

## Holtek HT32 Series

How to upgrade Firmware Library from the past 165x  
Firmware Library to HT32\_STD\_1xxxx\_FWLib

如何由旧有 165x Firmware Library 升级到  
HT32\_STD\_1xxxx\_FWLib

Revision History

Version	Date	Descriptions
0.01	2017-06-15	✓ Initial version
0.02	2017-06-17	✓ Fix typing error and descriptions

In order to reduce the overhead of moving the application firmware code between different HT32 MCU devices, we merge the 165x and 1xxxx firmware library into a HT32 standard Firmware Library (which named “HT32\_STD\_1xxxx\_FwLib\_vnnn\_mmm”). The HT32\_STD\_1xxxx\_FwLib now supports the following HT32 MCU. We will try to support the new device in the same HT32\_STD\_1xxxx\_FwLib as far as possible in the future.

This document lists the different between past 165x Firmware Library and new HT32\_STD\_1xxxx\_FwLib. User can refer to the document to update the code and move to the new Firmware Library by simply rename the file and modify specify keyword such as functions name and definitions.

为了减少在不同的 HT32 MCU 设备之间移动应用程序固件代码花费的时间，我们将 165x 和 1xxxx 固件库合并到 HT32 标准固件库（命名为 “HT32\_STD\_1xxxx\_FwLib\_vnnn\_mmm”），HT32\_STD\_1xxxx\_FwLib 现在支持以下 HT32 MCU，未来我们将尽量在同一 HT32\_STD\_1xxxx\_FwLib 中支持新设备。

本文件列出了 165x 固件库与新 HT32\_STD\_1xxxx\_FwLib 固件库之间的不同之处。用户可以参考此文件来更新代码并移动到新的固件库，只需重命名文件和修改指定关键词，如函数名称和定义。

#### Supported HT32 Device / 支援的 HT32 Device:

HT32F1653, HT32F1654, HT32F1655, HT32F1656

HT32F12345, HT32F12365, HT32F12366, HT32F22366

#### Note:

New MCU device may not be updated in the above supported list. Check the list in the “Release\_Notes.txt” for the latest supported HT32 MCU of the HT32\_STD\_1xxxx\_FwLib. “Release\_Notes.txt” file can be found in the root path of Firmware Library.

新的 MCU 可能不会更新到上面的清单。请确认 “Release\_Notes.txt” 来得知 HT32\_STD\_1xxxx\_FwLib 最新支持的 HT32 MCU。“Release\_Notes.txt” 档案可以在 Firmware Library 的根目录下找到。

## Merge Steps

Please refer to the following merge steps. It shall reduce the try and error time.

<b>Step 1</b>	Unzip the “HT32_STD_1xxxx_FWLib_vnnn_mmm.zip” into a folder such as “d:\HT32_STD_1xxxx_FWLib_v001_167”.
<b>Step 2</b>	Copy the “project_template” into the same folder and rename it as your own name. Such as: Source: \\project_template\IP\Example\ Destination: \\project_user\app1\code\ Note: the subfolder “\\project_user\app1\Template” and “\\project_user\app1\Template_USB” can be deleted.
<b>Step 3</b>	Refer to the “File modification” list in the next section to merge the code from your old project source code into the new one. The “ht32xxxx_it.c” shall be pay special attention, others can be use directly with simple setting checking.
<b>Step 4</b>	Copy the other source code files (which are not in the “File modification” list) into the new path, “\\project_user\app1\code\” according to the original folder structure.
<b>Step 5</b>	Open the project files such as “...MDK_ARMv5\Project_1656.uvprojx” or “...EWARM\Project_1656.eww”. Add the necessary source code of the Step 4 into the project build list.
<b>Step 6</b>	Build the project and confirm the error or warning result with “API & definitions modification” section. For example, if the compiler report the following errors, find the keyword “USART_IER_THREIE” in the “API & definitions modification” section and fix it in your source code. Repeat the above process until all the error and warning are fixed.  <pre> compiling test.c... ..\test.c(577): error: #20: identifier "USART_IER_THREIE" is undefined     USART_IntConfig(     IFI_UART_PORT, USART_IER_THREIE, ENABLE); ..\test.c(1103): error: #20: identifier "USART_IER_THREIE" is undefined     USART_IntConfig(     IFI_UART_PORT, USART_IER_THREIE, DISABLE); ..\test.c(1124): error: #20: identifier "USART_LSR_RFDR" is undefined </pre>

## File modification:

Old	New Copy it from the path, "project_template\IP\Example".	Merge Type
ht32f165x_it.c	ht32f1xxxx_01_it.c	A
ht32f165x_usbdconf.h	ht32f1xxxx_01_usbdconf.h	B (Check USB setting)
system_ht32f165x.c	system_ht32f1xxxx_01.c	B (Check clock related setting)
ht32f1653_54_conf.h	ht32f1xxxx_conf.h	B
MDK_ARM/startup_ht32f165x.s	MDK_ARM/startup_ht32f1xxxx_01.s	B (Check stack/heap size)
MDK_ARM/HT32F12365_66_DebugSupport.ini	MDK_ARM/HT32F1xxxx_01_DebugSupport.ini	B (Check stack/heap size)
ERARM/startup_ht32f165x.s	ERARM/ startup_ht32f1xxxx_01.s	B (Check stack/heap size)

### Merge type:

**Type A:** Please compare and move your own code into the new one.

**Type B:** The new file can be use directly with setting checking.

## API & definitions modification:

IP	Modification	Note
ht32_serial	<b>Old:</b> N/A <b>New:</b> void SERIAL_Flush(void);	New function for re-target interrupt mode.
ADC	<b>Old:</b> N/A <b>New:</b> void ADC_Cmd(HT_ADC_TypeDef* HT_ADCn, ControlStatus NewState);	New dummy function for API consistency and compatibility.
CKCU	<b>Old:</b> N/A <b>New:</b> bool CKCU_HSIAutoTrimIsReady(void);	
	<b>Old:</b> CKCU_PeripClockConfig_TypeDef SCI <b>New:</b> CKCU_PeripClockConfig_TypeDef <b>SCIO</b>	
	<b>Old:</b> CKCU_PeripPrescaler_TypeDef CKCU_PCLK_SCI <b>New:</b> CKCU_PeripPrescaler_TypeDef <b>CKCU_PCLK_SCIO</b>	
	<b>Old:</b> CKCU_DBG_SCI_HALT <b>New:</b> CKCU_DBG_ <b>SCIO</b> _HALT	
CMP	<b>Old:</b> void CMP_DeInit(void); <b>New:</b> void CMP_DeInit( <b>HT_CMP_TypeDef* HT_CMPn</b> );	
CRC	<b>Old:</b> void CRC_DeInit(void); <b>New:</b> void CRC_DeInit( <b>HT_CRC_TypeDef* CRC</b> );	

IP	Modification	Note
GPIO	<p><b>Old:</b></p> <pre>void AFIO_GPAConfig(u32 AFIO_PIN_n, AFIO_MODE_Enum AFIO_MODE_n); void AFIO_GPBConfig(u32 AFIO_PIN_n, AFIO_MODE_Enum AFIO_MODE_n); void AFIO_GPCConfig(u32 AFIO_PIN_n, AFIO_MODE_Enum AFIO_MODE_n); void AFIO_GPDConfig(u32 AFIO_PIN_n, AFIO_MODE_Enum AFIO_MODE_n); void AFIO_GPEConfig(u32 AFIO_PIN_n, AFIO_MODE_Enum AFIO_MODE_n);</pre> <p><b>New:</b></p> <pre>void <b>AFIO_GPxConfig(u32 GPIO_Px, u32 AFIO_PIN_n, AFIO_MODE_Enum AFIO_MODE_n);</b></pre>	Where GPIO_Px is GPIO_PA ~ GPIO_PE.
GPTM	<p>All the "GPTM*" keywords are replace by "TM*", for example:</p> <pre>GPTM_TimeBaseInit -&gt; TM_TimeBaseInit HT_GPTM_TypeDef -&gt; HT_TM_TypeDef</pre> <p>expect the following items:</p> <pre>ht32f1xxxx_01.h HT_GPTM0, HT_GPTM1, GPTM0_IRQn, GPTM1_IRQn</pre>	The same with M0+ Firmware Library.

IP	Modification	Note
MCTM	<p>All the "MCTM*" keywords are replace by "TM*", for example:  MCTM_TimeBaseInit -&gt; TM_TimeBaseInit  HT_MCTM_TypeDef -&gt; HT_TM_TypeDef  expect the following items:  a. All define in the ht32f1xxxx_mctm.h  b. ht32f1xxxx_tm_type.h  MCTM_OIS_LOW, MCTM_OIS_HIGH  MCTM_COMUS_STIOFF, MCTM_COMUS_STION  c.ht32f1xxxx_01.h  HT_MCTM0, HT_MCTM1,  MCTM0BRK_IRQn, MCTM0UP_IRQn, MCTM0TR_IRQn, MCTM0CC_IRQn  MCTM1BRK_IRQn, MCTM1UP_IRQn, MCTM1TR_IRQn, MCTM1CC_IRQn</p>	The same with M0+ Firmware Library.
PDMA	<p><b>Old:</b> PDMA_SCI_TX, PDMA_SCI_RX  <b>New:</b> PDMA_<b>SCIO</b>_TX, PDMA_<b>SCIO</b>_RX</p>	
PWRCU	<p><b>Old:</b> void PWRCU_V18RDYSourceConfig(PWRCU_V18RDYSC_Enum Sel);  <b>New:</b> void <b>PWRCU_VRDYSourceConfig(PWRCU_VRDYSC_Enum Sel);</b></p>	
	<p><b>Old:</b> PWRCU_V18RDYSC_BKISO  <b>New:</b> PWRCU_<b>VRDYSC</b>_BKISO</p>	
	<p><b>Old:</b> PWRCU_V18RDYSC_V18POR  <b>New:</b> PWRCU_<b>VRDYSC</b>_VPOR</p>	
RSTCU	<p><b>Old:</b> RSTCU_PeripReset_TypeDef SCI  <b>New:</b> RSTCU_PeripReset_TypeDef <b>SCIO</b></p>	



IP	Modification		Note
RTC	<b>Old:</b> void RTC_LSI_LoadTrimData(void); <b>New:</b> <b>Removed (unnecessary)</b>		
USART UART	<b>Old:</b> N/A <b>New:</b> <b>u32 USART_GetLineStatusValue(HT_USART_TypeDef* USARTx);</b>		New function for the read clear bit in LSR.
	<b>Old:</b> void USART_TXTLConfig(HT_USART_TypeDef* USARTx, u32 USART_tl); <b>New:</b> void <b>USART_TXRXTLConfig</b> (HT_USART_TypeDef* USARTx, <b>u32 USART_CMD_x</b> , u32 USART_tl); <b>//void USART_RXTLConfig(HT_USART_TypeDef* USARTx, u32 USART_RFITL);</b> <b>//void USART_TXTLConfig(HT_USART_TypeDef* USARTx, u32 USART_TFITL);</b>		Where USART_CMD_x is USART_CMD_TX or USART_CMD_RX
	<b>Old</b>	<b>New</b>	
	USART_RFITL_01 USART_RFITL_04 USART_RFITL_08 USART_RFITL_14	USART_ <b>RXTL</b> _01 USART_ <b>RXTL</b> _04 USART_ <b>RXTL</b> _08 USART_ <b>RXTL</b> _14	
	USART_TFITL_00 USART_TFITL_02 USART_TFITL_04 USART_TFITL_08	USART_ <b>TXTL</b> _00 USART_ <b>TXTL</b> _02 USART_ <b>TXTL</b> _04 USART_ <b>TXTL</b> _08	
	USART_IER_RDAIE USART_IER_THREIE USART_IER_RLSIE USART_IER_MSIE	USART_ <b>INT_RXDR</b> USART_ <b>INT_TXDE</b> USART_ <b>INT_RLSIE</b> USART_ <b>INT_MSIE</b>	
	USART_LSR_RFDR USART_LSR_THRE USART_LSR_TE USART_LSR_ERR	USART_ <b>FLAG_RXDNE</b> USART_ <b>FLAG_THRE</b> USART_ <b>FLAG_TXC</b> USART_ <b>FLAG_ERR</b>	

IP	Modification		Note
	USART_LSR_OEI USART_LSR_PEI USART_LSR_FEI USART_LSR_BII USART_LSR_RSADDEF	USART_LSR_OE USART_LSR_PE USART_LSR_FE USART_LSR_BI USART_LSR_RSADD	
	<p><b>Note:</b> The LSR flags (USART_LSR_OEI ~ USART_LSR_RSADD) is not suitable for the “USART_GetLineStatus function (rename as USART_GetFlagStatus)” since there are read clear bit in LSR register of HT32F165x. We add new function, USART_GetLineStatusValue to return the LSR directly. User can use those flag with the new function to confirm which bits of LSR are set.</p> <p><b>Example Code:</b></p> <pre> u32 uLSRValue; uLSRValue = USART_GetLineStatusValue(HT_USART0); if (uLSRValue &amp; USART_LSR_OE) {     //OE process } if (uLSRValue &amp; USART_LSR_PE) {     //OE process }           </pre>		