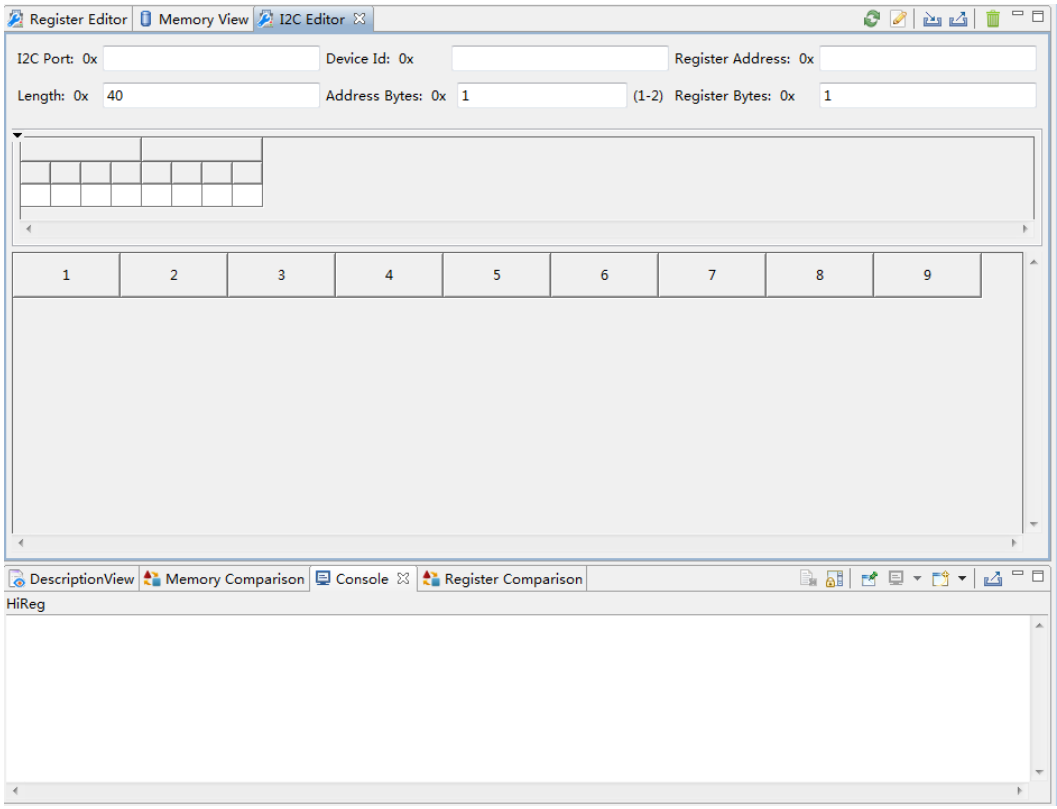
The bit editing area is described as follows:

- Description: bit values corresponding to the current value of the register
- Status: If the register contains members with the write property, it is writable; otherwise, it is read-only.
- Editing mode: When the register has a value, the bit editor is available. When you double-click a bit in the register, the bit value automatically changes between 0 and 1.

3.4 I2C Editor

Figure 3-11 shows **I2C Editor**.

Figure 3-11 I2C Editor



3.4.1 Toolbar

Figure 3-12 shows the toolbar of **I2C Editor**.

Figure 3-12 Toolbar of the I2C Editor

The following describes the icons on the toolbar:

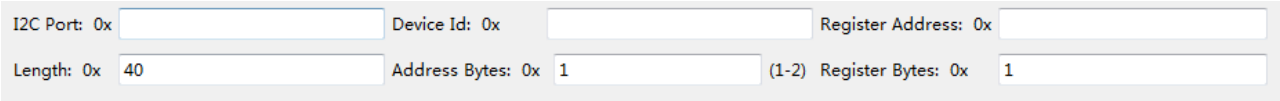
- 
  - Function: reading data
  - Description: Reads data of all registers in the editor and updates the data to the GUI.
- 
  - Function: writing data
  - Description: Writes data of all writable registers in the editor to the device.
- 
  - Function: importing data
  - Description: Imports register data from the register data file to the editor.
- 
  - Function: exporting data

  - Description: Exports data of all registers in the editor to a file.
- 
  - Function: removing registers
  - Description: Removes all registers from the editor.

3.4.2 I2C Device Control Area

Figure 3-13 shows the I²C device control area.

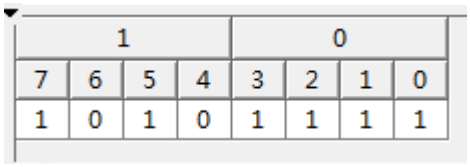
Figure 3-13 I²C device control area



- **I2C Port**: I²C port, a 1-byte hexadecimal character string excluding the prefix 0x
- **Device Id**: device ID, a 1-byte hexadecimal character string excluding the prefix 0x
- **Register Address**: register address, a 4-byte hexadecimal character string excluding the prefix 0x
- **Length**: length, a 1-byte hexadecimal character string excluding the prefix 0x

  After **I2C Port**, **Device Id**, **Register Address**, and **Length** are configured, press **Enter** to add the register to be read to the editor. A register cannot be added repeatedly.
- **Address Bytes**: number of bytes of the register address
- **Register Bytes**: number of bytes of the register

Figure 3-14 Register bit editor



The bit editing area is described as follows:

- Description: bit values corresponding to the current value of the register
- Status: If the register contains members with the write property, it is writable; otherwise, it is read-only.
- Editing mode: When the register has a value, the bit editor is available. When you double-click a bit in the register, the bit value automatically changes between 0 and 1.

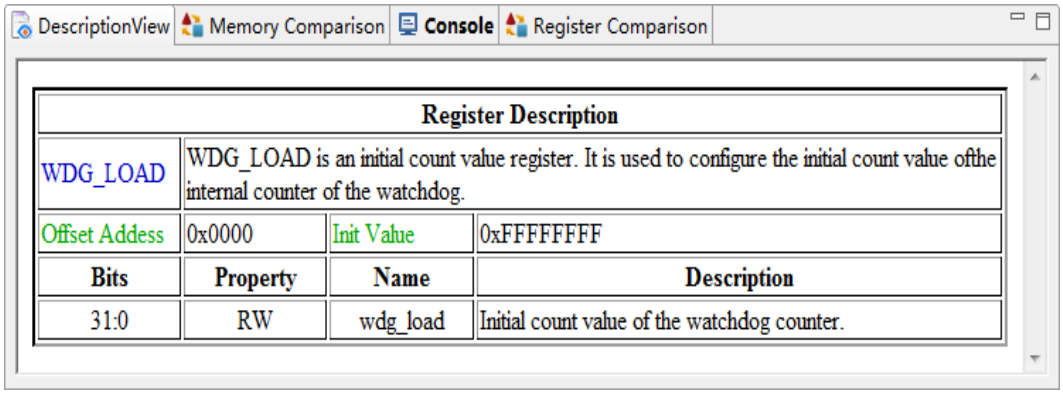
3.4.3 I2C Device Function

The I²C device function is similar to the register editor, except that the I²C device function is used to edit registers of modules that support the I²C function. Registers added to the I²C device contain the device ID and 8-bit values, and you can read and write to the registers.

3.5 Description View

Figure 3-15 shows **Description View**.

Figure 3-15 Description View



After a board is connected successfully and registers are added from **HiReg Explorer** to **Register Editor**, select a register in **Register Editor**. The corresponding description of the register in the data sheet is displayed in **Description View**.

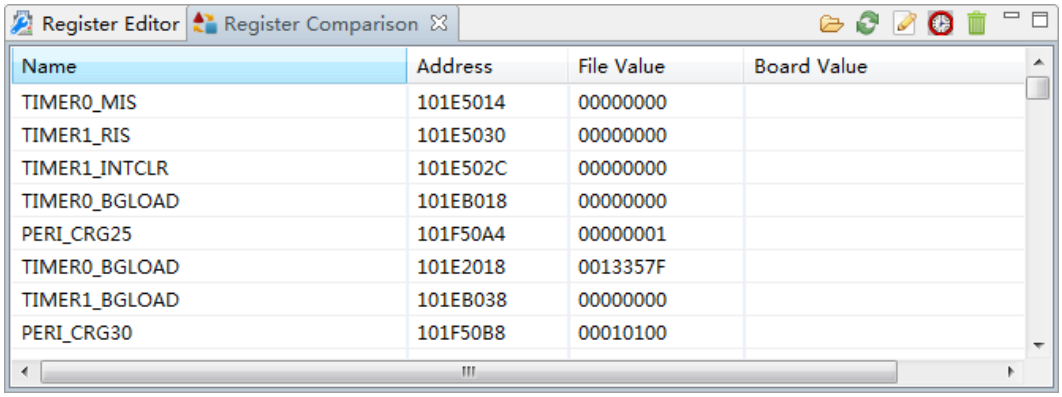
The **Description View** displays the following information about a register:

- Register name
- Description
- Offset address
- Initial value
- Member information
  - Bit information
  - Attribute
  - Member name
  - Description

3.6 Register Comparison

You can import and export register data, and compare the register data of the current chip with exported file data. Figure 3-16 shows the **Register Comparison** view.

Figure 3-16 Register Comparison view



- Importing data

  Clicking on the toolbar imports file data into the register comparator. The format of imported data must be the same as the format of data exported from the register editor.
- Updating data

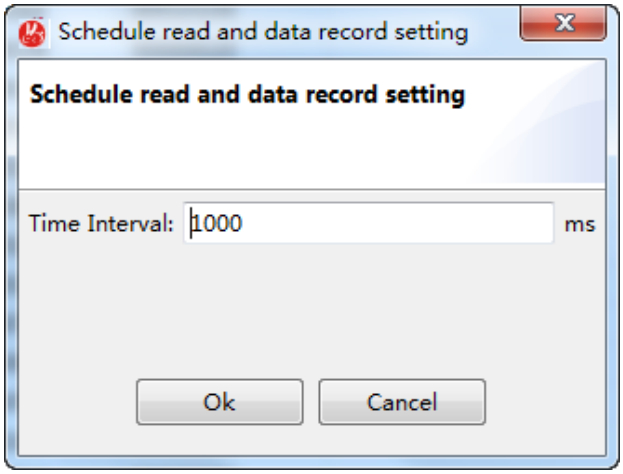
  Clicking on the toolbar reads the values of imported registers in the current device and compares the values with the imported data. The differences are highlighted as red characters with the yellow background. Other buttons on the toolbar are unavailable during this operation.
- Writing all data

  Clicking on the toolbar writes all register values in the imported data to the board. Other buttons on the toolbar are unavailable during this operation.
- Reading data periodically

  When there is data in the comparator, clicking on the toolbar displays the dialog box for setting the scheduled read and data recording parameters, as shown in Figure 3-17. Set the time interval, which cannot be less than 200 ms, and click **OK**. Then the comparator reads register values from the board at the configured interval and compares the read values with imported values until you click again or inconsistency between the read values and the imported values occurs. Other buttons on the toolbar are unavailable during this operation.
- Removing all data

  Clicking on the toolbar removes all imported data in the comparator.

Figure 3-17 Setting the time interval



3.7 Register Debugging Instance

3.7.1 Description

In this instance, a board is connected for reading and writing to registers, importing, exporting, and comparing data. This instance is designed to familiarize you with the uses of the register debugging tool.

3.7.2 Operations

Perform the following steps:

Step 1 Connect the PC for development to a board.

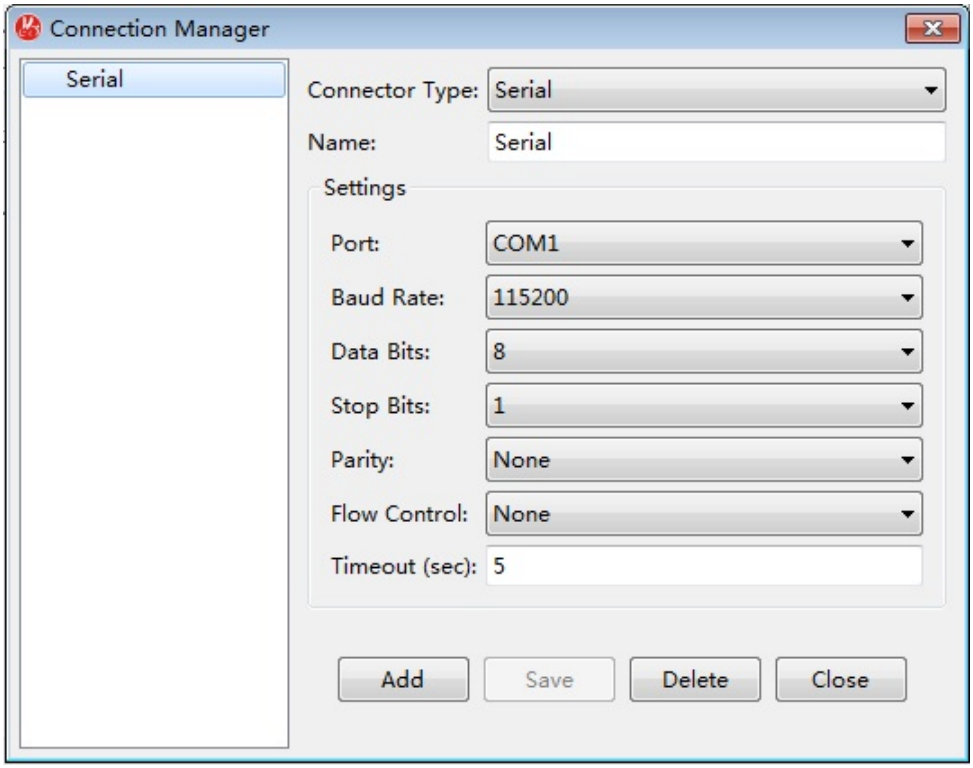
[Serial port connection]

1. Click  on the toolbar.
2. Create a serial port connection, and modify the parameters as follows:
   - **Port**: COM1
   - **Baud Rate**: 115200
   - **Data Bits**: None
   - **Stop Bits**: None
   - **Parity**: None
   - **Flow Control**: None
   - **Timeout (sec)**: 5
3. Click **Add**.
4. Connect the PC for development to the board over the serial port COM1, and power on the board.
5. After the board is started, select the newly created serial port connection from the drop-down list on the toolbar, and click the connect icon  on the toolbar. Then the

HiTool establishes the connection to the board. If the connection is successfully established, the connect icon is dimmed, and the disconnect icon is available, which is used to end the connection to the board. If the connection fails to be established, the system displays a message indicating the connection failure. In this case, check whether the physical connection and the configured connection parameters are correct.

Figure 3-18 shows the dialog box for creating a serial connection.

Figure 3-18 Creating a serial connection



[Network port connection]

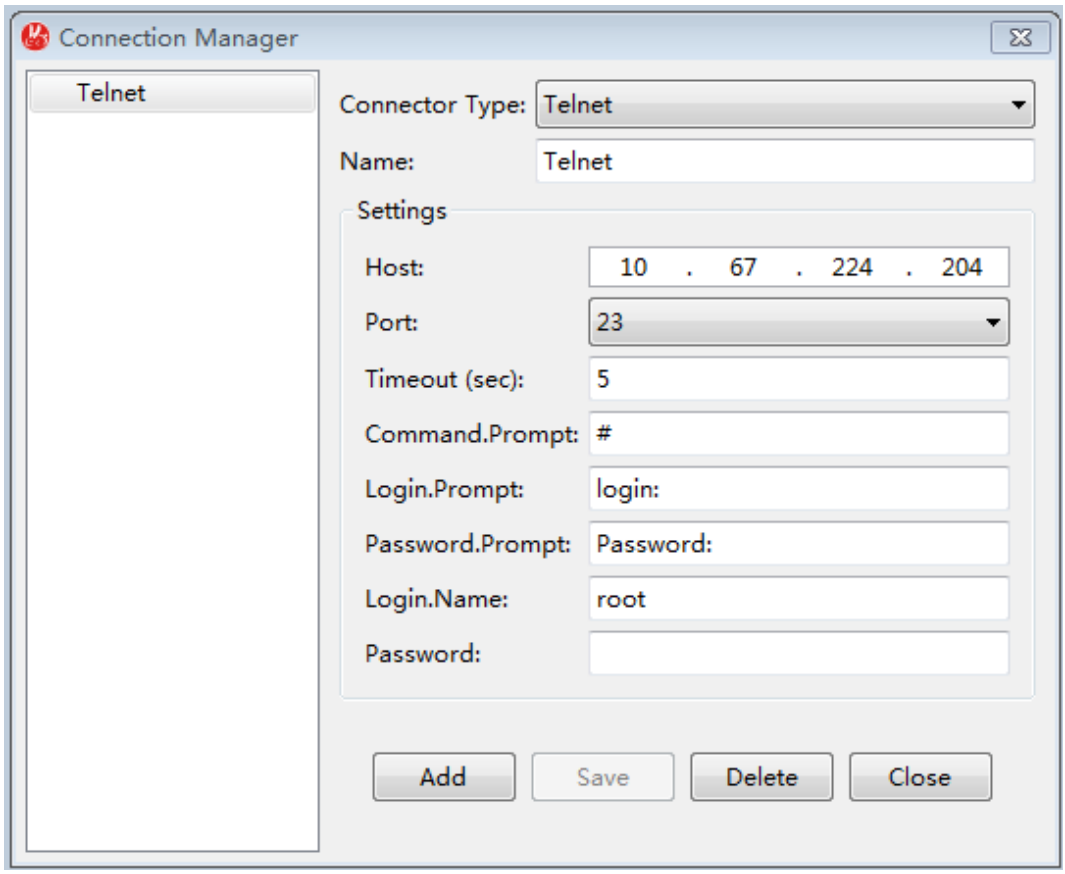
1. Click on the toolbar.
2. Create a telnet connection and modify the parameters as follows:

   Host: 192.168.1.6 (enter the IP address for the connected board)

   Port: 23
3. Click **Add**.
4. Connect the PC for development to the board over the network port, and power on the board.
5. After the board is started, select the newly created telnet connection from the drop-down list on the toolbar, and click the connect icon on the toolbar. Then the HiTool establishes the connection to the board. If the connection is successfully established, the connect icon is dimmed, and the disconnect icon is available, which is used to end the connection to the board. If the connection fails to be established, the system displays a message indicating the connection failure. In this case, check whether the physical connection and the configured connection parameters are correct.

Figure 3-19 shows the dialog box for creating a telnet connection.

Figure 3-19 Creating a telnet connection



Step 2 Check the chip model.

- After the HiReg on the PC connects to the board successfully, the HiReg detects the chip model of the connected board and checks whether the model of the chip on the board is the same as that you select on the tool platform.
- If the chip model matches, you can continue reading and writing to registers. If the chip model mismatches, the system displays a message indicating that the check fails, and you cannot debug registers. In this case, check the chip model and try again.

Step 3 Query and add registers, and view register description information.

- Querying registers

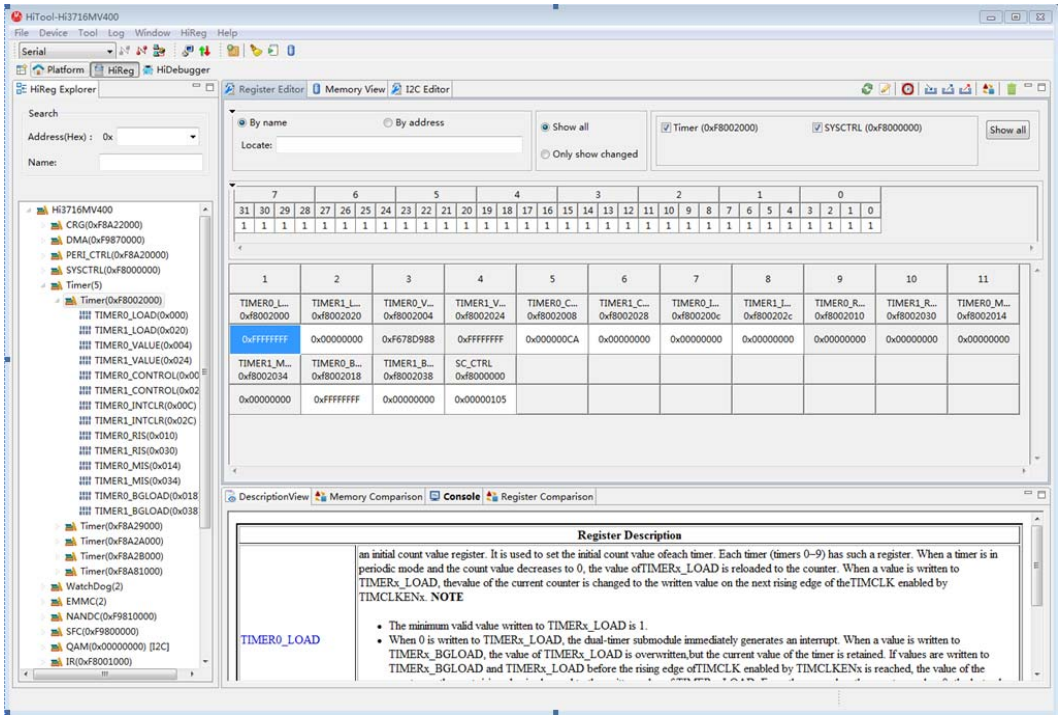
  After the chip model is verified, the **HiReg Explorer** view displays the tree structure of the chip. You can then search for a register by name or address.
- Adding registers

  You can add a specific register or add multiple registers in batches to the register editor. You can read and write to registers only in the register editor.
- Viewing information about a register

  After you click a register in the register editor, information about the register is displayed in the description view.

  See Figure 3-20.

Figure 3-20 Querying and adding registers, and viewing register description information



Step 4 Edit, read, and write register data.

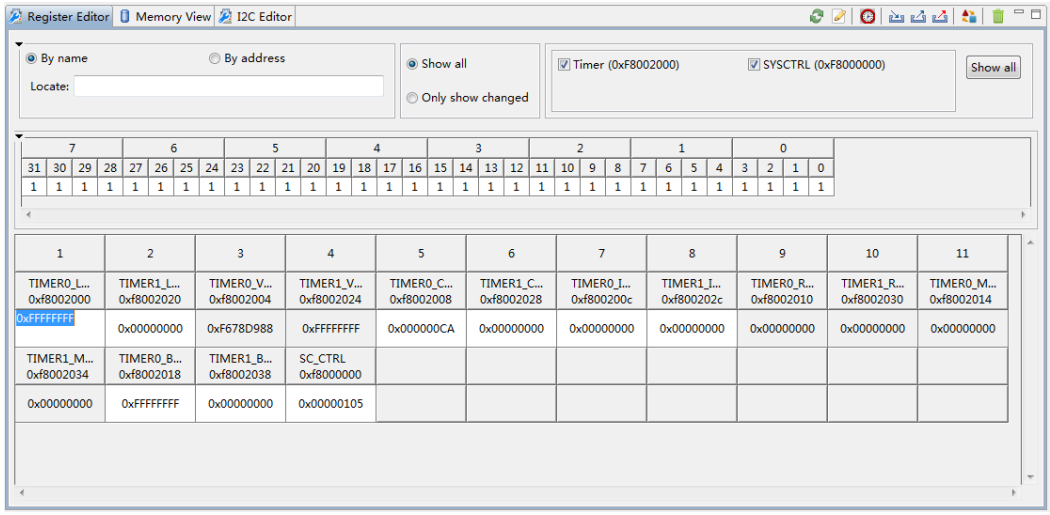
- Editing register data
  - Method 1: Select and read a register in the register editor, and then double-click the cell below the editor to directly edit the register value. (Only data in the memory of the PC is edited at this time. The value is written to the board only after the write operation is performed.)
  - Method 2: Select a register, and double-click a cell in the bit editor to edit the register data by bit. If a value is modified, it turns into red.
- Reading and writing to registers
  - Reading registers

    After selecting a register in the register editor, you can read the register or multiple registers in batches by using the shortcut menu.
  - Writing to registers

    You can write to a single register cell or multiple cells in batches in the editor.

  See Figure 3-21.

Figure 3-21 Editing, reading, and writing register data



Step 5 Import and export register data.

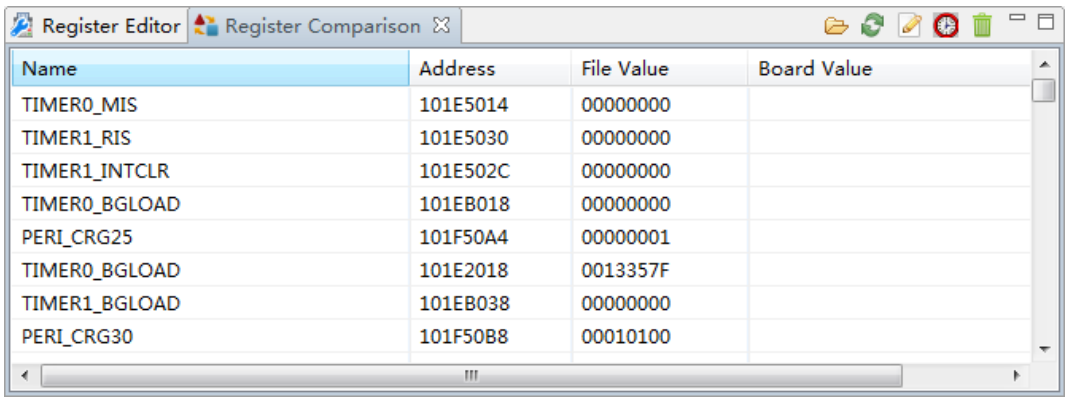
Clicking the export icon on the HiReg toolbar exports data in the current register editor to the .data file. Clicking the import icon on the toolbar imports data in the .data file to the register editor.

Step 6 Compare register data.

Import register data from a file to the register comparator, click , read register data from the board to the register comparator, and then compare the register data from the file with that from the board. If the register values obtained from the board mismatch those from the file, the mismatched values are highlighted. You can select a mismatched value from the board and write the corresponding value from the file to the board by using the shortcut menu; or you can click on the toolbar to write all mismatched values from the file to the board. Note that not all register values can be written to the board successfully because some registers cannot be modified.

See Figure 3-22.

Figure 3-22 Comparing register data

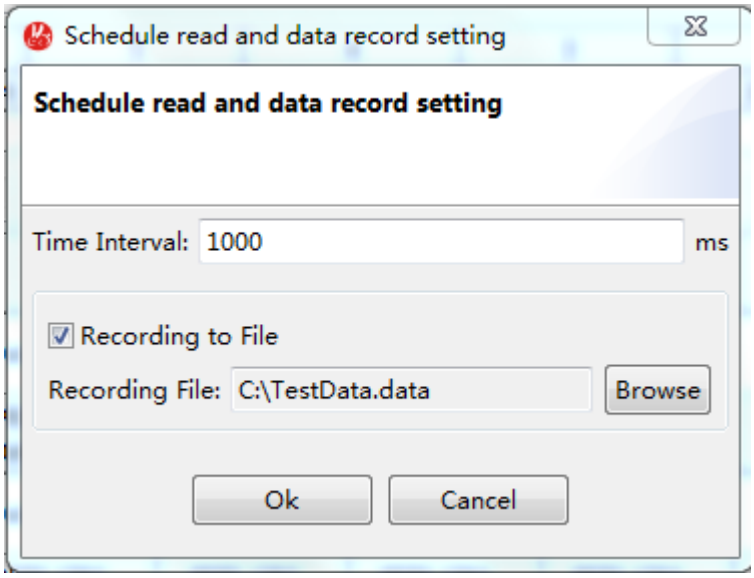


Step 7 Read registers periodically and record register data.

Click  on the toolbar of the register editor, set **Time Interval** to 300 ms, select **Recording to File**, and set **Recording File** to **C:\TestData.data**, and start recording data periodically. You can observe from the console that data is being read and updated in the editor. Data is being written to the file for storing recorded data, and the file size is gradually increasing.

See Figure 3-23.

Figure 3-23 Setting the time interval

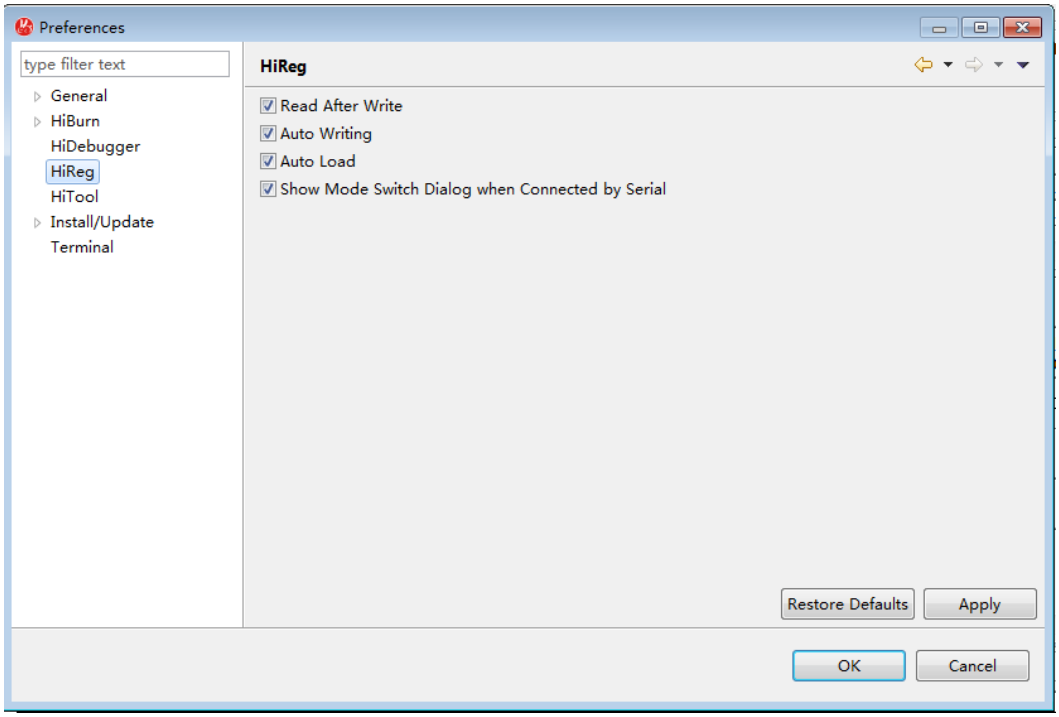


Step 8 Set parameters in the **Preferences** dialog box.

- **Read After Write**: If this option is selected, a read operation is performed after each write operation to update the data.
- **Auto Writing**: If this option is selected, the new values are automatically written to the board after modification.
- **Auto Load**: If this option is selected, data is loaded from the board when registers in the HiReg Explorer are added to the register editor.

See Figure 3-24.

Figure 3-24 Preferences



----End

4 Debugging the Memory

4.1 Functions

The HiReg can be used to debug the DDR memory data. It has the following functions:

- Supports two connection modes: serial port connection and network port connection.
- Detects the chip model.

  The memory view can be used no matter whether the model of the chip on the connected board matches the selected chip model.
- Supports multiple memory views.

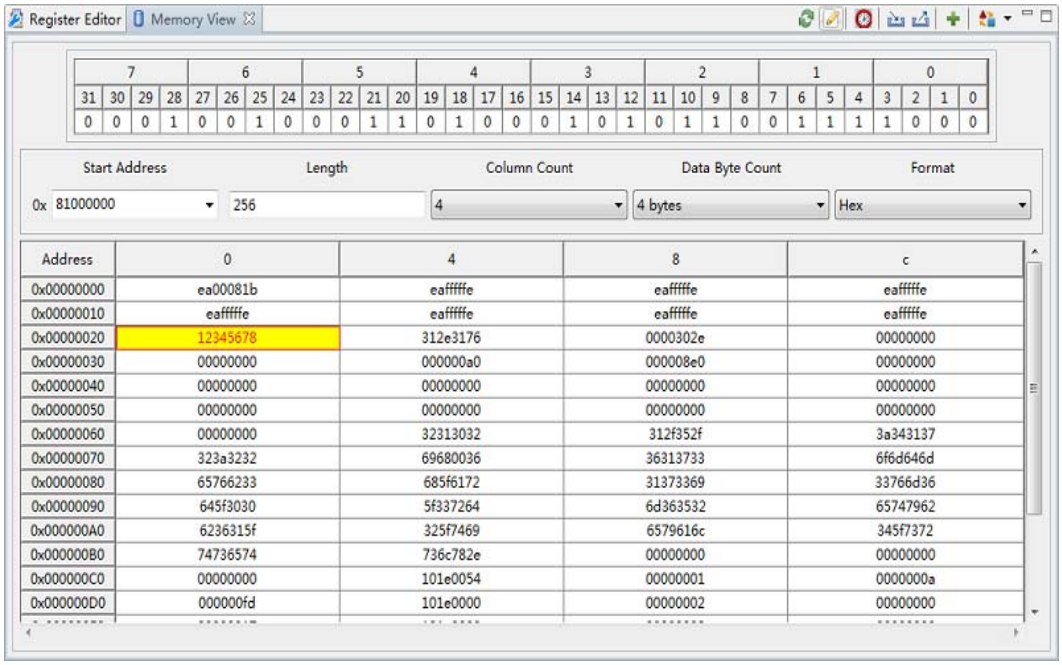
  Multiple memory views can be added for editing and comparing data blocks of various memory addresses.
- Reads and writes memory data.

  One or more memory data blocks can be read or written at a time, and memory data can be read periodically.
- Views and edits memory data.
- Exports, imports, and compares memory data.
- Supports the console output.

4.2 Memory View

Figure 4-1 shows the memory view.

Figure 4-1 Memory view



4.2.1 Toolbar

Figure 4-2 shows the toolbar of the memory view.

Figure 4-2 Toolbar of the memory view

The following describes the icons on the toolbar:

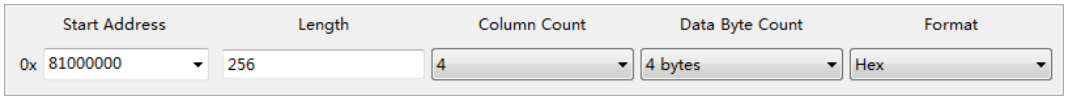
- 
  - Function: reading data
  - Description: Reads data from the board.
- 
  - Function: reading data periodically
  - Description: Starts or stops the scheduled read task.
- 
  - Function: writing data
  - Description: Writes data in the editor to the board.
- 
  - Function: importing data
  - Description: Imports memory data in a file to the editor.
- 
  - Function: exporting data

  - Description: Exports data in the editor to a file.
- 
  - Function: adding memory views
  - Description: Adds a memory view.
- 
  - Function: memory comparison
  - Description: Opens the memory comparator.

4.2.2 Data Definition Area

Figure 4-3 shows the data definition area of the memory view.

Figure 4-3 Data definition area of the memory view



- Start Address
  - Description: start address of data to be read (there is no initial value)
  - Format: hexadecimal number (prefix 0x excluded)
- Length
  - Description: length of data to be read (the initial value is **256**)
  - Format: decimal number, 4-byte-aligned. It is automatically adjusted if it is not aligned. For example, if you enter **1**, it is adjusted to 4; if you enter **7**, it is adjusted to 8.
- Column Count
  - Description: absolute address of the register (the initial value is **4**)
  - Format: decimal number (1, 2, 4, 8, or 16)
- Data Byte Count
  - Description: number of bytes allowed in the data unit of the editor (the initial value is **4**)
  - Format: decimal number (1, 2, or 4)
- Format
  - Description: bit format corresponding to the current value of the register (the initial format is hexadecimal)
  - Format: character strings, hexadecimal or decimal

4.2.3 Data Editing Area

Figure 4-4 shows the area for editing memory data. It displays memory data of the target board based on the configured address and data length. The display format can be specified, including the number of columns, and number of bytes in each column.

Figure 4-4 Data editing area

Address	0	4	8	c
0x00000000	80010060	90240300	000810c0	92a00480
0x00000010	04020000	04081001	04202410	00840200
0x00000020	8540c040	00002008	40411800	00000840
0x00000030	54080120	0c00a200	00240010	10002100
0x00000040	40a20404	00400080	e0008080	00040082
0x00000050	10120002	00100000	08020000	20000002
0x00000060	01480400	0000104c	02400040	07080004
0x00000070	18089122	08001000	00202022	00000020
0x00000080	06000000	00000008	00024344	04000100
0x00000090	10380100	00001000	0c002002	10005021
0x000000A0	08000004	00000004	80628008	00480040
0x000000B0	a0483202	08040004	80202001	04143021
0x000000C0	c4000020	4000c401	40028004	000080c0
0x000000D0	80300008	20100202	04020240	00848001
0x000000E0	80818000	00004000	01088040	00031804
0x000000F0	3e001a00	00000000	20a01161	10080800

CAUTION

- The format of the entered value must be consistent with the selected data format.
- If **Hex** is selected, you need to enter a hexadecimal value, and the length of entered data cannot be greater than that of the original data. If the length of entered data is less than that of the original data, the upper bits are automatically stuffed with 0s.
- If **Dec** is selected, you need to enter a decimal value. If the entered value is too large, it is automatically changed to an appropriate value, which is related to the number of bytes.

4.2.4 Bit Editing Area

The memory editor allows you to view and edit the value of a data cell by bit. Select the memory cell to be viewed and edited. The bit editor automatically displays the content of the memory cell.

Figure 4-5 Bit editor

7				6				5				4				3				2				1				0			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0

- Double-click a bit value. The bit value automatically changes between 0 and 1 and is updated to the data area.
- Double-click a cell. The editor is activated. Then enter the corresponding value and press **Enter** to complete editing or press **Esc** to cancel the editing.

4.2.5 Operations

- Reading memory data

To read the DDR memory data, press **Enter** or click on the toolbar of the memory view after entering the start address.

- Writing memory data

  To write the DDR data to the board, click on the toolbar.

- Reading memory data periodically

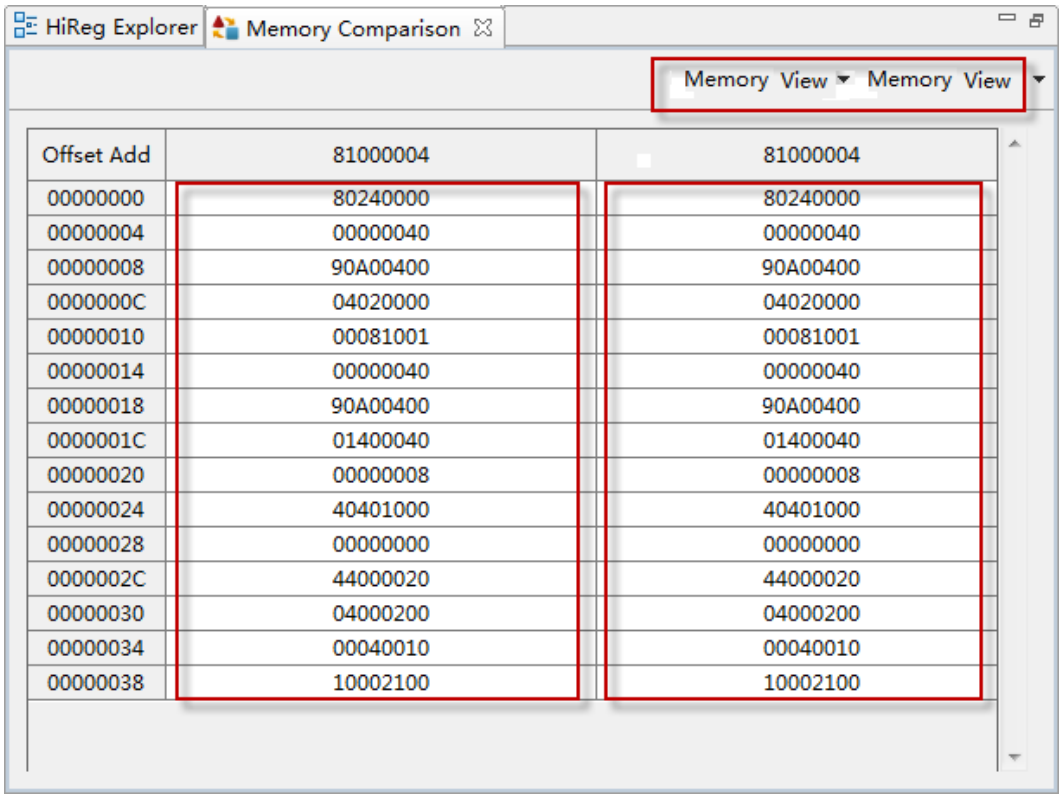
  The method of reading DDR memory data periodically is similar to that of reading registers periodically.

- Changing the format
  - Length: Specify **Length** (65536 at the maximum) in the data definition area, and press **Enter**.
  - Number of columns: Specify **Column Count** in the data definition area.
  - Number of bytes: Specify **Data Byte Count** in the data definition area.
  - Data format: Specify **Format** in the data definition area.

4.3 Memory Comparison

Figure 4-6 shows the **Memory Comparison** view.

Figure 4-6 Memory Comparison view



Offset Add	81000004	81000004
00000000	80240000	80240000
00000004	00000040	00000040
00000008	90A00400	90A00400
0000000C	04020000	04020000
00000010	00081001	00081001
00000014	00000040	00000040
00000018	90A00400	90A00400
0000001C	01400040	01400040
00000020	00000008	00000008
00000024	40401000	40401000
00000028	00000000	00000000
0000002C	44000020	44000020
00000030	04000200	04000200
00000034	00040010	00040010
00000038	10002100	10002100

This tool is designed for comparing memory data. Therefore, the **Memory Comparison** view must work with the **Memory View**.

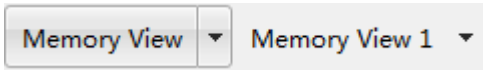
CAUTION

Data with the same offset address is compared. The start address is not related.

4.3.1 Toolbar

Figure 4-7 shows the toolbar of the **Memory Comparison** view.

Figure 4-7 Toolbar of the Memory Comparison view

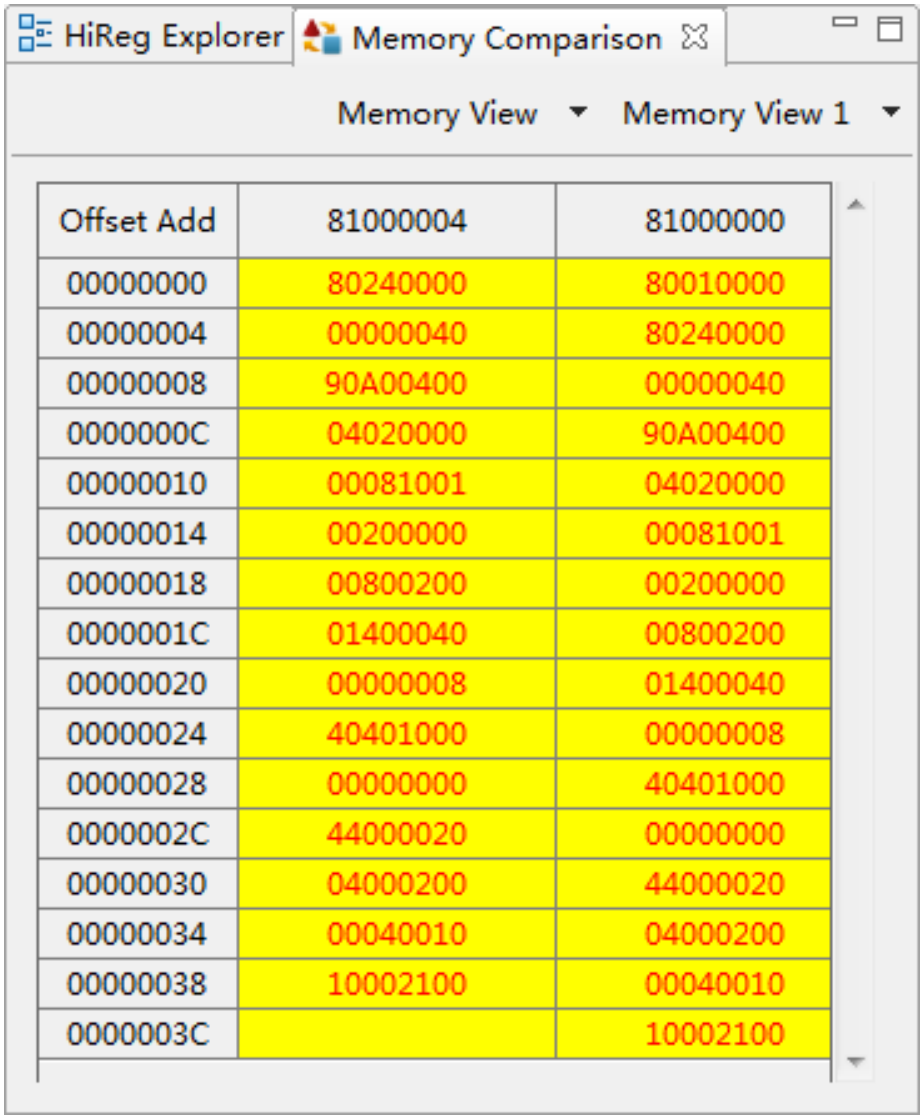


You can choose two same or different memory views on the toolbar for memory comparison.

4.3.2 Data Area

The inconsistencies between memory views are highlighted.

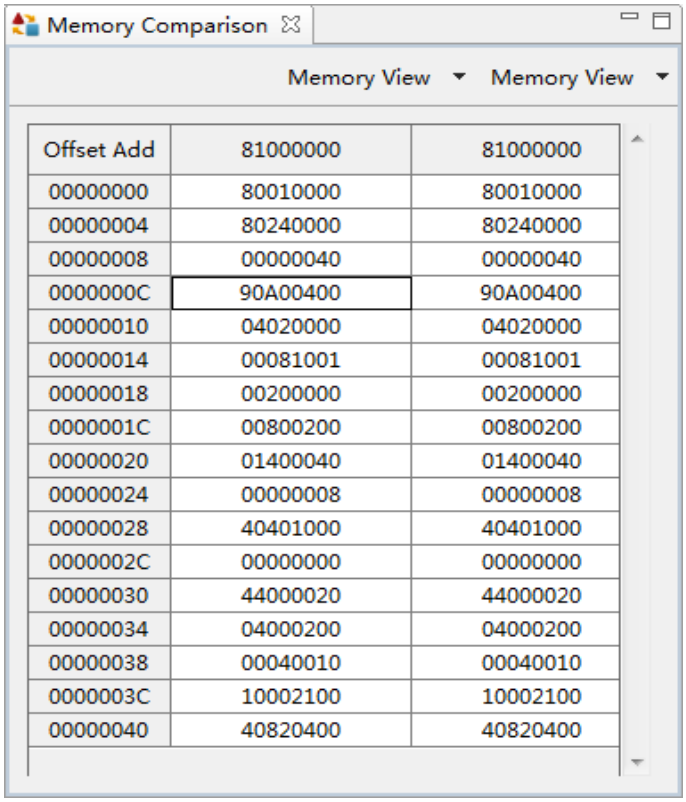
Figure 4-8 Memory comparison (inconsistency)



Offset Add	81000004	81000000
00000000	80240000	80010000
00000004	00000040	80240000
00000008	90A00400	00000040
0000000C	04020000	90A00400
00000010	00081001	04020000
00000014	00200000	00081001
00000018	00800200	00200000
0000001C	01400040	00800200
00000020	00000008	01400040
00000024	40401000	00000008
00000028	00000000	40401000
0000002C	44000020	00000000
00000030	04000200	44000020
00000034	00040010	04000200
00000038	10002100	00040010
0000003C		10002100

If data in two memory views is the same, the data format is not changed.

Figure 4-9 Memory comparison (the same)



Offset Add	81000000	81000000
00000000	80010000	80010000
00000004	80240000	80240000
00000008	00000040	00000040
0000000C	90A00400	90A00400
00000010	04020000	04020000
00000014	00081001	00081001
00000018	00200000	00200000
0000001C	00800200	00800200
00000020	01400040	01400040
00000024	00000008	00000008
00000028	40401000	40401000
0000002C	00000000	00000000
00000030	44000020	44000020
00000034	04000200	04000200
00000038	00040010	00040010
0000003C	10002100	10002100
00000040	40820400	40820400

4.4 Memory Debugging Instance

4.4.1 Description

In this instance, a board is connected for reading, writing, importing, exporting, and comparing memory data. This instance is designed to familiarize you with the uses of the memory debugging tool.

4.4.2 Operations

Perform the following steps:

Step 1 Connect the PC for development to a board.

[Serial port connection]

1. Click  on the toolbar.
2. Create a serial port connection and modify the parameters as follows:
   - **Port**: COM1
   - **Baud Rate**: 115200
   - **Data Bits**: None
   - **Stop Bits**: None

- **Parity**: None
- **Flow Control**: None
- **Timeout (sec)**: 5

3. Click **Add**.
4. Connect the PC for development to the board over the serial port COM1, and power on the board. After the board is started, select the newly created serial port connection from the drop-down list on the toolbar, and click the connect icon on the toolbar. Then the HiTool establishes the connection to the board. If the connection is successfully established, the connect icon is dimmed, and the disconnect icon is available, which is used to end the connection to the board. If the connection fails to be established, the system displays a message indicating the connection failure. In this case, check whether the physical connection and the configured connection parameters are correct.

[Network port connection]

1. Click on the toolbar.
2. Create a telnet connection and modify the parameters as follows:
   - **Host**: 192.168.1.6
   - **Port**: 23
3. Click **Add**.
4. Connect the PC for development to the board over the network port, and power on the board.
5. After the board is started, select the newly created telnet connection from the drop-down list on the toolbar, and click the connect icon on the toolbar. Then the HiTool establishes the connection to the board. If the connection is successfully established, the connect icon is dimmed, and the disconnect icon is available, which is used to end the connection to the board. If the connection fails to be established, the system displays a message indicating the connection failure. In this case, check whether the physical connection and the configured connection parameters are correct.

Step 2 Check the chip model.

After the HiReg on the PC connects to the board successfully, the HiReg checks the chip model of the connected board and checks whether the model of the chip on the board is the same as that you select on the tool platform. If the chip model matches, you can go on reading and writing to the memory. If the chip model mismatches, the system displays a message indicating that the check fails. In this case, check the chip model and try again.

Step 3 Add memory data to the editing area.

Enter the start address and length of the memory data to be added to the editing area, and press **Enter**. The memory data with the specified length at the specified start address is then displayed in the memory editor.

Step 4 Editing, reading, and writing memory data.

- Editing memory data
  - Method 1: Select and read a memory data value in the memory editor, and then double-click the cell below the editor to directly edit the memory value. (Only data in the memory of the PC is edited at this time. The value is written to the board only after the write operation is performed.)

  - Method 2: Select the memory data, and double-click a cell in the bit editor to edit the memory data by bit.
- Reading and writing memory data
  - Reading memory data

    Click  on the toolbar. The HiReg reads data from the board and then updates the memory data that is added to the memory editor.
  - Writing memory data

    Click  on the toolbar. The memory data in the memory editor is written to the board. If the automatic write function is enabled, modified data is automatically written to the corresponding memory address of the board after a cell is edited.

Step 5 Import and export memory data.

Clicking the export icon on the HiReg toolbar exports data in the current register editor to the .data file. Clicking the import icon on the toolbar imports data in the .data file to the memory editor.

Step 6 Compare memory data.

If there are multiple memory views, you can compare data in any two memory views in the **Memory Comparison** view. Click the two buttons on the upper right corner of the **Memory Comparison** view to specify the two memory views to be compared. Inconsistencies are highlighted as red characters with yellow background.

Step 7 Read memory data periodically and record memory data.

Click  on the toolbar of the memory editor, set the read time interval to 300 ms and click **OK**. You can observe from the console that data is being read and updated in the editor.

Step 8 Set parameters in the **Preferences** dialog box.

- **Read After Write**: If this option is selected, a read operation is performed after each write operation to read the written data.
- **Auto Writing**: If this option is selected, the new values are automatically written to the board after modification.
- **Auto Load**: If this option is selected, data is loaded from the board when memory data is added to the memory editor.

----End

5 Debugging in U-Boot Mode

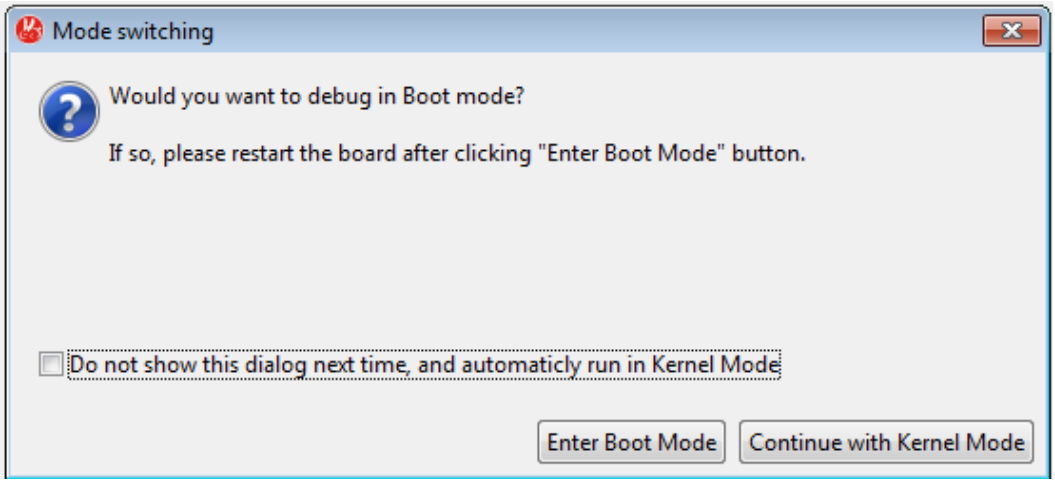
5.1 Overview

The HiReg allows you to debug registers in U-boot mode as well as in kernel mode. However, only serial port connections are supported for debugging registers in U-boot mode. Meanwhile, the I^2C is not supported for debugging in U-boot mode.

5.2 Switching to the U-Boot Mode

When the connection manager of the platform is used to establish a serial port connection, the **Mode switching** dialog box is displayed, asking whether you want to debug registers in the U-boot mode (this dialog box is not displayed if **Show Mode Switch Dialog when Connected by Serial** is deselected in the **Preferences** dialog box of the HiReg).

Figure 5-1 Switching to the U-boot mode



Click **Enter Boot Mode** to debug registers in U-boot mode.

CAUTION

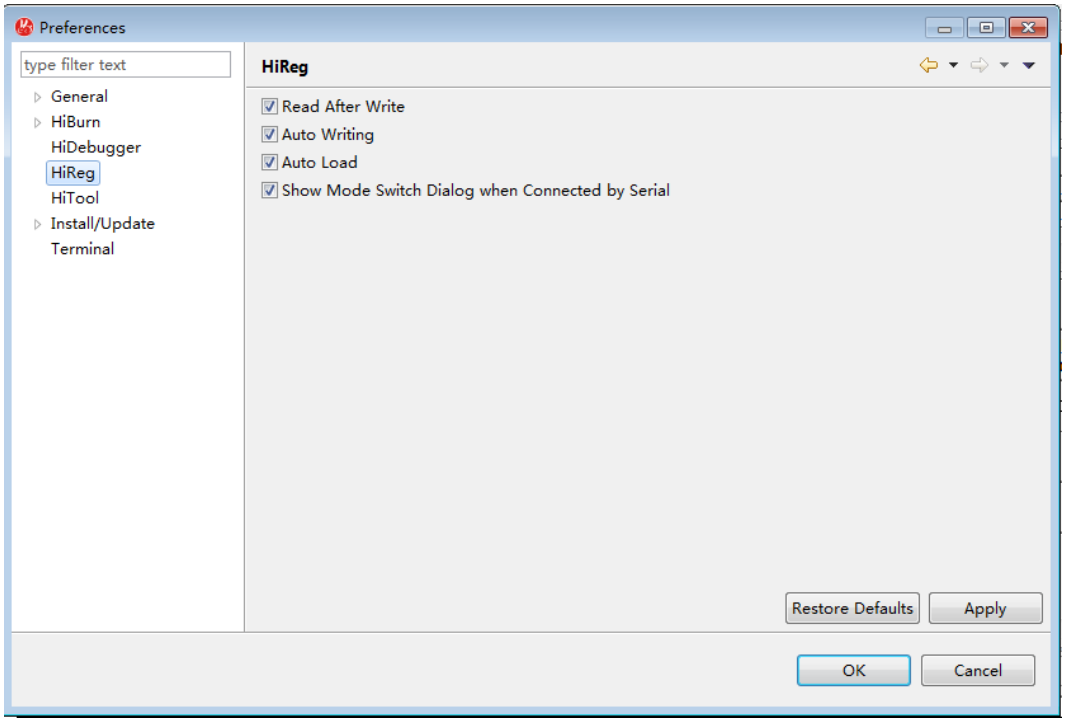
After you click **Enter Boot Mode**, the board is restarted. Follow instructions of the HiReg console.

5.3 Displaying the Mode Switch Dialog Box

If you need the debugging function in only kernel mode, you can select **Do not show this dialog next time, and automatically run in Kernel Mode** in the **Mode switching** dialog box. Then the system does not display the **Mode switching** dialog box again when you establish a serial port connection.

If you want the **Mode switching** dialog box to be displayed, choose **Window** > **Preferences**, click **HiReg**, select **Show Mode Switch Dialog when Connected by Serial**, and click **OK**. See Figure 5-2.

Figure 5-2 Preferences dialog box



6 FAQs

6.1 What Do I Do If the Serial Port Cannot Be Connected?

Problem Description

After a serial port connection is successfully configured and saved by clicking  on the toolbar, and the connect icon is clicked, the serial port cannot be connected.

Solution

To locate the problem, perform the following steps:

Step 1 Check whether the physical connection between the serial ports of the PC and the board is normal, and whether the ports are correct.

Step 2 Check whether the connection parameters are correctly configured.

Step 3 Restart the board, and try again.

----End

6.2 What Do I Do If the Network Port Cannot Be Connected?

Problem Description

After a network port connection is successfully configured and saved by clicking  on the toolbar and the connect icon is clicked, the network port cannot be connected.

Solution

To locate the problem, perform the following steps:

Step 1 Check whether the physical connection between the PC and the board is normal, and whether the IP address is correctly configured and can be pinged.

Step 2 Check whether the connection parameters are correctly configured.

Step 3 Restart the board, and try again.

----End

6.3 What Do I Do If the Selected Chip Model and Actual Chip Model Are Mismatched?

Problem Description

After a chip is selected on the main GUI of the HiTool and the HiReg is opened, the system displays a message in the HiReg explorer view indicating that the chip model of the board and the selected chip model are mismatched.

Solution

Check whether the selected chip model is the same as the actual chip model on the board. You can view the selected chip model in the HiTool by choosing **Device** > **Current Device**.

A Acronyms and Abbreviations

D

DDR	double data rate

I

I^2C	inter-integrated circuit
IP	Internet Protocol

P

PC	personal computer