MT793X IoT SDK for Easy PinMux Tool User Guide

Version: 1.0
Release date: 2020-12-04

Version History

Version	Date	Description
1.0	2020-12-04	Official release

Table of Contents

List of Figures

1 Overview

MediaTek Easy PinMux Tool (EPT) is a convenient and user-friendly graphical user interface (GUI) to configure pin multiplexor (PinMux) and supported driver settings for MediaTek chipsets, including the MT793X and MT768X. The tool provides modes and options for each PinMux and enables customized settings for input and output (I/O) characteristics according to design requirements.

Once configured, all settings can be saved as a workspace file that can be reloaded to apply the preconfigured tool settings. The results can also be output as C header and source files.
The analog-to-digital converter (ADC) pins are controlled by the ADC driver, not by the EPT. For more information about how to configure the ADC pins, please refer to the ADC module in the Hardware Abstraction Layer (HAL) section of MediaTek IoT SDK for RTOS API Reference Manual.

1.1 Environment

The EPT can be used on Windows (32- or 64-bit editions, Windows XP, Vista, 7, 8 and 10) and Linux (Ubuntu 32- or 64-bit, Ubuntu version 14.04 and higher).

1.2 Configuring Your Device with the EPT

To use the tool:

1) Launch the executable (ept.exe) under the EPT package folder. Create a new workspace or open an existing one, and edit it according to requirements. See Section 1.2.1, “Workspace File Options”, for more details.
2) Apply user settings. See Section 2, “Driver Settings”, for more details on how to configure the I/O parameters and PinMux settings.
3) Go to the **Gen** menu and click **GenCode** to generate the source code of the driver stored under \output\7933(768x) folder of the tool. The main GUI is shown in Figure 1 and Figure 2.

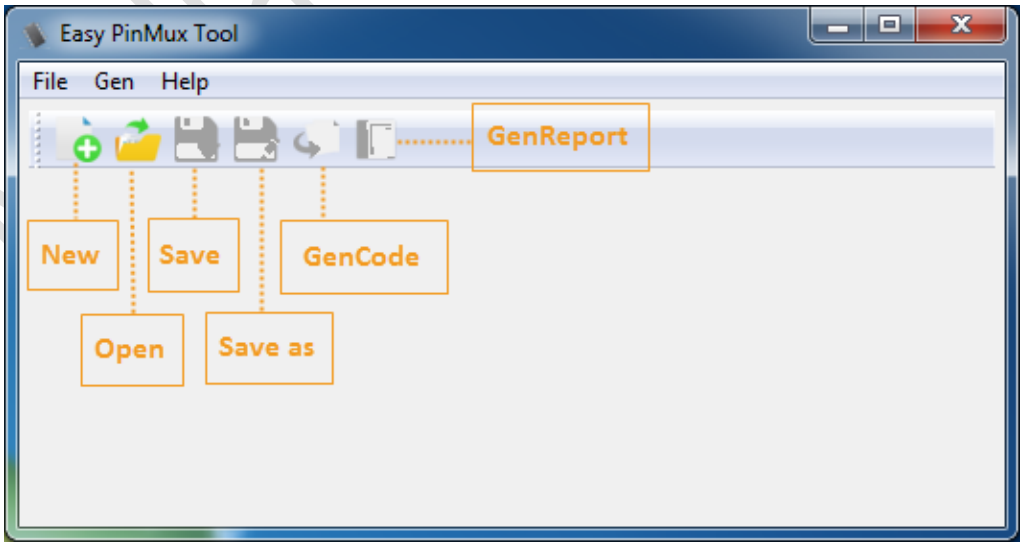

Figure 1. Main UI of the EPT with an empty workspace

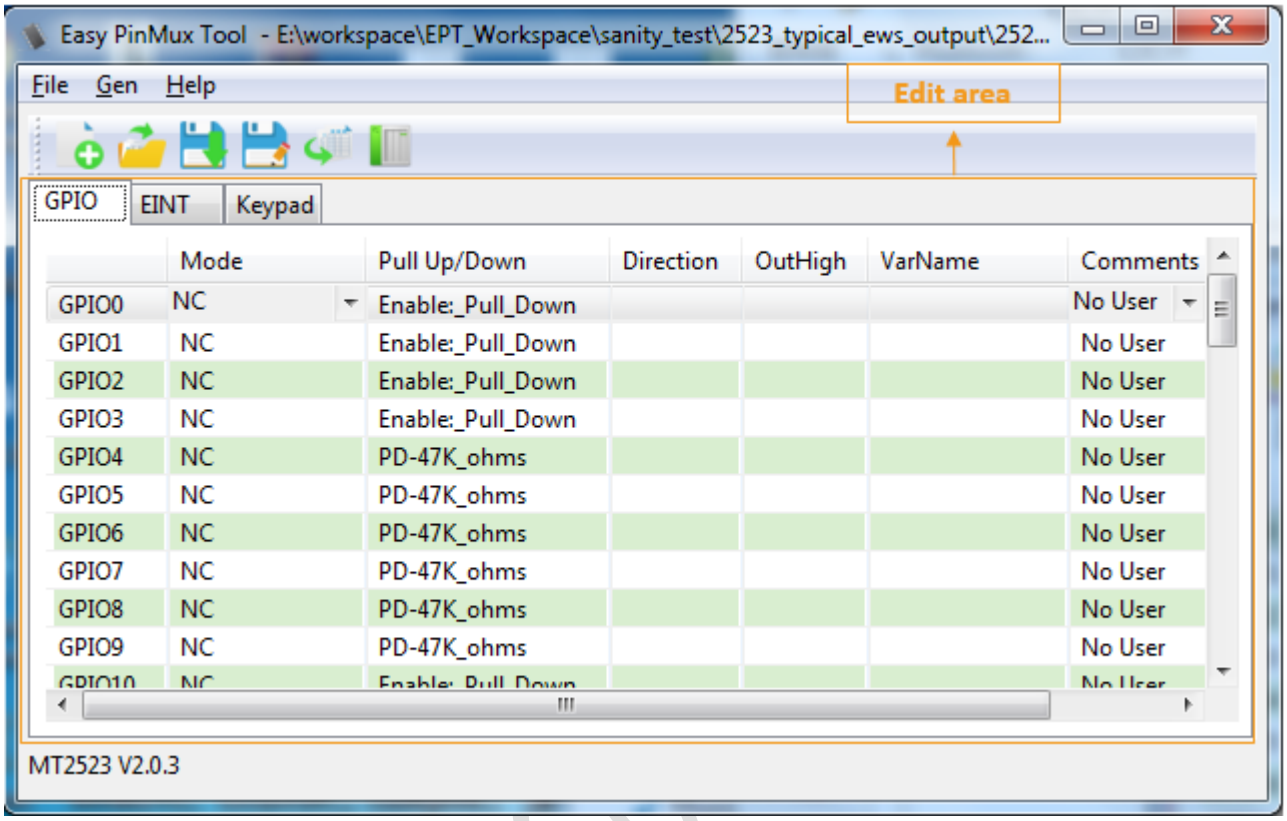

Figure 2. Main UI of the EPT with a workspace

1.2.1 Workspace File Options

To create a new workspace:

1) Go to the **File** menu, click **New**, and select the chipset under **Chip Selection** to create an empty workspace, as shown in Figure 3.

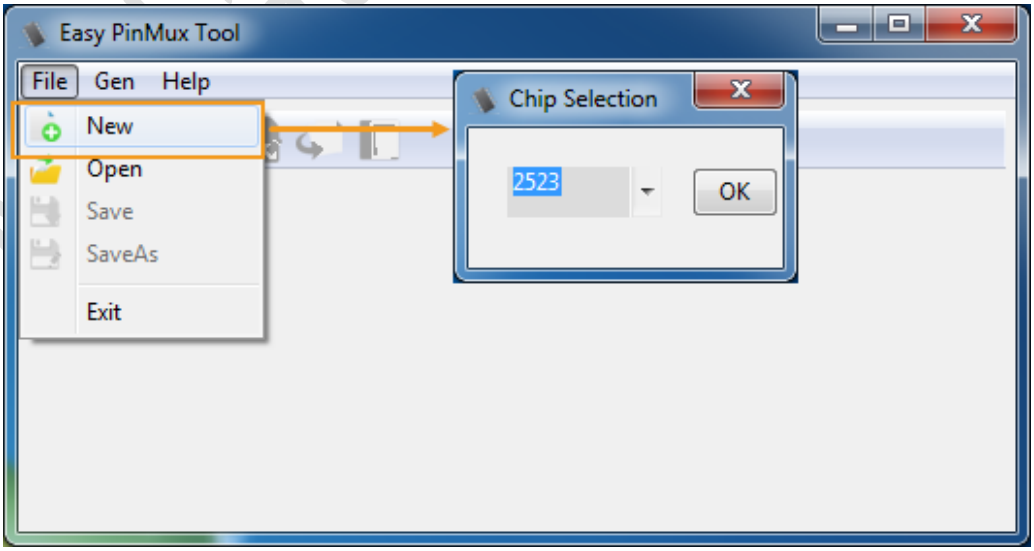

Figure 3. Create a new workspace

2) Save the workspace as a .ews file by clicking **Save** or **SaveAs** on the **File** menu.

To open an existing workspace:

1) Go to the **File** menu and click **Open** to open an existing workspace. You can use your own workspace file or a demo workspace file with the extension .ews or .dws provided by MediaTek. The demo workspace file (.ews or .dws) is located under:

```
<sdk_root>\project\<board>\apps\<application>\ept_ews
```

To edit the workspace:

1) When an existing workspace is opened or a new workspace is created, you can configure its parameters by choosing an item on the dropdown menu according to your requirements, as shown in Figure 4.

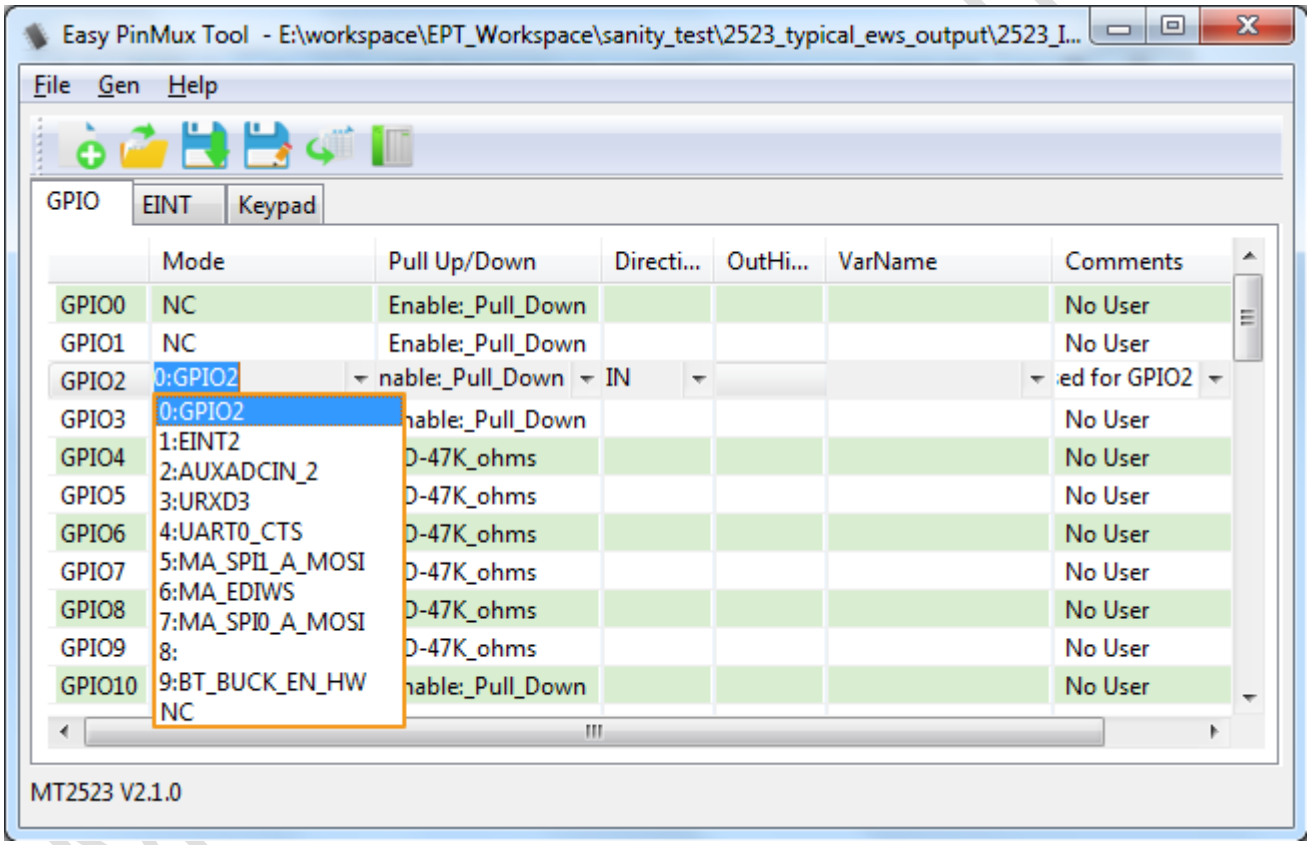

Figure 4. Edit a workspace

2) Save the workspace as a .ews file.

To generate the source code for the driver:

1) Go to the **Gen** menu and click **GenCode** to generate the source code for the driver. All generated files are saved in \output\7933(768x) folder. The header files (.h files) are saved in the sub-directory inc, while the source files (.c files) are saved in the sub-directory src. Once the source code is generated successfully, a popup message will prompt the file path, as shown in Figure 5.

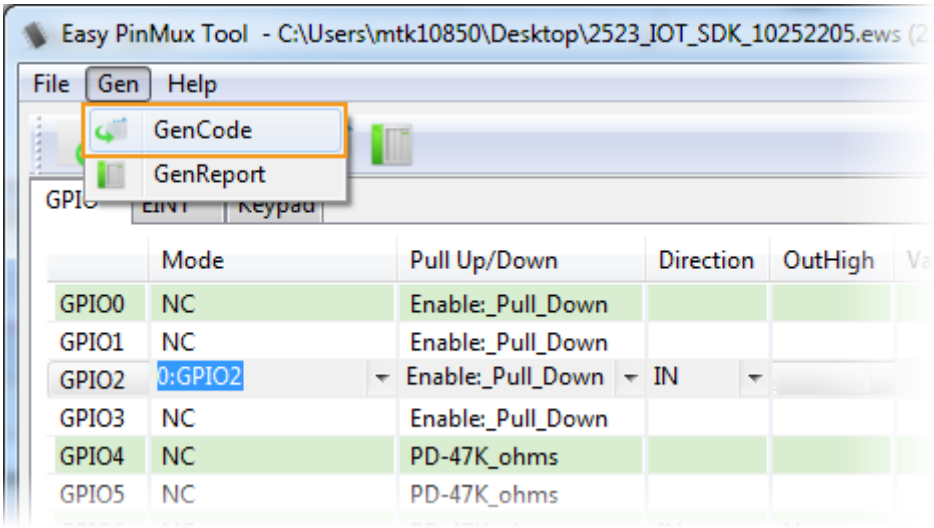

Figure 5. Generate code for the current configuration

A confirmation message then appears, as shown in Figure 6.

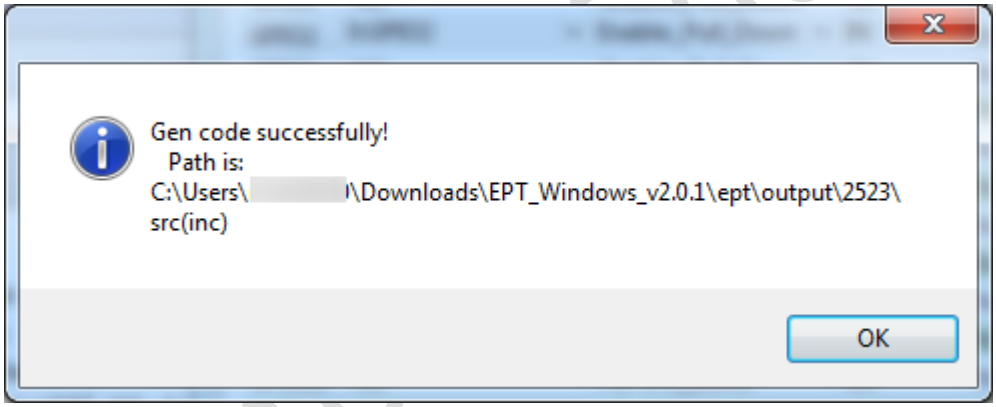

Figure 6. Code is successfully generated

2) Copy the generated source code to destination driver folder of the project. The full paths of the driver folder for source and header files are as follows.

```
<sdk_root>\project\<board>\apps\<application>\src
<sdk_root>\project\<board>\apps\<application>\inc
```

<board> is the name of your board, such as mt7933_hdk and <application> is the name of your project, such as iot_sdk_dev, iot_sdk.

The GPIO settings configured by the EPT take effect only when they are written to GPIO registers. Details about this can be found in the readme file of EPT example code located under the folder:

```
<sdk_root>\project\<board>\apps\<application>\
```

Once the configuration is set, build the load on the target device. More information about building the load can be found in MediaTek IoT Development Platform for RTOS Get Started Guide.

An example use case to generate the files based on given inputs is shown in Figure 7. In this use case, the user provides .ews, .chip and .conf files as an input to the EPT and the expected outcomes are .ews, .h and .c files.

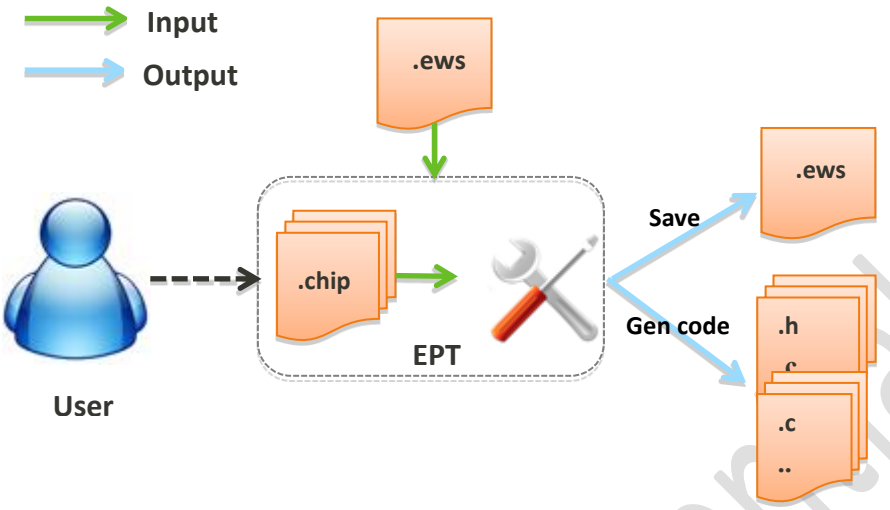

Figure 7. An example to generate configuration and executable files with given input files

1.3 Folder Structure of the EPT

The EPT contains five main folders and two files. The folder structure is shown in Figure 8.

1) configuration — This folder stores the configuration files for MediaTek chipsets.
2) output — This folder contains output files such as source (.c, saved in the src folder) and header (.h, saved in the inc folder) files and the pinout report (pinout_report.csv) generated by the EPT.
3) project — This folder contains the workspace files (.ews) saved in this folder by default.
4) EPT — This folder contains libraries to run the EPT.
5) jre1.8_win — This folder includes the supporting files to run EPT on Windows OS.
6) generate_script.bat — Provides a command line mode to generate code.

The ept folder also contains the executable file (ept.exe) to start the configuration. Launch the program by double-clicking the executable file. There is no need to install it.

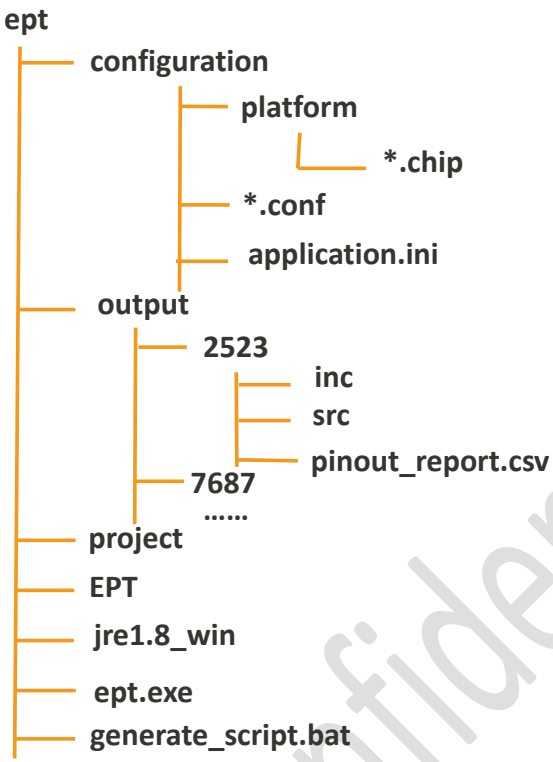

Figure 8. Folder structure

1.3.1 Chip Files

Chip files store the chip parameters and options shown on the EPT UI. Every chip has its own chip file. You must not modify the content of the chip file.

1.3.2 Configuration Files

The chipsets share the same configuration (.conf) files. Usually there are multiple *.conf files for different modules, such as gpio.conf, eint.conf and keypad.conf. An example .conf file is shown in Figure 9.

1) gpio.conf — contains the header and tail information of the .h file for GPIOs.
2) keypad.conf — contains the key symbols of the keyboard.
3) eint.conf — contains the header and tail information of the .h file for EINTs.

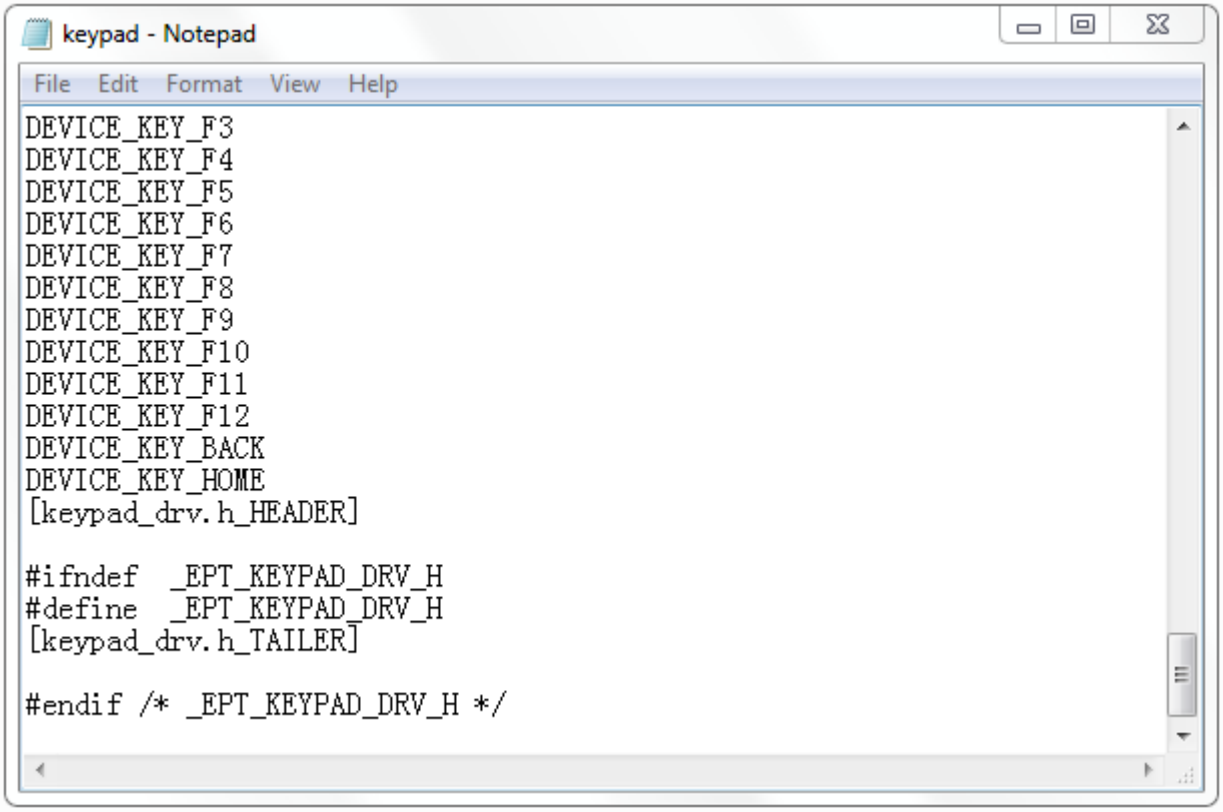

Figure 9. Keypad .conf file

1.3.3 EWS Files

EWS (.dws for versions before 2.0.1) files are the EPT workspace files. The file contains customized PinMux and I/O settings. See Section 1.2.1, “Workspace File Options” to create or modify a workspace file. MediaTek also provides demo workspace files where many PinMux settings are already configured. These demo .dws or .ews files are stored under the /project folder. We recommend that you modify and save an existing demo workspace file to create custom design configurations.

1.3.4 Pinout Report

To generate a pinout report (pinout_report.csv), click **GenReport** on the **Gen** menu. The report contains the PinMux information set by the user, including seven columns for **Name**, **Mode**, **Pull Up & Pull Down**, **Direction**, **OutHigh**, **VarName** and **User Information**.

2 Driver Settings

This chapter describes the driver settings for General Purpose Input and Output (GPIO).

2.1 GPIO

Open the **GPIO** page by opening or creating a workspace on the main UI, as shown in Figure 2. This page is used to set the GPIO parameters. The GUI enables setting the **Mode**, **Pull Up/Down**, **Direction**, **OutHigh**, variable name **VarName** and **User Information** for the GPIO pins.

2.1.1 Setting the Mode Option

Figure 10 shows the relationship between the GPIO **Mode** selected by the user and the code generated in the ept_gpio_drv.h header file. The **GPIO0** pin has 10 modes. The following example shows how to select **EINT0** corresponding to **Mode 1** for **GPIO0**.

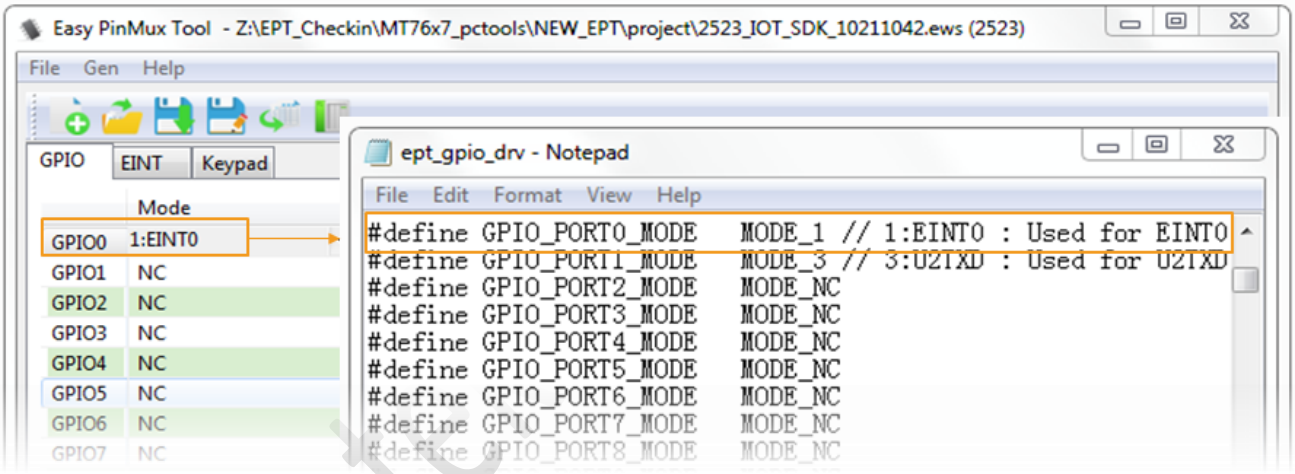

Figure 10. GPIO0 mode and generated code

2.1.2 Setting the Pull Up/Down, Direction and OutHigh Options

The **Pull Up/Down** option is available when the **Mode** is selected. If the selected **Mode** is **GPIO**, **Direction** is available too. See the options available for **GPIO10** in Figure 11. Additionally, the **OutHigh** checkbox is invisible until the GPIO pin **Direction** is set to **OUT**. Figure 11 shows the **GPIO10** available options and generated code for **Pull Up/Down**. **Direction** and **OutHigh** values are again stored in the ept_gpio_drv.h header file.

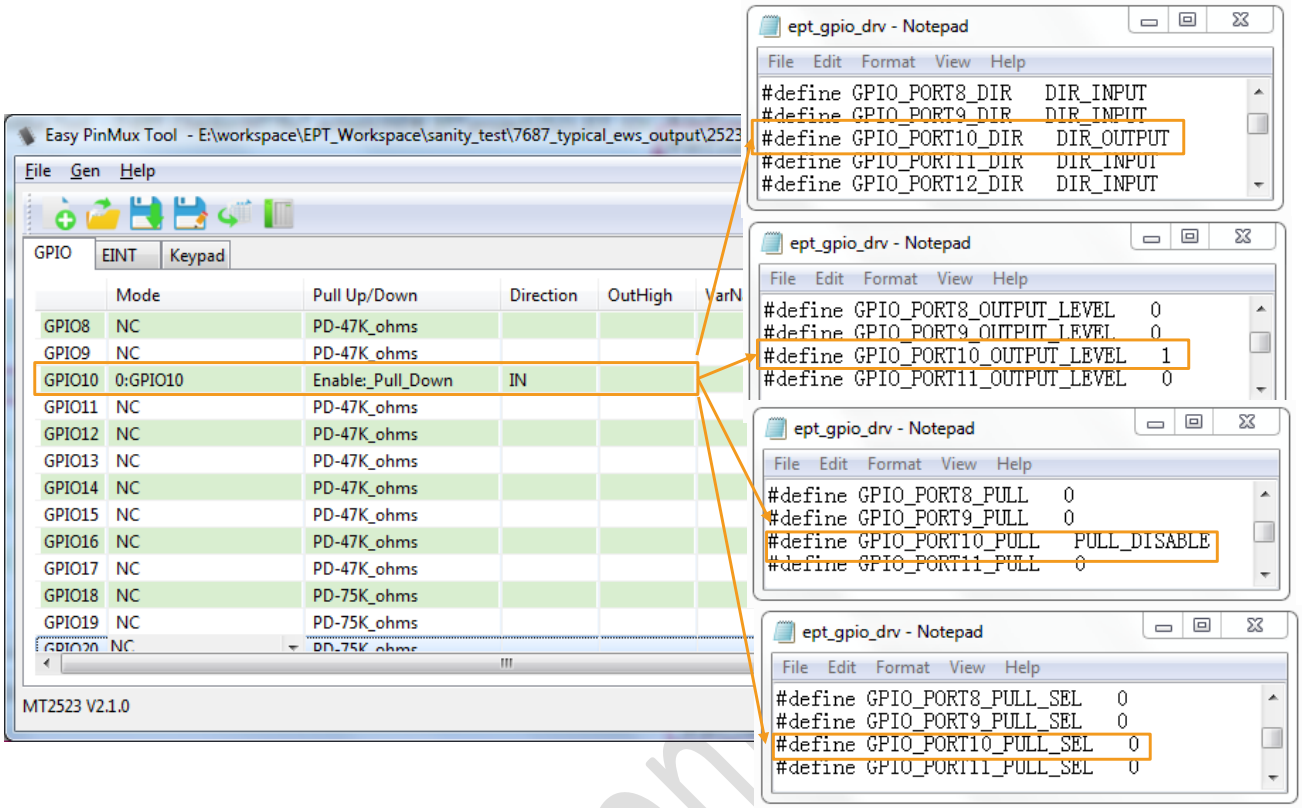

Figure 11. GPIO10 options and generated code in the `ept_gpio_drv.h` header file

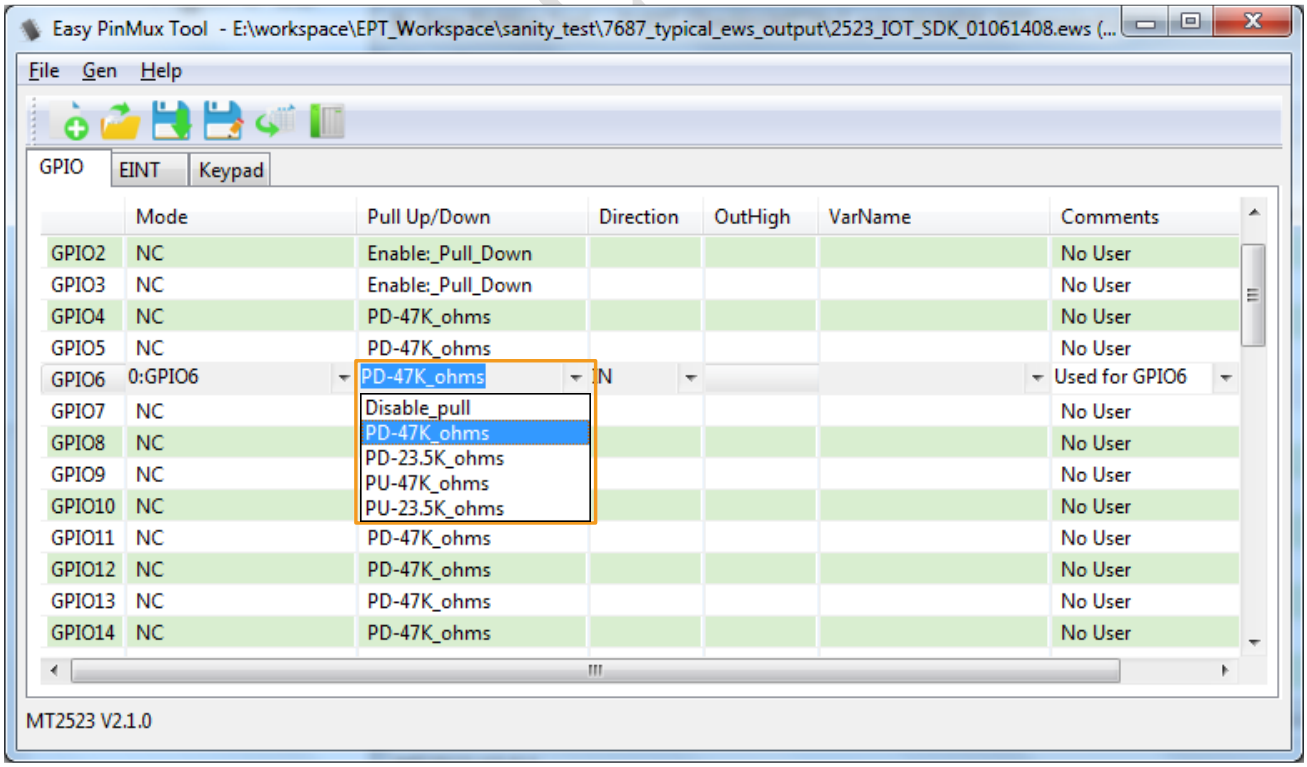

Figure 12. Pull Up/Down options for GPIO6

The option list shown in Figure 12 includes five items. The mapping relationship between the item selected for **Pull Up/Down** and generated code is shown in Figure 13.

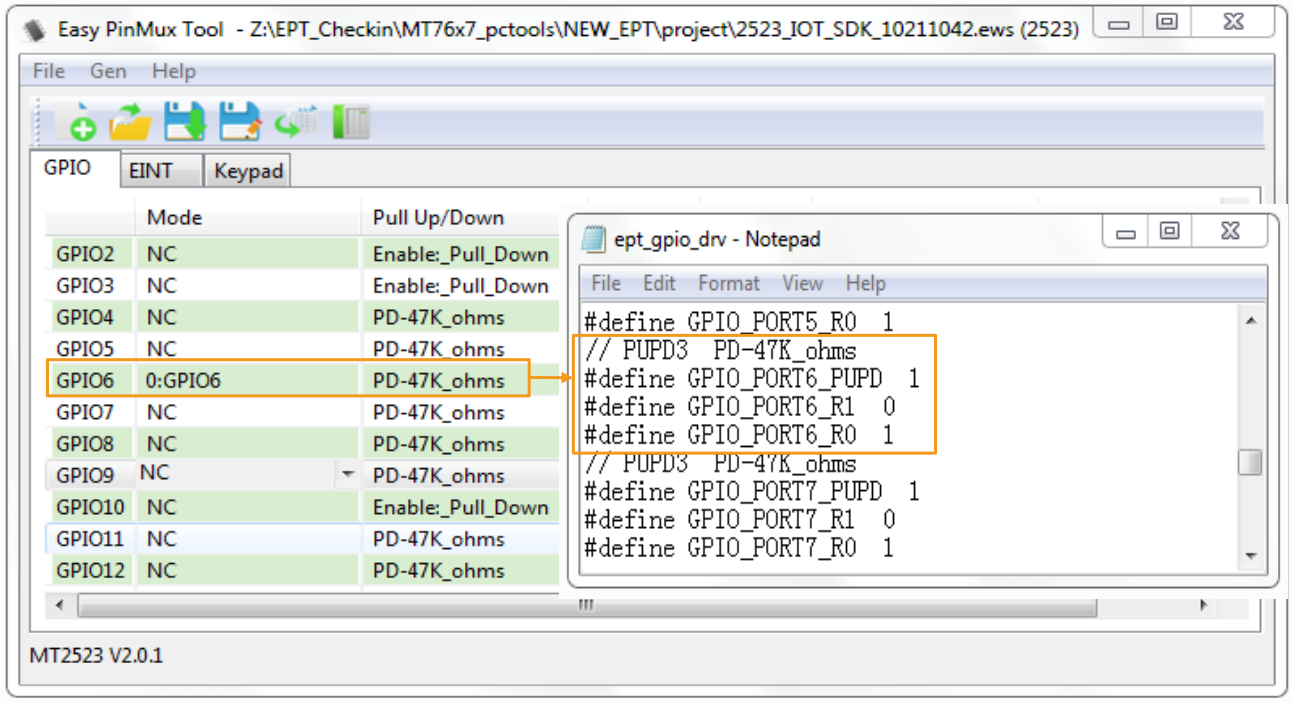

Figure 13. GPIO6 Pull Up/Down state

2.1.3 Setting the VarName

You can also select a variable name for the given GPIO pin. The selected variable name should be unique. There is a list of options to choose from **VarName** and the default value is set and stored in the ept_gpio_var.h file, as shown in Figure 14.

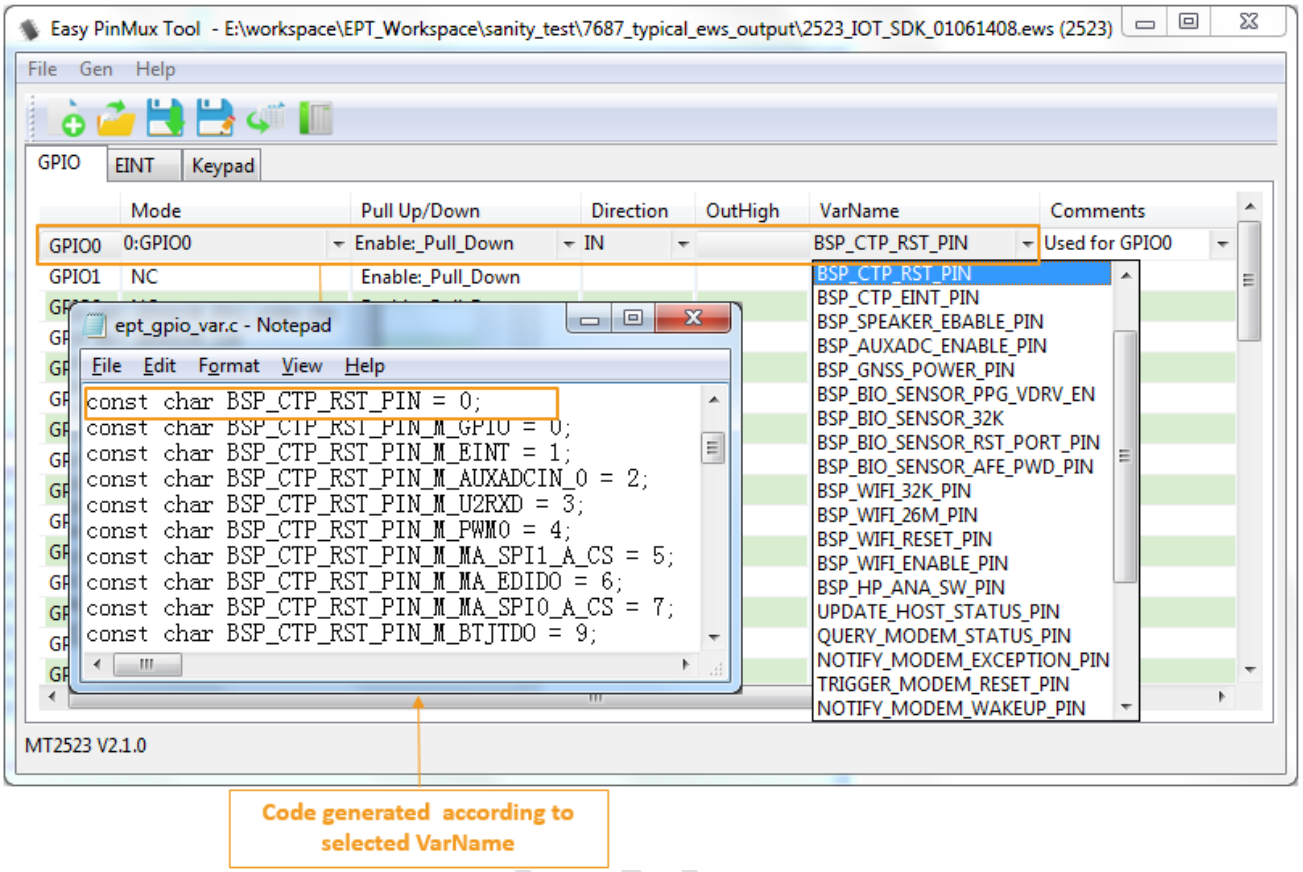

Figure 14. Variable name set for the GPIO0

The final configurations of all pins with their corresponding modes are stored in a file, as shown in Figure 15.

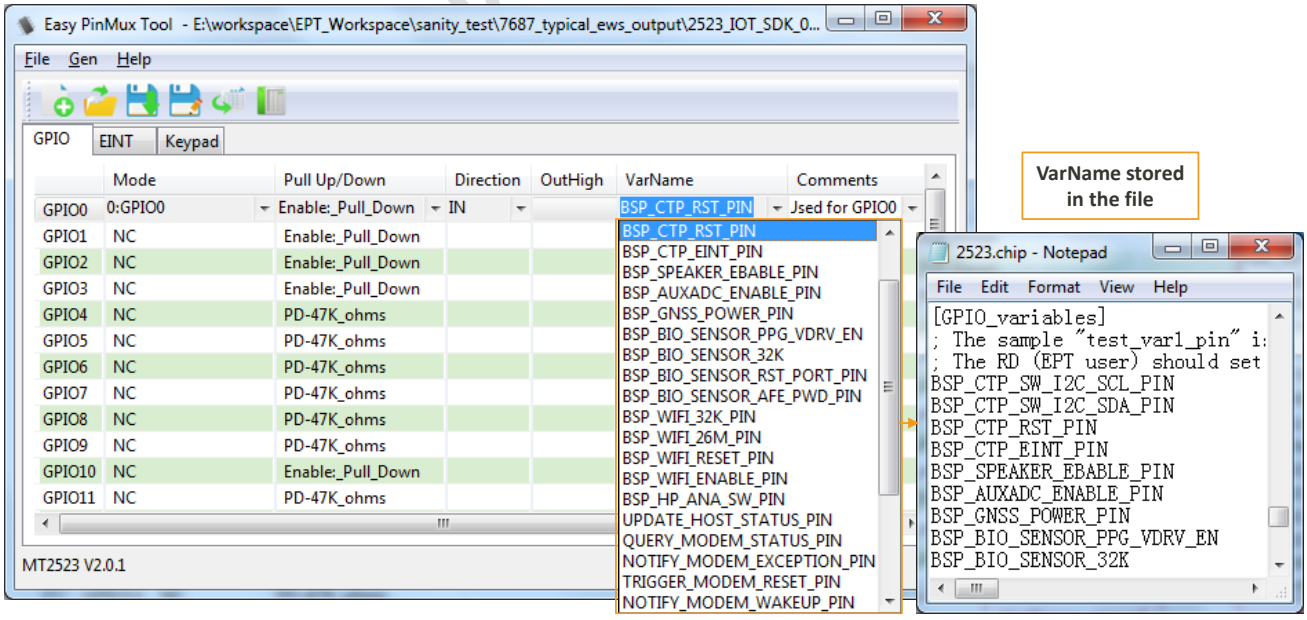

Figure 15. VarName column stored in file

2.1.4 Setting the Comments

The **Comments** defines the use of its corresponding pin, as shown in Figure 16.

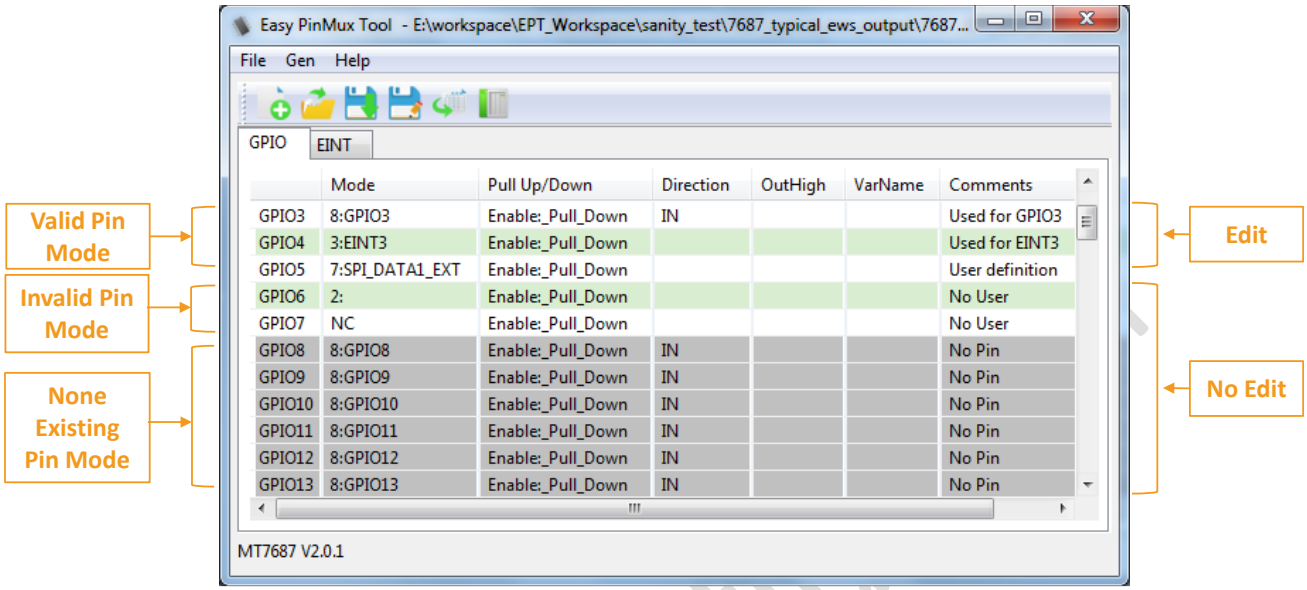

Figure 16. User Information for corresponding GPIO mode

By default, there are three options to choose from:

1) **Used for "pin name"** — assigned to a valid pin mode selected. You can modify the default option. However, we do not recommend changing the default setting unless it's necessary.
2) **No User** — assigned to an invalid pin mode and cannot be edited.
3) **No Pin** — assigned to the pin mode that's not available for the current chip and cannot be edited.

Exhibit 1 Terms and Conditions

Your access to and use of this document and the information contained herein (collectively this "Document") is subject to your (including the corporation or other legal entity you represent, collectively "You") acceptance of the terms and conditions set forth below ("T&C"). By using, accessing or downloading this Document, You are accepting the T&C and agree to be bound by the T&C. If You don't agree to the T&C, You may not use this Document and shall immediately destroy any copy thereof.

This Document contains information that is confidential and proprietary to MediaTek Inc. and/or its affiliates (collectively "MediaTek") or its licensors and is provided solely for Your internal use with MediaTek's chipset(s) described in this Document and shall not be used for any other purposes (including but not limited to identifying or providing evidence to support any potential patent infringement claim against MediaTek or any of MediaTek's suppliers and/or direct or indirect customers). Unauthorized use or disclosure of the information contained herein is prohibited. You agree to indemnify MediaTek for any loss or damages suffered by MediaTek for Your unauthorized use or disclosure of this Document, in whole or in part.

MediaTek and its licensors retain titles and all ownership rights in and to this Document and no license (express or implied, by estoppels or otherwise) to any intellectual propriety rights is granted hereunder. This Document is subject to change without further notification. MediaTek does not assume any responsibility arising out of or in connection with any use of, or reliance on, this Document, and specifically disclaims any and all liability, including, without limitation, consequential or incidental damages.

THIS DOCUMENT AND ANY OTHER MATERIALS OR TECHNICAL SUPPORT PROVIDED BY MEDIATEK IN CONNECTION WITH THIS DOCUMENT, IF ANY, ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. MEDIATEK SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, COMPLETENESS OR ACCURACY AND ALL WARRANTIES ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. MEDIATEK SHALL NOT BE RESPONSIBLE FOR ANY MEDIATEK DELIVERABLES MADE TO MEET YOUR SPECIFICATIONS OR TO CONFORM TO A PARTICULAR STANDARD OR OPEN FORUM.

Without limiting the generality of the foregoing, MediaTek makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does MediaTek assume any liability arising out of the application or use of any product, circuit or software. You agree that You are solely responsible for the designing, validating and testing Your product incorporating MediaTek's product and ensure such product meets applicable standards and any safety, security or other requirements.

The above T&C and all acts in connection with the T&C or this Document shall be governed, construed and interpreted in accordance with the laws of Taiwan, without giving effect to the principles of conflicts of law.