



HT32_CodeConfig 使用手册

版本：V1.00 日期：2023-02-20

www.holtek.com

目录

1. 简介与安装	3
1.1 简介	3
1.2 安装	3
2. 快速上手	3
3. Clock 页面的设置	8
4. MISC 页面的设置	9
5. AFIO 页面的设置	10
6. GPIO 页面的设置	11
7. EXTI 页面的设置	12
8. Timer 页面的设置	13
9. ADC/DAC 页面的设置	16
9.1 ADC 的说明	16
9.2 DAC 的说明	17
10. CMP 页面的设置	18
11. SPI 页面的设置	19
12. UART/USART 页面的设置	20
13. I²C 页面的设置	21
14. SCI 页面的设置	22
15. EBI 页面的设置	23
15.1 EBI 基础设置	23
15.2 驱动 TFT 屏	24
15.3 LCD 平台	25
16. I²S 页面的设置	28
17. SDIO 页面的设置	29
18. CSIF 页面的设置	29
19. MonoLCD 页面的设置	30
20. SLED 页面的设置	31
21. LED 页面的设置	32
22. Other 页面的设置	33
23. Tools 页面的设置	34
23.1 UART IAP 设置	34
23.2 USB IAP 设置	36
23.3 I ² C Emulation	45
23.4 EEPROM Emulation	46
24. Help 页面	47

1. 简介与安装

1.1 简介

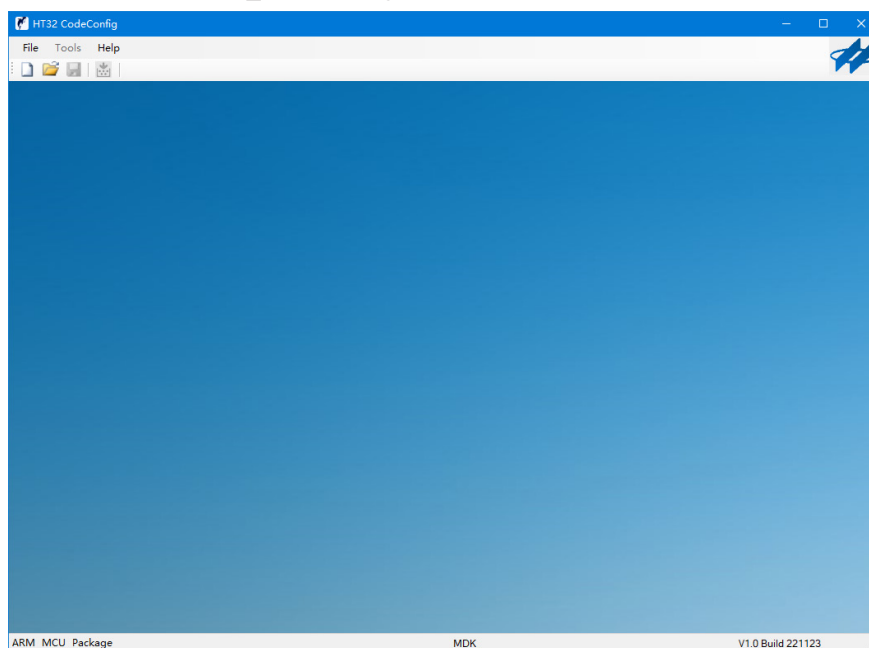
HT32_CodeConfig 是一个 HT32 系列 MCU 的辅助开发软件，可以协助建立不同型号 MCU 的 MDK/IAR 工程，并可以按图形化设定选项，完成 MCU 基础外设的初始化。除此之外还附加了 USB/UART IAP 功能、LCD 开发平台等功能。

1.2 安装

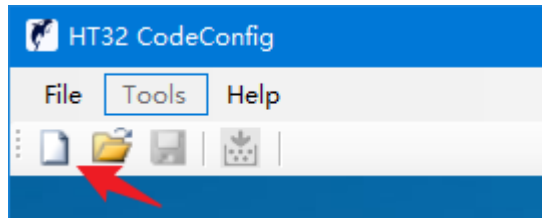
- 安装于 Microsoft WIN7/WIN8/WIN10/WIN11 桌面操作系统
- 系统需要安装 .NET Framework 4 或者更新版本。一般 WIN10/WIN11 都已经随系统安装，若没有安装，请于 Microsoft 官网下载安装：<https://dotnet.microsoft.com/download>
- 具体安装过程里，按提示一直点“下一步”就可以完成安装

2. 快速上手

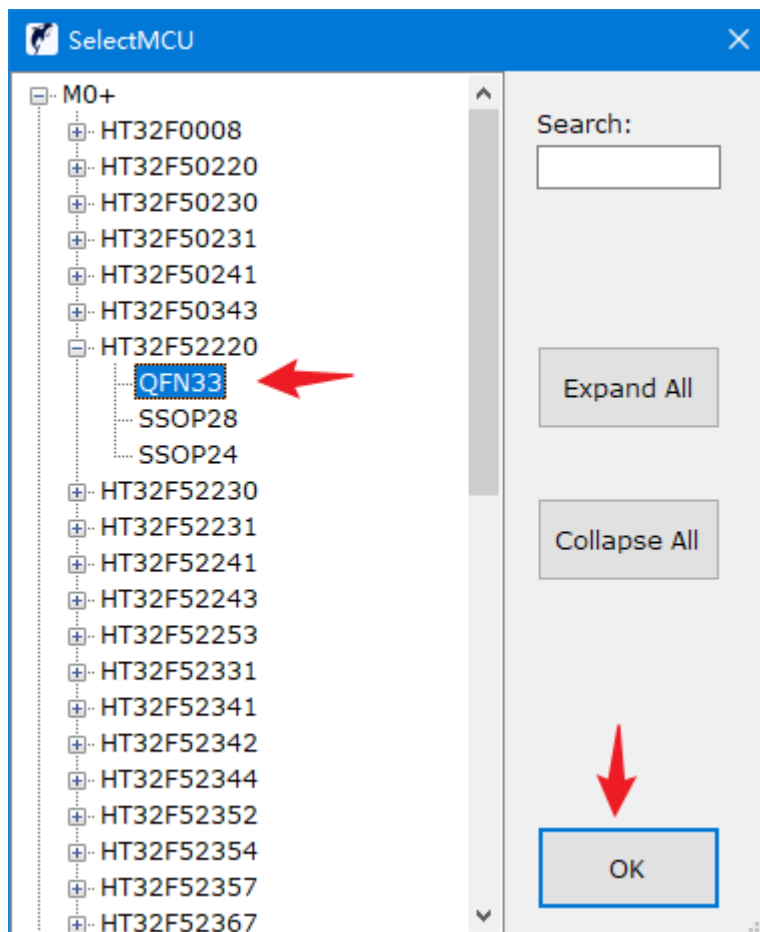
步骤 1. 打开安装好的 HT32_CodeConfig.exe，初始界面如下图。



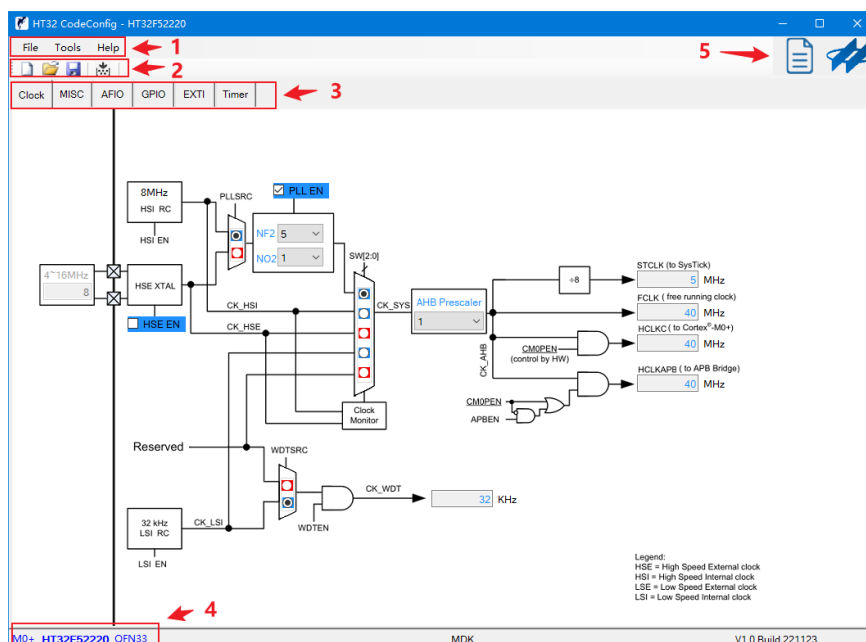
步骤 2. 选择菜单 File→New，或者点击图标 New，或者鼠标点击主窗口任意位置 (中间蓝色部分)。



步骤 3. 在弹出的型号 / 封装选择窗口里，选择所需的型号和封装，例如下图的 HT32F52220→QFN33，点 OK。也可以在搜索框内输入关键词进行筛选后再选择型号。



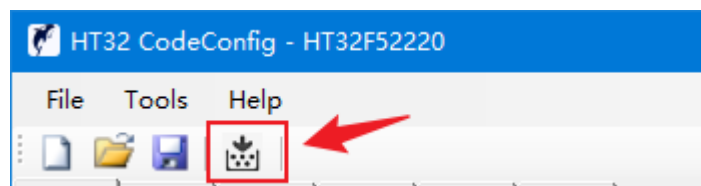
步骤 4. 主窗口会变成下图所示。



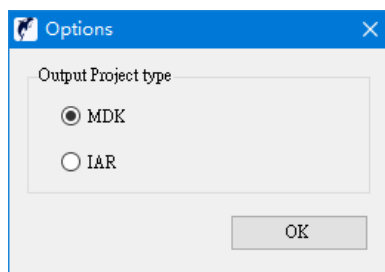
1. 菜单栏；2. 工具栏；3. 功能标签；
4. 内核 / 型号 / 封装；5. 在线文档链接

步骤 5. 选择菜单 Tools→Generate Code，或者点击图标 Generate Code。

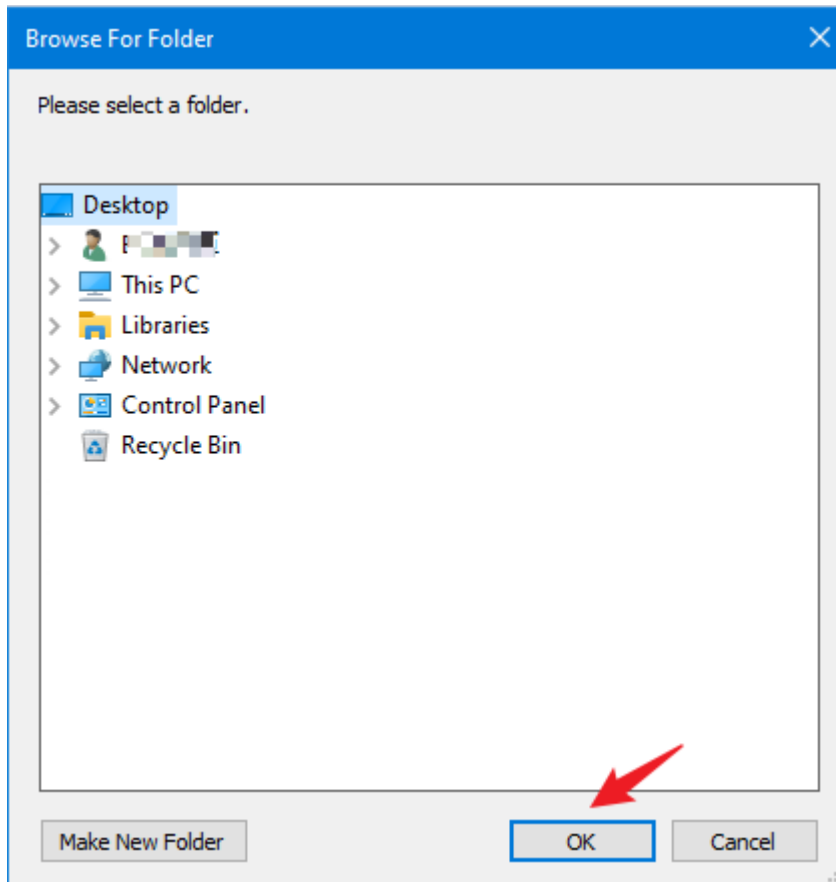
保存 (Save) 功能是为了保存所有配置项，并不输出 MDK/IAR 工程。
保存好的配置项可以通过打开 (Open) 功能重新打开。



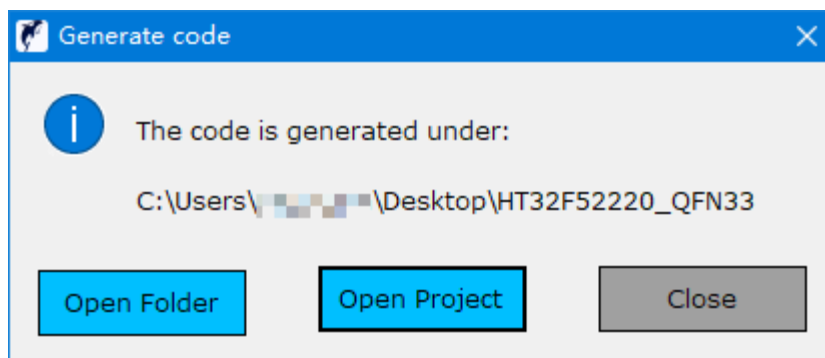
输出的工程格式可以通过选择菜单 Tools→Options 进入设置页面。目前只支持 MDK/IAR 格式。下面是以 MDK 为例进行说明。

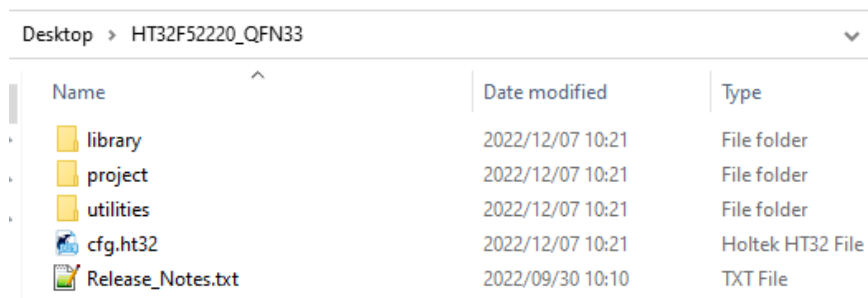


步骤 6. 在弹出的浏览文件夹窗口，选择路径保存，会按选择的 MCU 名称和封装命名保存，若有同文件夹会自动增加数字后缀。



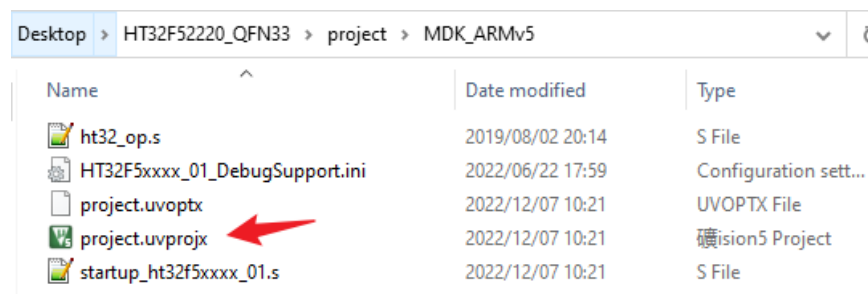
步骤 7. 工程创建好会弹出提示打开对应文件夹或者工程的对话框，按提示打开。





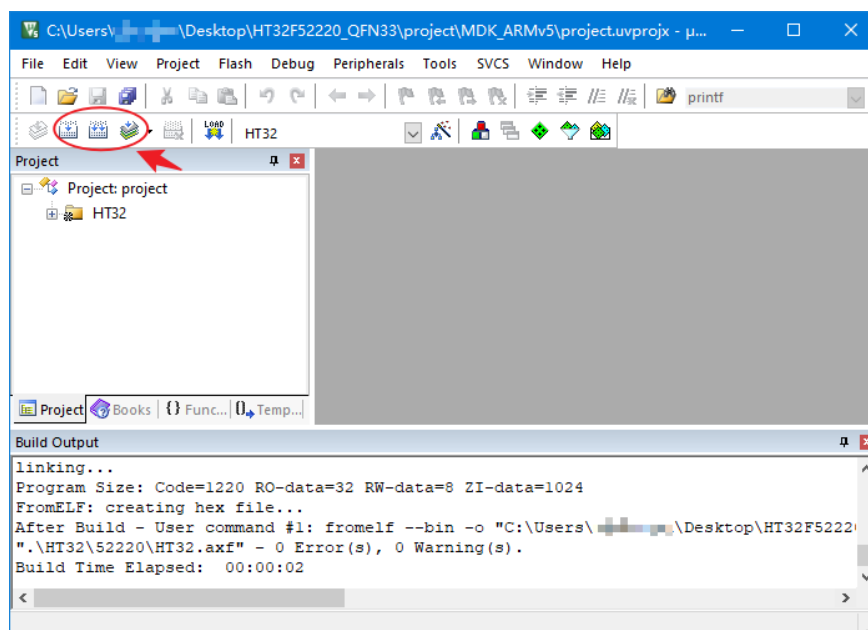
Name	Date modified	Type
library	2022/12/07 10:21	File folder
project	2022/12/07 10:21	File folder
utilities	2022/12/07 10:21	File folder
cfg.ht32	2022/12/07 10:21	Holtek HT32 File
Release_Notes.txt	2022/09/30 10:10	TXT File

步骤 8. 进入 project\MDK_ARMv5 文件夹，如果已经安装了 MDK，双击 project.uprojx 就可以启动 MDK 打开工程了。



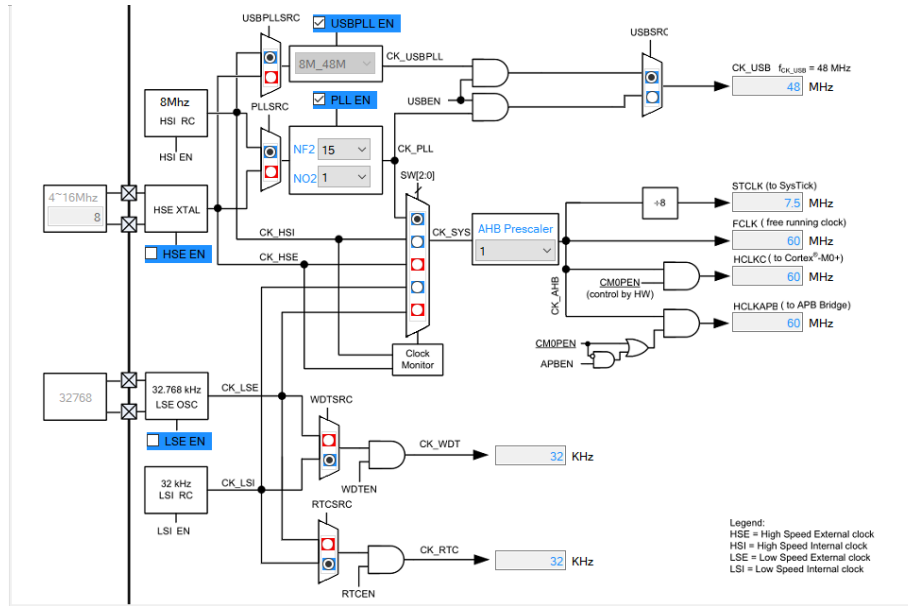
Name	Date modified	Type
ht32_op.s	2019/08/02 20:14	S File
HT32F5xxxx_01_DebugSupport.ini	2022/06/22 17:59	Configuration sett...
project.uvoptx	2022/12/07 10:21	UVOPTX File
project.uprojx	2022/12/07 10:21	磳ision5 Project
startup_ht32f5xxxx_01.s	2022/12/07 10:21	S File

步骤 9. 点击 MDK 的 Build 编译工程。编译完成后，若对应硬件已经通过 e-Link32 Lite/Pro 连接，就能点击 Download 下载到 MCU 内了。



3. Clock 页面的设置

Clock 页面用于设置 HSE、LSE、HSI、LSI、PLL、Prescaler 等，如下图。蓝色位置为可以点选的部分，点选不同的单选框能在最右侧看到不同时钟的不同值。注意：不同型号 MCU 的 Clock 设置页面可能不相同。



- HSE EN: 使能外部高速晶振，使能后 PLLSRC/USBPLLSRC/CLKSRC 将会有对应的单选框变成可选状态 (蓝色)，并且 HSE 的值可在指定范围内修改。反之关闭 HSE 之后对应的单选框变成红色。
- LSE EN: 使能外部低速晶振，使能后 WDTSRC/RTCSRC/CLKSRC 将会有对应的单选框变成可选状态 (蓝色)。反之关闭 LSE 之后对应的单选框变成红色。
- PLL EN: 使能锁相环，使能后 USBSRC/CLKSRC 将会有对应的单选框变成可选状态 (蓝色)。反之关闭 PLL 之后对应的单选框变成红色。PLL 使能后 NF2、NO2 的值可以修改以便输出不同的时钟值。不是所有型号的 MCU 都有 PLL。
- USBPLL EN: 使能 USB 时钟专用锁相环，使能后 USBSRC 对应的单选框变成可选状态 (蓝色)。反之关闭 USBPLL 之后对应的单选框变成红色。USBPLL 会根据其输入源的时钟尽量自动输出 48MHz 的 USB 时钟，如果无法计算出 48MHz 时钟将会标记为 error，此时需要用户修改外部晶振 HSE 的值。不是所有型号的 MCU 都有 USBPLL。
- AHB Prescaler: 可以对 CK_SYS 进行预分频设置，以输出更低的 CK_AHB。
- HSI 与 LSI 推荐一直打开，所以是不可设置状态。
- 红色的单选框表明不能设置，需要打开前级的开关，使其变成蓝色才是可设置状态。
- CK_USB 是提供给 USB 的时钟源，需要为 48MHz；如果不是 48MHz 将会被标记成红色。
- STCLK 是提供给 SysTick 的时钟源；SysTick 也可以设置 CK_AHB 为时钟源。
- FCLK 是自由运行时钟 (free running clock)，与内核的时钟 HCLKC 是相等的。
- HCLKAPB 是提供给 APB Bridge 的时钟源。

4. MISC 页面的设置

MISC 包含 USB、CRC、LVD/BOD、Sleep、S/W DIV、S/W Rand、PDMA 等。

USB

☐ Enable USB

Class demo: HID Vendor Class

CRC

☐ Enable CRC

Mode: CRC-16

LVD

☐ Enable LVD
 ☐ Enable BOD

LVD level: LVD_S_2V25

Sleep

☐ Enable
 DeepSleep1

S/W DIV

☐ Enable Software Division

S/W Rand

☐ Enable Rand

PDMA

	SrcMode	DstMode	Counter	Width	TC Interrupt
<input type="checkbox"/> CH0	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES
<input type="checkbox"/> CH1	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES
<input type="checkbox"/> CH2	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES
<input type="checkbox"/> CH3	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES
<input type="checkbox"/> CH4	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES
<input type="checkbox"/> CH5	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES
<input type="checkbox"/> CH6	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES
<input type="checkbox"/> CH7	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES
<input type="checkbox"/> CH8	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES
<input type="checkbox"/> CH9	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES
<input type="checkbox"/> CH10	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES
<input type="checkbox"/> CH11	ADR_LIN_INC	ADR_LIN_INC	1	8BIT	YES

- USB: 可设置是否使能, 以及 USB class 类型。详细的端点设置、描述符设置请在 source code 里修改。
- CRC: 可设置是否使能, 以及 CRC 类型。
- LVD: 可设置是否使能 LVD, LVD 的电压选择; 可设置是否使能 BOD。
- Sleep: 可设置 Sleep、DeepSleep1、DeepSleep2、PowerDown。
- S/W DIV: 可设置是否启用软件除法器; 带硬件 DIV 的 MCU 无需打开此项。
- S/W Rand: 可设置是否启用软件随机数发生器。
- PDMA: 可以独立使能各个通道, 每个通道可以设置源地址的模式、目标地址的模式、传输的数量、数据宽度以及是否在传输结束时触发中断。PDMA 搭配外设使用时, 需要从 MCU 的开发使用手册中查询该外设对应的 PDMA 通道是哪个, 以使 PDMA 工作正常。例如 HT32F52352 的 ADC 只能使用 PDMA 的 CH0。

5. AFIO 页面的设置

AFIO 页面是最主要的设置页面。AFIO 有 0~15 一共 16 种功能选择，包含有 Default、GPIO、ADC、CMP、GPTM/MCTM、SPI、UART/USART、I²C、SCI、EBI、I²S、SDIO、CSIF、MonoLCD、Other System。不是每个型号的 MCU 都包含以上所有功能，请对照产品规格书的引脚图章节，其中的分配表与 AFIO 页面是一致的。

PIN NO.	AF0 Default	AF1 GPIO	AF2 ADC	AF4 TM	AF5 SPI	
1	PA0		ADC_IN0	GT_CH0	SPI_SCK	USF
2	PA1		ADC_IN1	GT_CH1	SPI_MOSI	USF
3	PA2		ADC_IN2	GT_CH2	SPI_MISO	USF
4	PA3		ADC_IN3	GT_CH3	SPI_SEL	USF
5	PA4		ADC_IN4	GT_CH0	SPI_SCK	UR
6	PA5		ADC_IN5	GT_CH1	SPI_MOSI	UR
7	PA6		ADC_IN6	GT_CH2	SPI_MISO	
8	PA7		ADC_IN7	GT_CH3	SPI_SEL	
9	CLDO					
10	VDD					
11	VSS					
12	nRST					
13	NC					
14	NC					
15	PB12				SPI_MISO	UR
16	XTALIN	PB13				UR
17	XTALOUT	PB14				UR
18	PA9_BOOT				SPI_MOSI	
19	SWCLK	PA12				
20	SWDIO	PA13				
21	PA14			GT_CH0	SPI_SEL	USF
22	PA15			GT_CH0	SPI_SCK	USF
23	PB0			GT_CH1	SPI_MOSI	USF
24	PB1			GT_CH1	SPI_MISO	USF

- 绿色表示已选中的功能，灰色为不能修改的引脚。
- 当需要某个功能，请用鼠标双击对应位置，例如双击 I2C_SCL/I2C_SDA，双击的位置将会变成绿色，功能标签也会多显示出 I²C，详见后面章节的“I²C 页面的设置”。
- 右侧的 MCU 引脚图周边标示出当前每个脚的 AFIO 功能，也可以鼠标点击引脚从弹出的下拉菜单选择 AFIO 功能。
- 鼠标靠近引脚对应的引脚会标示为淡红色。
- 引脚的 AFIO 设置为非默认值则会标示为绿色。
- 鼠标滚轮可以控制放大 / 缩小引脚图，鼠标右键按住拖动可以移动引脚图。也可以通过下方的按键控制缩放。

6. GPIO 页面的设置

GPIO 页面用于设置 I/O 方向、输入使能、上拉 / 下拉选择、输出状态、开漏输出和输出电流。

PIN NO.	GPIO Name	Direction	Input Enable	Pull State	Output State	Open Drain	Current
1	PA0	Input	Disable	NoPull	Low	Disable	4mA
2	PA1	Input	Enable	PullUp	Low	Disable	4mA
3	PA2	Output	Disable	NoPull	High	Enable	16mA
4	PA3	Input	Disable	NoPull	Low	Disable	4mA
5	PA4	Input	Disable	NoPull	Low	Disable	4mA
6	PA5	Input	Disable	NoPull	Low	Disable	4mA
7	PA6	Input	Disable	NoPull	Low	Disable	4mA
8	PA7	Input	Disable	NoPull	Low	Disable	4mA
9	CLDO						
10	VDD						
11	VSS						
12	nRST						
13	NC						
14	NC						
15	PB12	Input	Disable	NoPull	Low	Disable	4mA
16	PB13	Input	Disable	NoPull	Low	Disable	4mA
17	PB14	Input	Disable	NoPull	Low	Disable	4mA
18	PA9	Input	Disable	NoPull	Low	Disable	4mA
19	PA12	Input	Disable	NoPull	Low	Disable	4mA
20	PA13	Input	Disable	NoPull	Low	Disable	4mA
21	PA14	Input	Disable	NoPull	Low	Disable	4mA
22	PA15	Input	Disable	NoPull	Low	Disable	4mA
23	PB0	Input	Disable	NoPull	Low	Disable	4mA
24	PB1	Input	Disable	NoPull	Low	Disable	4mA
25	PB2	Input	Disable	NoPull	Low	Disable	4mA

- 绿色表示在 AFIO 页面里，对应的 I/O 已经被设置成 GPIO 功能。
- 灰色表示在 AFIO 页面里，对应的 I/O 没有被设置成 GPIO 功能，例如图上的 PB13、PB14；或者是不可设置的引脚，例如 VDD、VSS。
- 当设置输入 I/O 时，应该设置的列有 Direction、Input Enable、Pull State。
- 当设置输出 I/O 时，应该设置的列有 Direction、Output State、Open Drain 和 Current。
- 修改过的位置将会标示为黄色。

7. EXTI 页面的设置

EXTI 有从 0~15 一共 16 个 channel，可以选择映射到哪一个脚，可以选择是用于触发中断还是触发唤醒功能，还可以设置中断的类型、Debounce 以及唤醒电平类型。

	NO.	EXTI Name	Port	Purpose	Interrupt Trigger Type	Debounce	Wakeup Trigger Type
	1	EXTI0	PA0	Interrupt	POSITIVE_EDGE	100	HIGH_LEVEL
	2	EXTI1	PA1	Wakeup	POSITIVE_EDGE	0	LOW_LEVEL
	3	EXTI2		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	4	EXTI3		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	5	EXTI4		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
▶	6	EXTI5		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	7	EXTI6		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	8	EXTI7		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	9	EXTI8		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	10	EXTI9		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	11	EXTI10		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	12	EXTI11		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	13	EXTI12		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	14	EXTI13		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	15	EXTI14		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL
	16	EXTI15		Interrupt	POSITIVE_EDGE	0	HIGH_LEVEL

- 当一个 I/O 在 AFIO 和 GPIO 页面里设置成 GPIO+Input 模式，就可以在本页面 Port 列里的下拉列表里刷新出来，例如 PA2 被设置成 Output 了，那 EXTI2 的 Port 就没有 PA2。如果 PA2 已经设置成 EXTI，之后再修改 PA2 为 Output，则 EXTI 中 Port 列的设定会清除 PA2。
- EXTI0 只能选择 PA0、PB0、PC0 或者是 PD0 其中一个。同理，EXTI1 只能选择 PA1、PB1、PC1 或者是 PD1 其中一个。
- Purpose 可以选择用于产生中断还是唤醒，或者是两种都有。
- 中断类型可以设置为高电平、低电平、上升沿、下降沿或者是两种边沿都触发。
- Debounce 是设置中断的抖动抑制，可以在 0~65535 之间设置，越大则抑制抖动的能力越强，不过也会带来迟滞，需要合理设置。
- 唤醒类型可以选择高电平或低电平。

8. Timer 页面的设置

Timer 页面可以设置选用的 MCU 所有 Timer，包括 BFTM、WDT、RTC、SysTick、GPTM、MCTM、PWM、SCTM。不同 MCU 拥有的标签不相同。

BASIC GPTM0 GPTM1 MCTM0 MCTM1

BFTM0

☒ Enable BFTM ☒ Enable Interrupt

Cycle time: ms [1 ~ 44739]

BFTM1

☐ Enable BFTM ☒ Enable Interrupt

Cycle time: ms [1 ~ 44739]

WDT

☐ Enable WDT ☒ Enable WDT Reset MCU

WDT Reload Value [1~4095]:

WDT Prescaler Selection:

WDT Sleep Halt mode:

RTC

☐ Enable RTC ☒ Enable Interrupt

☐ Calendar function

Clock Source:

Prescaler:

SysTick

☐ Enable SysTick ☒ Enable Interrupt

Cycle time: ms [1 ~ 174]

- BFTM0/1：可设置是否使能、是否产生中断以及周期。
- WDT：可设置是否使能 WDT、WDT 的重载值、预分频、溢出时是否复位 MCU 以及休眠模式里是否暂停 WDT。
- RTC：可设置是否使能、是否产生中断、预分频以及日历的功能。
- SysTick：可设置是否使能、是否产生中断以及周期。

BASIC **GPTM0** GPTM1 MCTM0 MCTM1

☒ Timer Mode

☐ Enable Timer ☒ Enable Interrupt

Cycle time: ms [1 ~ 44739]

☐ Output PWM Mode (Set AFIO at first)


Frequency: Hz Range: [1 ~ 960000]
Note: Please edit the code if you need freq < 1.

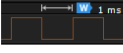
Resolution: 100 1000


CH0 On Duty: % CH2 On Duty: %


CH1 On Duty: % CH3 On Duty: %

☐ Input Capture Mode (Set AFIO at first)

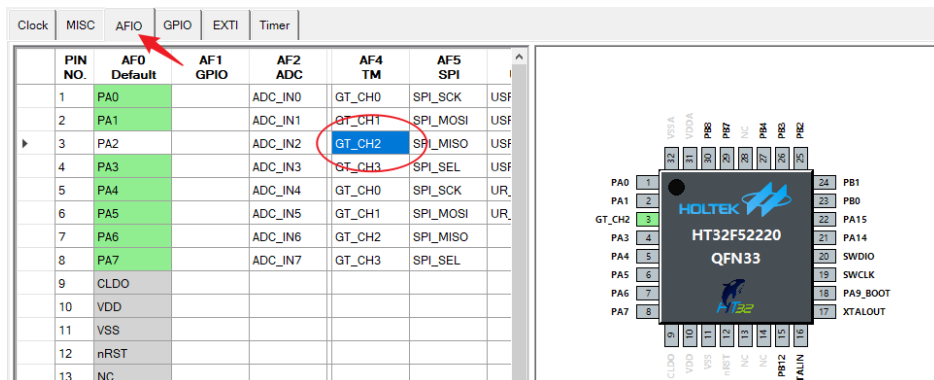
☐ Enable CH0 TI0 Capture Low Level  1 ms

☐ Enable CH1 TI1 Capture Low Level  1 ms

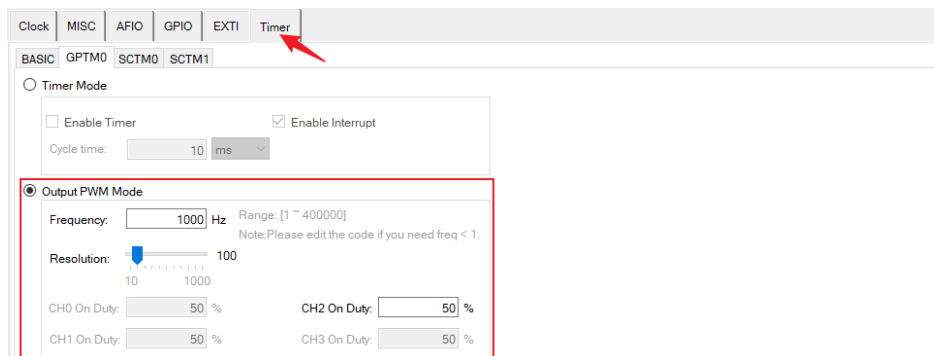
☐ Enable CH2 TI2 Capture Low Level  1 ms

☐ Enable CH3 TI3 Capture Low Level  1 ms

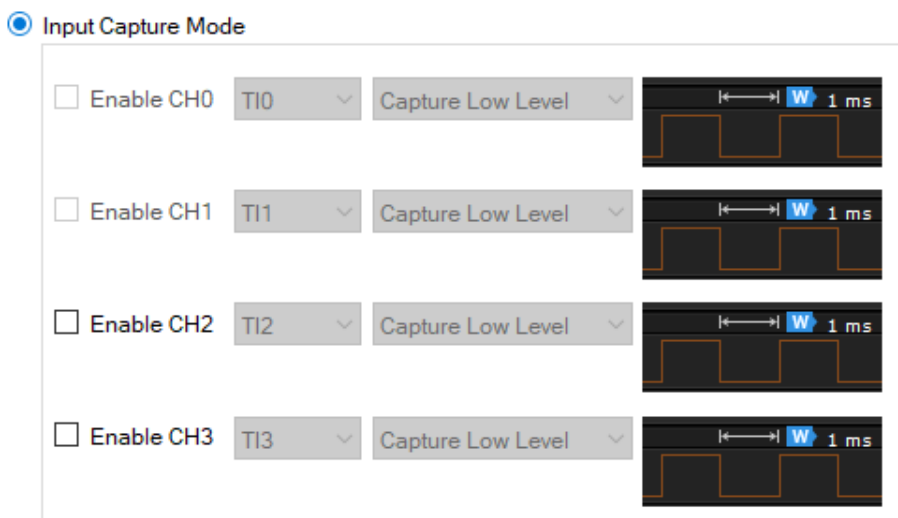
- 如果 AFIO 页面里未选择 GPTM，则本页面里 GPTM 只有基础的 Timer Mode，用于基础的定时器功能，可设置是否使能、是否产生中断以及周期。
- 如果 AFIO 页面里选择了 GT_CH2，则本页面可以设置为 PWM 输出模式或者输入捕捉模式。



- 当其为 PWM 输出模式时，可设置 PWM 的频率、占空比。GT_CH0、GT_CH1、GT_CH3 为灰色是因为没在 AFIO 页面里选中。



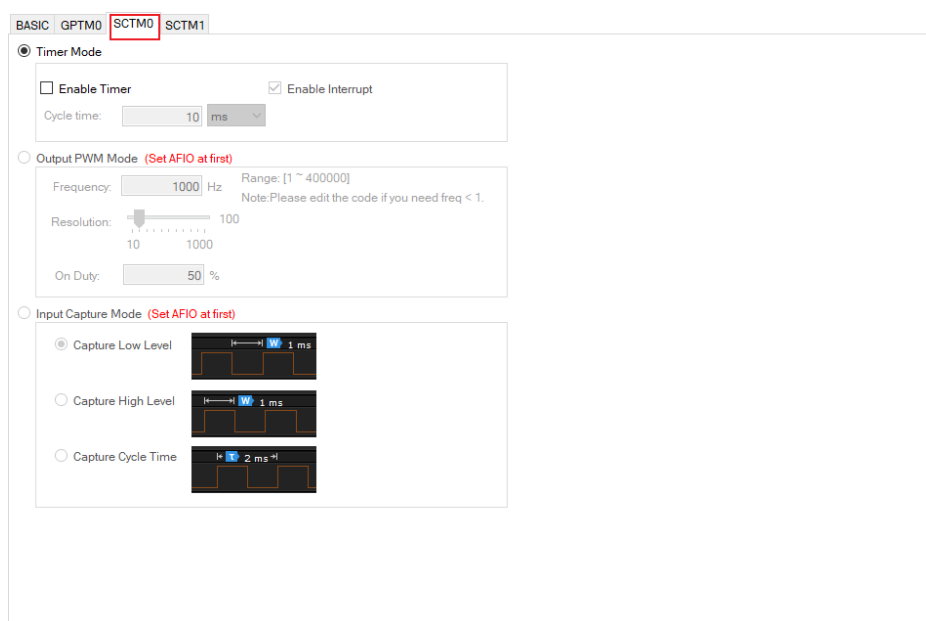
- 当其为输入捕捉模式时，可设置捕捉高电平宽度、低电平宽度或者周期。因为 GT_CH2 的特殊性，使能 CH2 可以同时使用 CH2、CH3 的捕捉功能，详细请参考 MCU 的开发使用手册。



- MCTM 除了类似 GPTM 的功能之外，还有更多功能，本软件暂只提供与 GPTM 类似的设置，其它更高级的设置请参考开发使用手册自行修改 source code 实现。

MCTM 类似的设置项参考 GPTM。

- PWM 与 GPTM 相比，少了捕捉功能，其它功能类似 GPTM。
PWM 并不是每个型号的 MCU 都有，若有，则会显示本页面。
- SCTM 与 GPTM 相比，只有一个通道，其它功能类似 GPTM。
SCTM 并不是每个型号的 MCU 都有，若有，则会显示本页面。



The screenshot displays the configuration interface for the SCTM0 module. At the top, there are tabs for BASIC, GPTM0, SCTM0 (which is highlighted with a red box), and SCTM1. Below the tabs, the 'Timer Mode' is selected with a radio button. Under 'Timer Mode', there are two checkboxes: 'Enable Timer' (unchecked) and 'Enable Interrupt' (checked). A 'Cycle time' field is set to '10' with a unit dropdown menu showing 'ms'. Below this, the 'Output PWM Mode (Set AFIO at first)' is selected with a radio button. It includes a 'Frequency' field set to '1000' Hz, a range indicator '[1 ~ 400000]', and a note 'Note: Please edit the code if you need freq < 1.'. A 'Resolution' slider is set to '10' between '10' and '1000'. An 'On Duty' field is set to '50' %.

The 'Input Capture Mode (Set AFIO at first)' is not selected. It contains three radio button options, each with a corresponding waveform diagram and a time scale:

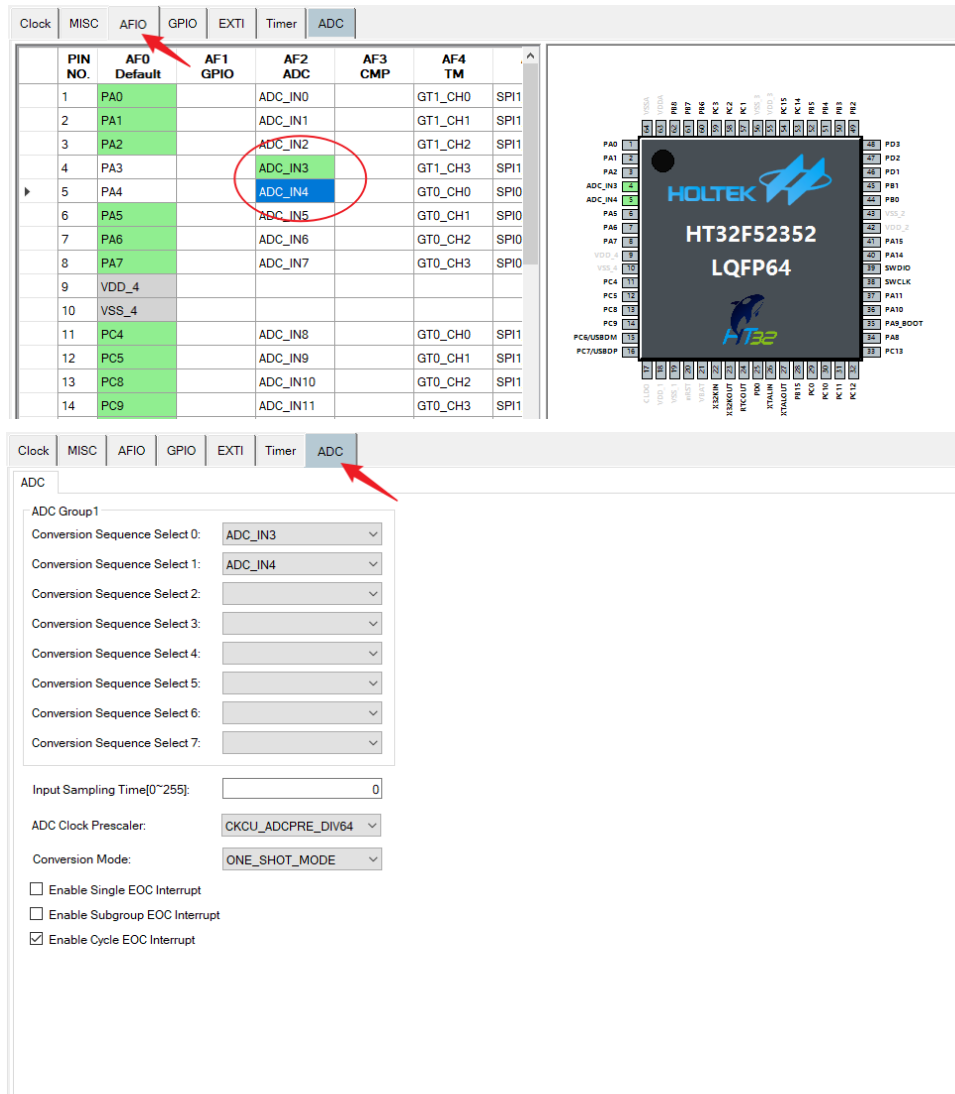
- ☒ Capture Low Level: Shows a square wave with a 1 ms time scale.
- ☐ Capture High Level: Shows a square wave with a 1 ms time scale.
- ☐ Capture Cycle Time: Shows a square wave with a 2 ms time scale.

9. ADC/DAC 页面的设置

本页面需要先在 AFIO 页面里选择了 ADC 或者 DAC，才会显示出来。

9.1 ADC 的说明

先在 AFIO 页面选中 ADC_IN3、ADC_IN4。



Clock	MISC	AFIO	GPIO	EXTI	Timer	ADC
PIN NO.	AF0 Default	AF1 GPIO	AF2 ADC	AF3 CMP	AF4 TM	
1	PA0		ADC_IN0		GT1_CH0	SPI1
2	PA1		ADC_IN1		GT1_CH1	SPI1
3	PA2		ADC_IN2		GT1_CH2	SPI1
4	PA3		ADC_IN3		GT1_CH3	SPI1
5	PA4		ADC_IN4		GT0_CH0	SPI0
6	PA5		ADC_IN5		GT0_CH1	SPI0
7	PA6		ADC_IN6		GT0_CH2	SPI0
8	PA7		ADC_IN7		GT0_CH3	SPI0
9	VDD_4					
10	VSS_4					
11	PC4		ADC_IN8		GT0_CH0	SPI1
12	PC5		ADC_IN9		GT0_CH1	SPI1
13	PC8		ADC_IN10		GT0_CH2	SPI1
14	PC9		ADC_IN11		GT0_CH3	SPI1

ADC Group1

Conversion Sequence Select 0: ADC_IN3

Conversion Sequence Select 1: ADC_IN4

Conversion Sequence Select 2:

Conversion Sequence Select 3:

Conversion Sequence Select 4:

Conversion Sequence Select 5:

Conversion Sequence Select 6:

Conversion Sequence Select 7:

Input Sampling Time[0~255]: 0

ADC Clock Prescaler: CKCU_ADCPRE_DIV64

Conversion Mode: ONE_SHOT_MODE

☐ Enable Single EOC Interrupt

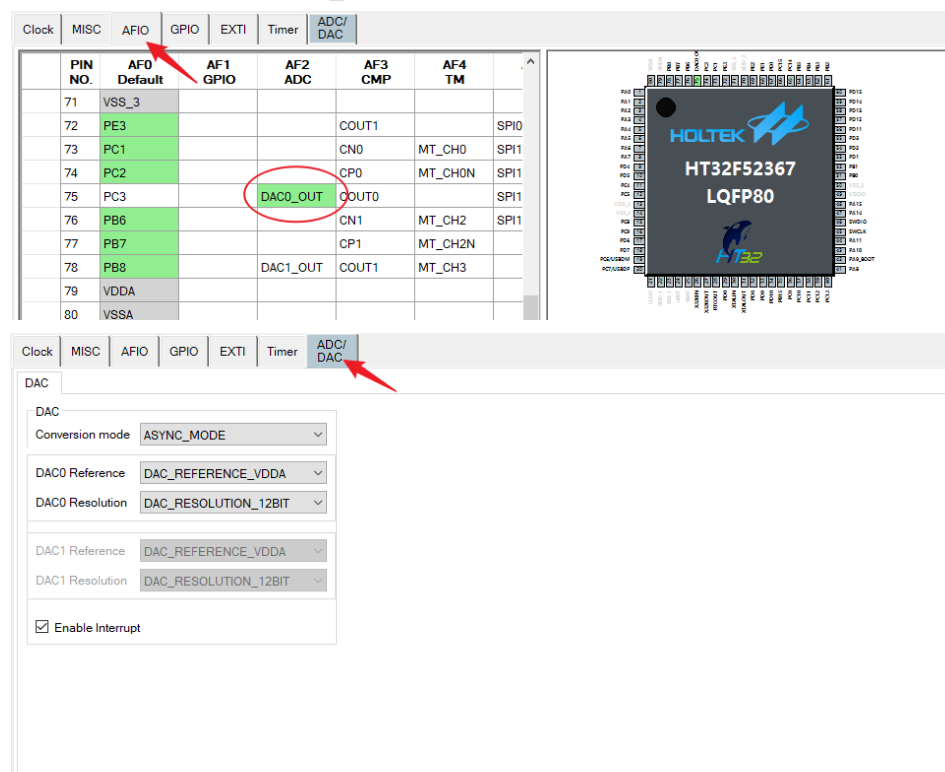
☐ Enable Subgroup EOC Interrupt

☒ Enable Cycle EOC Interrupt

- ADC 组成员会根据 AFIO 页面选择的 ADC 通道，自动填好。
- 可以设置采样时间，延长采样时间可以让采样值更准确，不过缺点是使得 ADC 转换总时间变长，所以需要合理设置。
- 可以设置 ADC 的时钟预分频，设置转换模式，设置转换完成的中断类型。
- 对于普通应用，本页面基本无需修改。

9.2 DAC 的说明

先在 AFIO 页面选中 DAC0_OUT，则 DAC 页面会显示出来。



- 不是所有型号的 MCU 都具有 DAC 功能，具体请查阅开发使用手册。
- DAC 可以设置转换模式为异步或者同步模式，参考电压 VDDA 或者 VREF，转换分辨率 12-bit 或者 8-bit，转换完成是否触发中断。
- 若对应的通道在 AFIO 页面被选中，该通道才是可选择状态，否则为灰色，如上图 DAC1。

10. CMP 页面的设置

若所选 MCU 具有 CMP 功能，并且 AFIO 页面选中了 CN、CP 或者 COUT，则 CMP 页面会显示出来。

The image shows two screenshots from the HT32F52352 LQFP64 pinout and the CMP configuration page. The top screenshot shows the pinout table with the AFIO tab selected, and the bottom screenshot shows the CMP configuration page with the CMP tab selected.

Pinout Table (AFIO tab):

PIN NO.	AF0 Default	AF1 GPIO	AF2 ADC	AF3 CMP	AF4 TM	
50	PB3				MT_CH2N	SPI0
51	PB4				MT_BRK	SPI0
52	PB5				MT_BRK	SPI0
53	PC14				MT_CH3	
54	PC15					
55	VDD_3					
56	VSS_3					
57	PC1			CN0	MT_CH0	SPI1
58	PC2			CP0	MT_CH0N	SPI1
59	PC3			COUT0		SPI1
60	PB6			CN1	MT_CH2	SPI1
61	PB7			CP1	MT_CH2N	
62	PB8			COUT1	MT_CH3	
63	VDDA					
64	VSSA					

CMP Configuration Page:

The CMP configuration page shows the internal block diagram of the comparator. The input is selected via a multiplexer (Mux) to either CP (Comparator Pin) or CN (Comparator Non-Inverting Input). The output is selected via a multiplexer (Mux) to either COUT (Comparator Output) or COUTPOL (Comparator Output Polarity). The output is then connected to the AFIO (Alternate Function I/O) block.

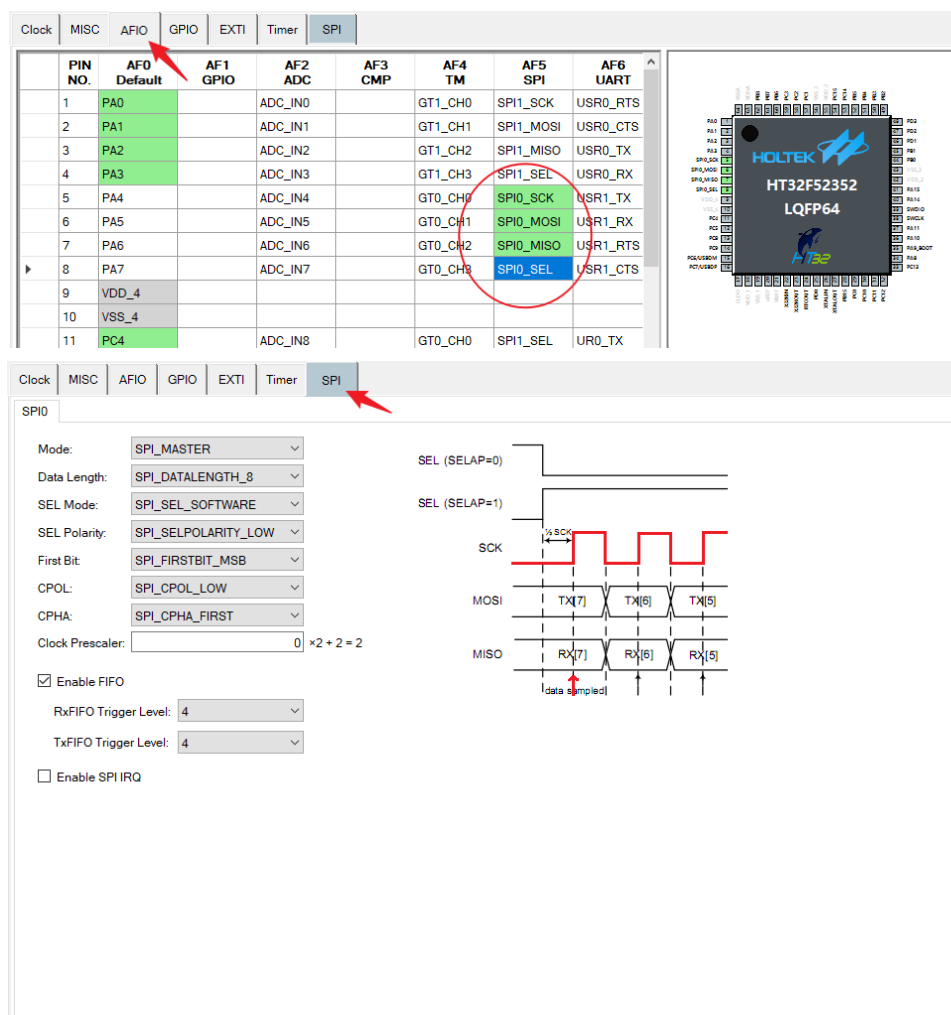
Configuration Options:

- Hysteresis:** 00: No
- Speed Mode:** Low speed
- CVREN:** ☐ (Comparator Voltage Reference Enable)
- CVREF:** ☐ (Comparator Voltage Reference)
- CVROE:** ☐ (Comparator Voltage Reference Output Enable)
- CVRVAL[5:0]:** 0
- CMP Output Selection:** 000: No selection
- ☐ Enable CMP output rising edge interrupt
- ☐ Enable CMP output rising edge detection
- ☐ Enable CMP output falling edge interrupt
- ☐ Enable CMP output falling edge detection

- 本页面显示所有 AFIO 页面选中的 CMP。
- 图形化的选择，按需点击选择功能。
- 可设置迟滞 (Hysteresis)，高速或低速模式，CN 的参考源来自外部或者内部；内部参考源可以设置是否分压，分压等级；COUT 输出是否触发中断，是否唤醒 MCU 等等。

11. SPI 页面的设置

SPI 是很常用的功能。在 AFIO 页面里选中任意 SPI 之后，SPI 页面就会显示出来。

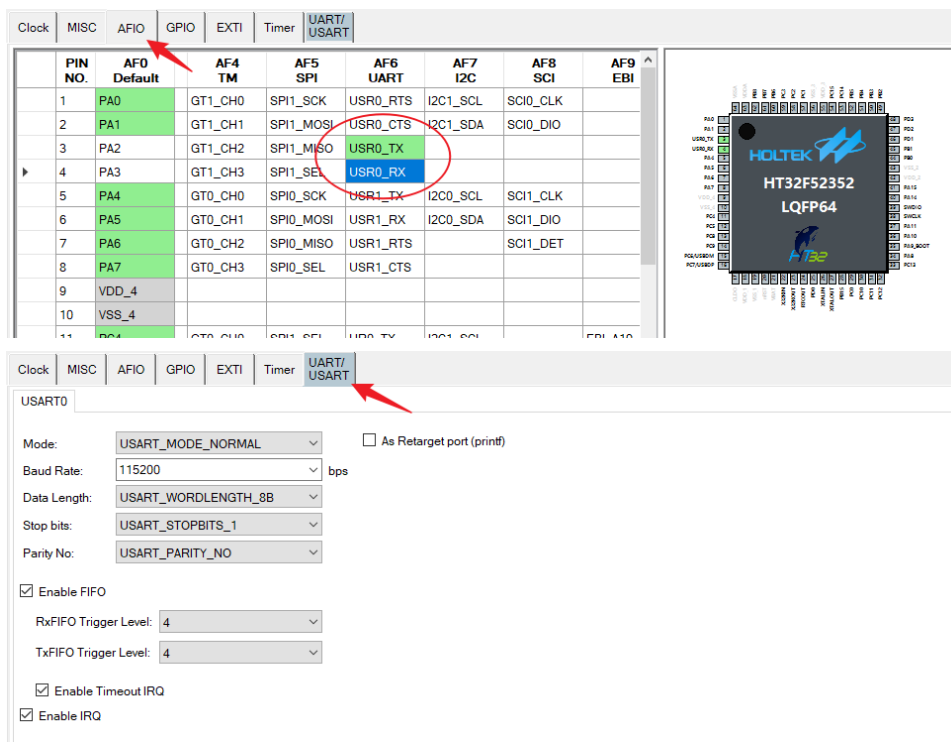


The screenshot displays the HT32_CodeConfig software interface. The top navigation bar includes tabs for Clock, MISC, AFIO, GPIO, EXTI, Timer, and SPI. The AFIO tab is selected, showing a table of pins and their functions. The SPI1_SCK, SPI1_MOSI, SPI1_MISO, and SPI1_SEL pins are highlighted in green. The SPI page is also visible, showing configuration options for SPI0. The Mode is set to SPI_MASTER, Data Length is SPI_DATALENGTH_8, SEL Mode is SPI_SEL_SOFTWARE, SEL Polarity is SPI_SELPOLARITY_LOW, First Bit is SPI_FIRSTBIT_MSB, CPOL is SPI_CPOL_LOW, CPHA is SPI_CPHA_FIRST, and Clock Prescaler is 0. The FIFO is enabled with Rx and Tx trigger levels set to 4. A waveform diagram on the right shows the timing of SEL, SCK, MOSI, and MISO signals.

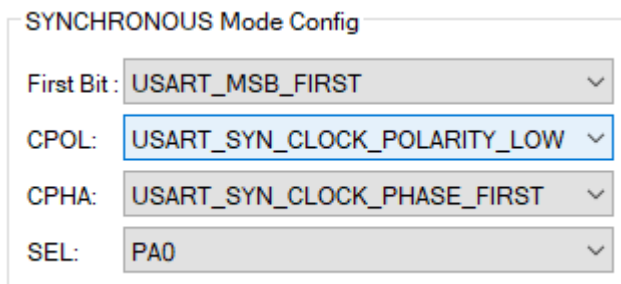
- 本页面显示所有 AFIO 页面选中的 SPI。
- SPI 可以设置主从模式、数据长度、SEL 模式、MSB/LSB、SPI 的 4 种模式以及时钟预分频、FIFO 深度、是否使能中断。
- 右侧会根据选择的选项，显示波形图示，直观的反馈选项是否正确。

12. UART/USART 页面的设置

UART/USART 也是很常用的功能。在 AFIO 页面里选中任意 UART/USART 之后，本页面就会显示出来。

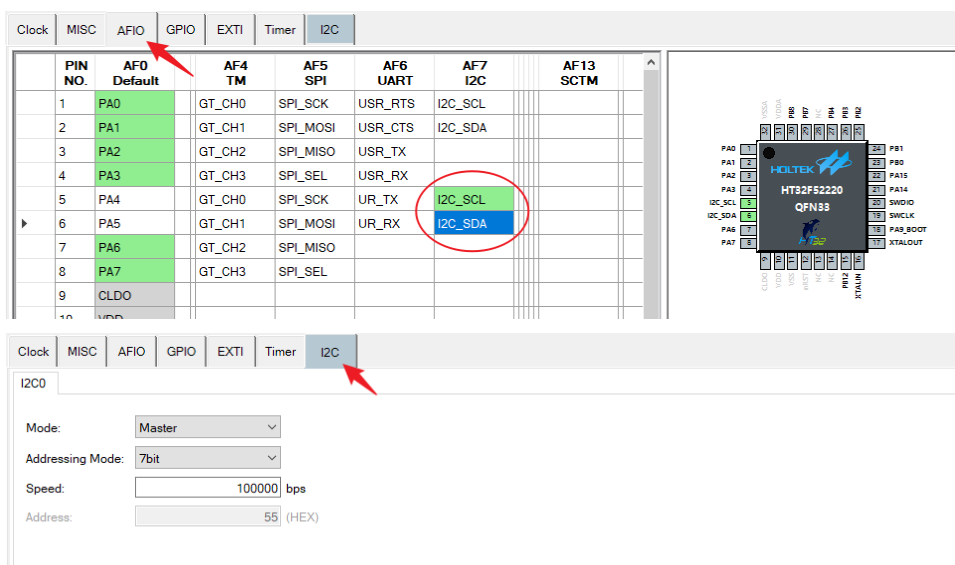


- 本页面显示所有 AFIO 页面选中的 UART/USART。
- 如果选中某个 As Retarget port 作为打印的串口，则其它 UART/USART 就不能再做打印串口。
- USART 可设置成普通串口模式、IRDA、RS485 或者同步模式，即把 USART 当成 SPI 使用，UART 没有这些功能。
- USART 有 FIFO 的设置以及 Timeout 中断设置，UART 没有这些功能。
- 可设置波特率，数据长度，停止位数，校验位。
- 同步模式有更多的类似 SPI 选项，如下图。



13. I²C 页面的设置

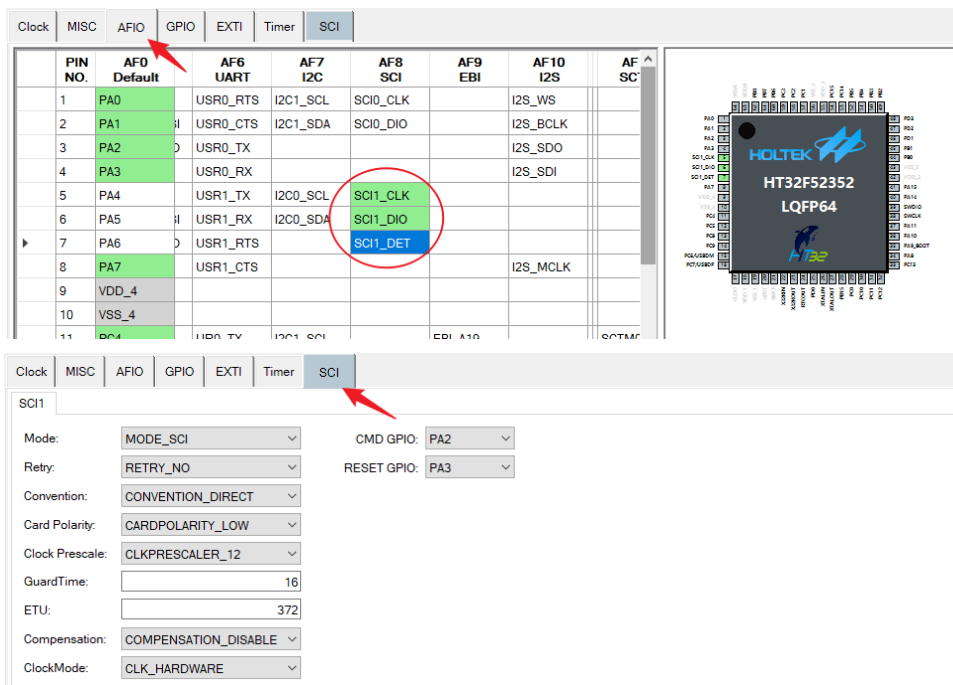
I²C 也是很常用的功能。在 AFIO 页面里选中任意 I²C 之后，本页面就会显示出来。



- 本页面显示所有 AFIO 页面选中的 I²C。
- 可以设置主从模式，地址 7-bit 或者 10-bit 模式；主模式可以设置速度，从模式可以设置被访问时的设备地址。
- 设置很简单，不过最终输出的代码不光会包含读与写操作，还有借助 BFTM 的纠错机制，防止干扰信号或者反复断开 / 接上等情况下的导致 I²C 锁死状况的出现。

14. SCI 页面的设置

在 AFIO 页面里选中任意 SCI 之后，本页面就会显示出来。

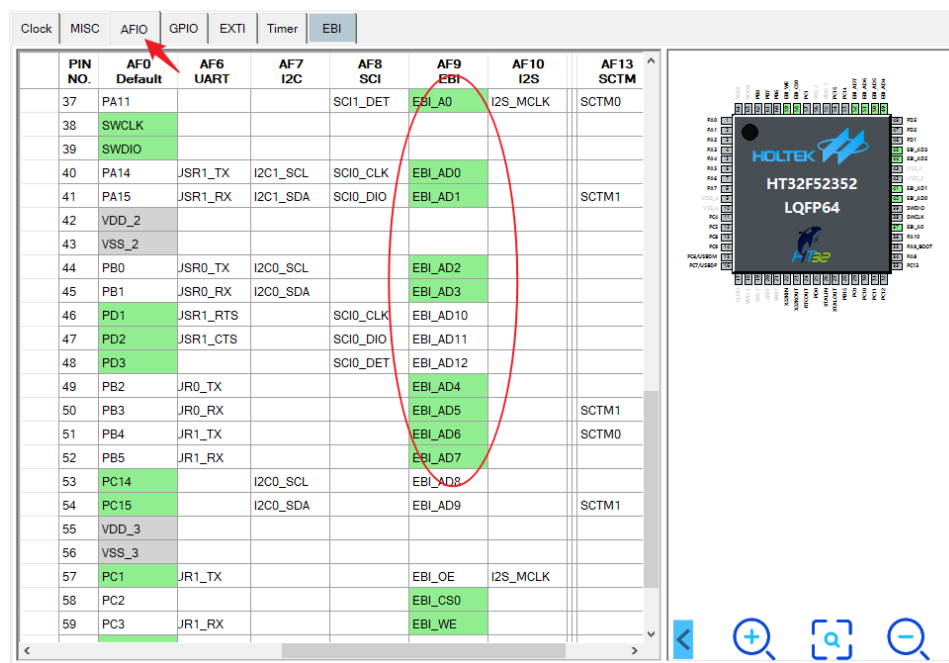


- 本页面显示所有 AFIO 页面选中的 SCI。
- 可以设置模式 (Mode)、重试 (Retry)、协议 (Convention)、卡极性 (Card Polarity)、时钟预分频 (Clock Prescaler)、保护时间 (Guard Time)、基本时间单元 (ETU)、补偿 (Compensation)、时钟模式 (Clock Mode)。
- CMD GPIO、RESET GPIO 会列出所有可用的 GPIO，供选择。

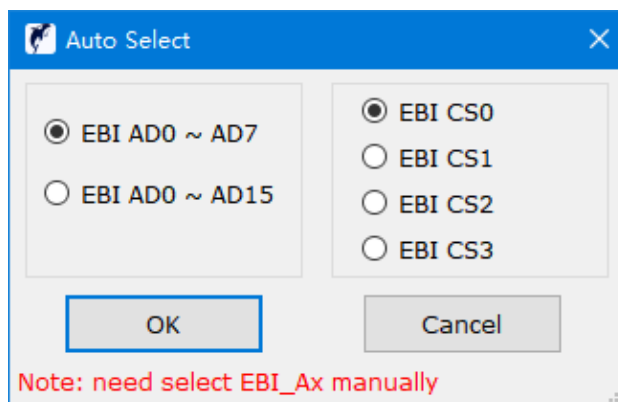
15. EBI 页面的设置

15.1 EBI 基础设置

在 AFIO 页面里选中任意 EBI 之后，本页面就会显示出来。



因为 EBI 引脚固定，并且引脚较多，第一次选择 EBI 功能会有如下的辅助选择窗口出现：



例如勾选 AD0~AD7 + EBI CS0，点击 OK，则 AFIO 页面中相关位置会自动被选中。

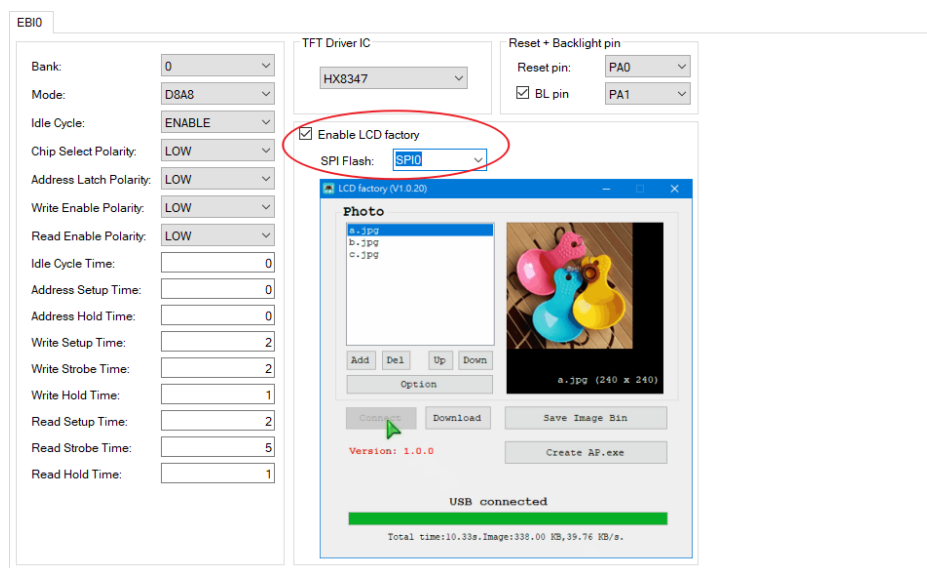
- EBI 的选项很多，如果不太清楚具体用途，就不要修改。

15.2 驱动 TFT 屏

- 如果需要用 EBI 来驱动 TFT 屏，则设置右侧的 TFT 驱动 IC 的类型，屏的复位脚，背光控制脚。背光也可能是 PWM 信号，那就请用 TM 或者 SCTM 页面设置一个 PWM。
- 到此处为止，输出的代码可以让屏显示一个纯色，例如全红色。

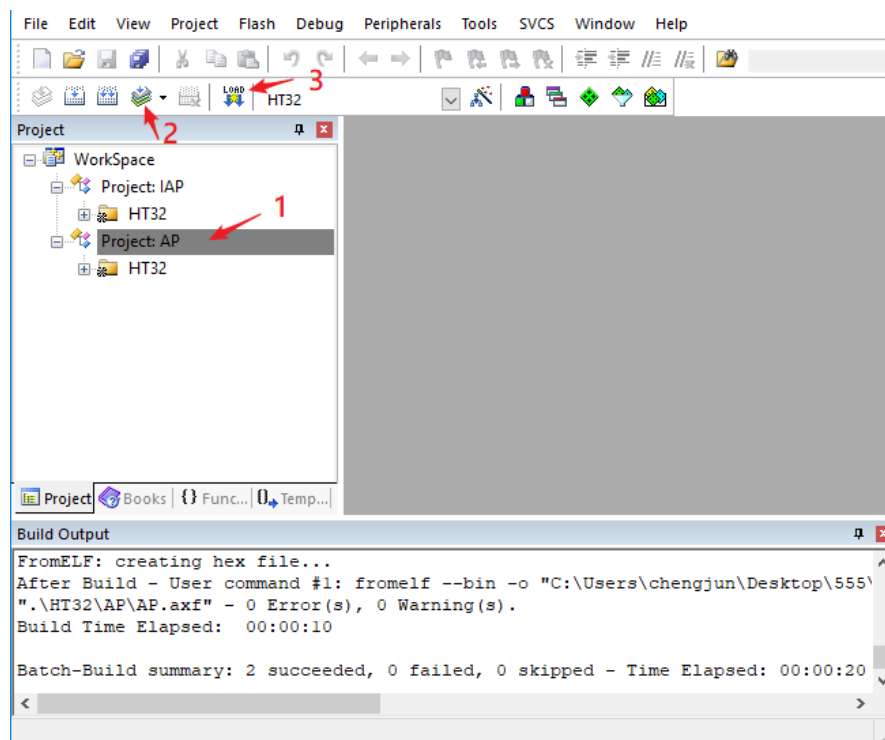
15.3 LCD 平台

- 如果还需要显示图片，更新图片等功能，需要使用合泰提供的 LCD 平台来协助。
- LCD 平台是一套包含上位机程序，通过 USB 接口更新图片的整套解决方案。因为相对 MCU ROM 而言图片数据比较大，所以都保存在外部 SPI Flash 内。刷图时从 SPI Flash 读出图片数据送到 TFT 屏上显示。
- 先在 AFIO 页面选中一个 SPI 用于连接外部的 SPI Flash，再在本页面里使能 LCD factory 并选择 SPI，就激活了 LCD 平台的功能。

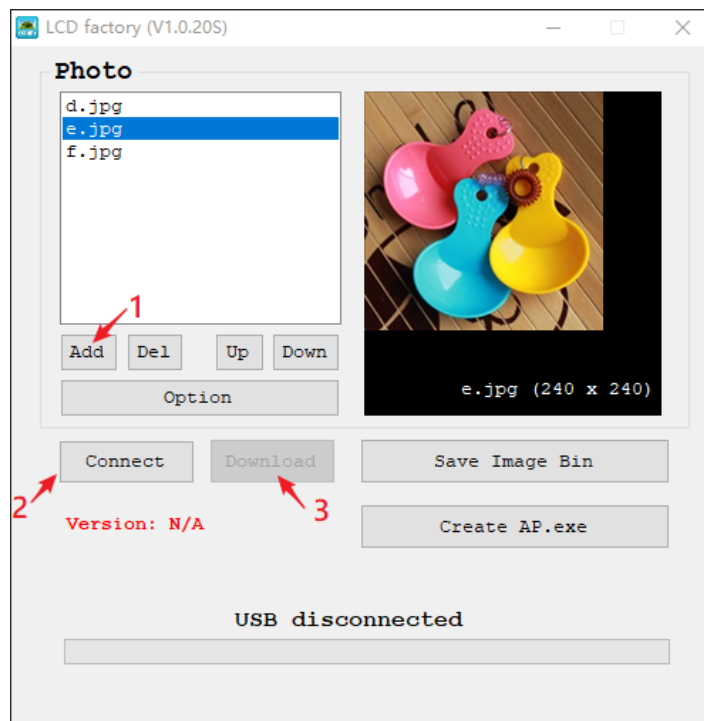


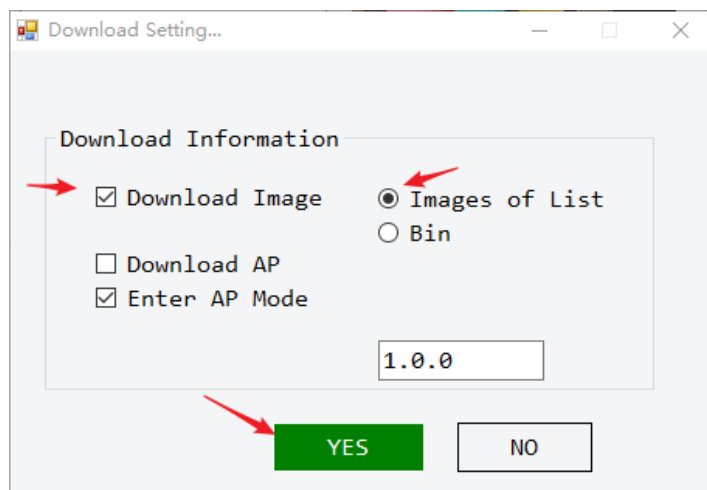
- 界面有动画显示操作流程。
- 输出的文件包含有上位机程序，和固件的源代码。
- 支持 8-bit EBI、16-bit EBI，支持图片压缩功能以尽量减小使用的 SPI Flash 大小。
- 详细使用步骤：
 - ◆ 生成的工程包含 IAP、AP 两个工程，点击 Multiproject 可打开两个工程，如下图：

Name	Date modified	Type	Size
AP.uvoptx	2022/12/07 10:28	UVOPTX File	4 KB
AP.uvprojx	2022/12/07 10:28	Microvision5 Project	27 KB
Decompress_Lib_M0p_Keil.lib	2020/07/14 09:30	Object File Library	12 KB
ht32_op.s	2019/08/02 20:14	S File	19 KB
HT32F5xxx_01_DebugSupport.ini	2022/06/22 17:59	Configuration sett...	6 KB
iap.s	2022/12/07 10:28	S File	1 KB
IAP.uvoptx	2022/12/07 10:28	UVOPTX File	4 KB
IAP.uvprojx	2022/12/07 10:28	Microvision5 Project	24 KB
LCD_Platform.uvmpw	2022/12/07 10:28	Microvision Multi-Proj...	1 KB
LcdPlatform_Lib_M0p_Keil.lib	2020/07/14 10:18	Object File Library	46 KB
linker.lin	2022/12/07 10:28	LIN File	2 KB
startup_ht32f5xxx_01.s	2022/12/07 10:28	S File	21 KB
version.s	2022/12/07 10:28	S File	1 KB
version.txt	2022/12/07 10:28	TXT File	1 KB



- ◆ 设置 AP 为 Active Project
- ◆ Batch Build IAP 和 AP 两个工程
- ◆ 下载到 MCU
- ◆ 用 USB 线连接 MCU 到电脑
- ◆ 打开 LCD factory.exe, Add 选择图片, Connect 连接, Download 下载





下载成功后就可以看到 TFT 上图片定时刷新了。

如果失败，检查：

- 屏的 EBI 设定是 8-bit 还是 16-bit;
- 如果屏有 OE 脚，检查 OE 脚的设置；
- 检查 SPI 各个引脚是否设置错误；
- 检查 RESET 脚是否设置错误；
- 检查背光控制脚是否设置错误。

16. I²S 页面的设置

在 AFIO 页面里选中任意 I²S 之后，本页面就会显示出来。

The screenshot displays the HT32_CodeConfig interface. The top section shows the 'AFIO' tab selected, with a table of pins and functions. The 'AF10 I2S' column is highlighted with a red circle. The bottom section shows the 'I2S' configuration page, which includes settings for Mode, Format, Word Width, Mclk Out, Mclk Inv, Bclk Inv, X_Div, Y_Div, N_Div, and a checkbox for 'Recommend X/Y/N for me'.

PIN NO.	AF0 Default	AF7 I2C	AF8 SCI	AF9 EBI	AF10 I2S	AF13 SCTM	AF15 Other
1	PA0	C1_SCL	SCI0_CLK		I2S_WS		
2	PA1	C1_SDA	SCI0_DIO		I2S_BCLK		
3	PA2				I2S_SDO		
4	PA3				I2S_SDI		
5	PA4	C0_SCL	SCI1_CLK				
6	PA5	C0_SDA	SCI1_DIO				
7	PA6		SCI1_DET				
8	PA7				I2S_MCLK		
9	VDD_4						
10	VSS_4						

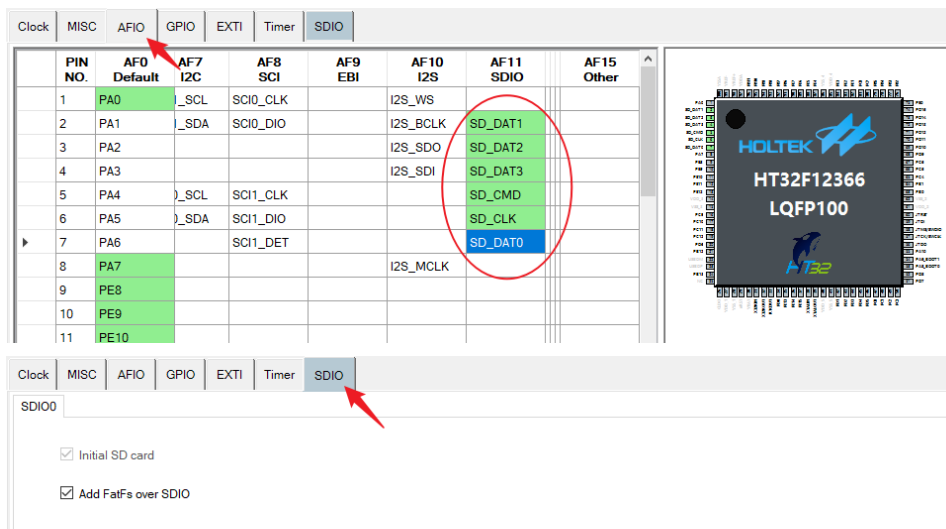
The I2S configuration page shows the following settings:

- Mode: MASTER_TX
- Format: I2S_JUSTIFIED_STEREO
- Word Width: WORDWIDTH_16
- Mclk Out: ENABLE
- Mclk Inv: DISABLE
- Bclk Inv: DISABLE
- X_Div: 1
- Y_Div: 2
- N_Div: 1
- MCLK = 12000000
- BCLK = 6000000
- Sample rate = 187500
- ☐ Recommend X/Y/N for me
- Target sample rate: [dropdown menu]

- 可以设置主从各种模式，数据格式，数据宽度，MCLK，BCLK。
- 通过设置 X、Y、N 的值可以设定采样率。
- 如果是常用的采样率，可以通过选择推荐值，自动计算出 X、Y、N。

17. SDIO 页面的设置

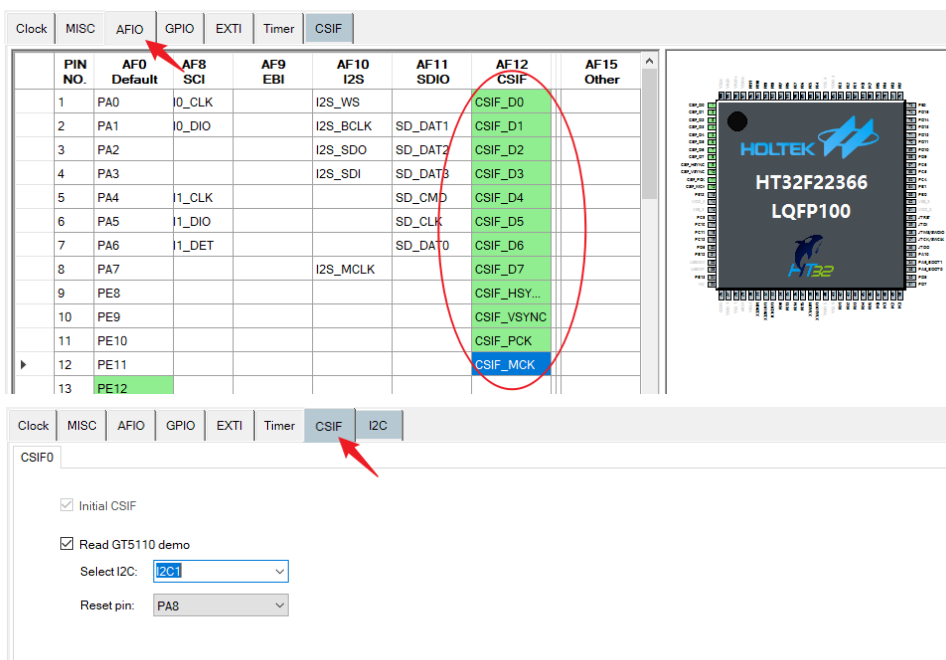
在 AFIO 页面里选中任意 SDIO 之后，本页面就会显示出来。



- 此处的 SDIO 主要用来读写 SD/TF 卡，所以选项很简化。
- 可以设置是否增加 FatFs 文件系统读写 SD/TF 卡。

18. CSIF 页面的设置

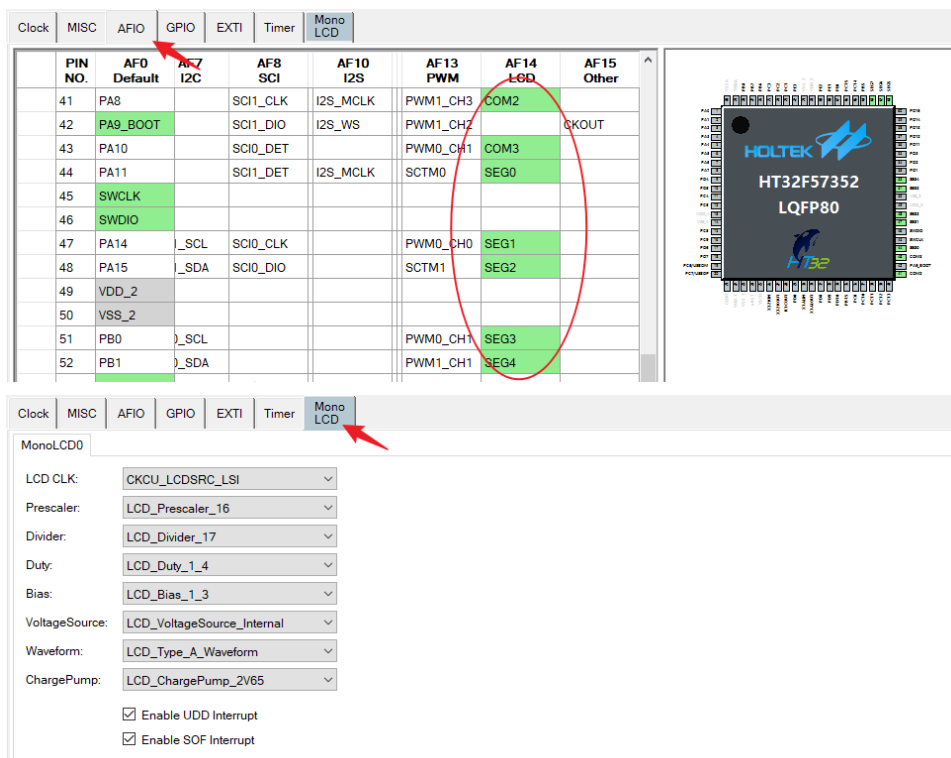
在 AFIO 页面里选中任意 CSIF 之后，本页面就会显示出来。



- 默认使能初始化 CSIF 接口。
- 可以设置是否初始化 GT5110 摄像头模块。初始化模块还需要一组 I²C 和复位脚。

19. MonoLCD 页面的设置

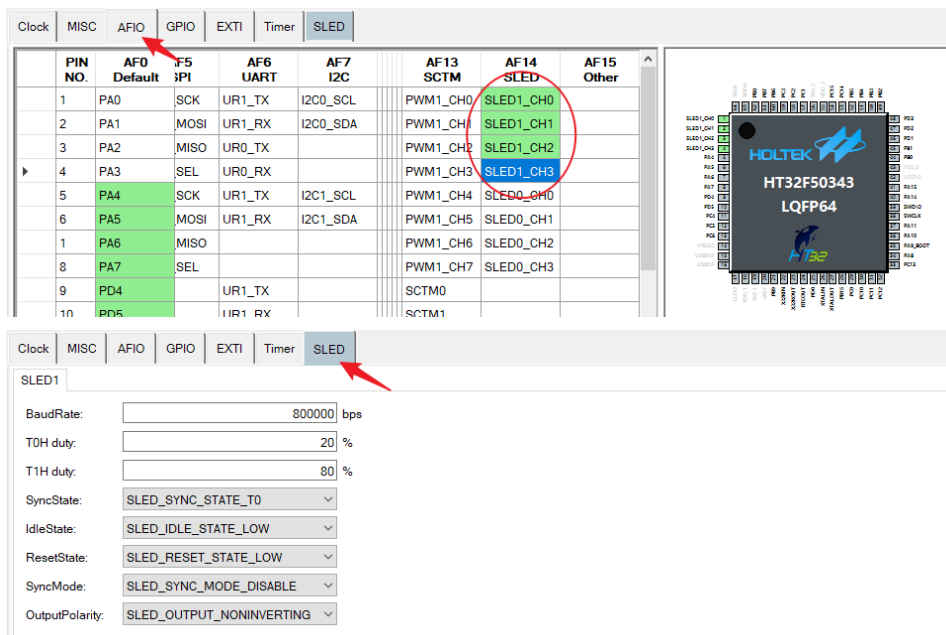
在 AFIO 页面里选中任意 MonoLCD 的 COM 或者 SEG 之后，本页面就会显示出来。



- 可以设置 MonoLCD 的时钟源，预分频，分频器，占空比，偏置，电压源，波形，充电泵电压以及中断使能位。
- 具体如何设置需要按选用的 LCD 屏而定。

20. SLED 页面的设置

在 AFIO 页面里选中任意 SLED 之后，本页面就会显示出来。



PIN NO.	AF0 Default	AF5 I2C	AF6 UART	AF7 I2C	AF13 SCTM	AF14 SLED	AF15 Other
1	PA0	SCK	UR1_TX	I2C0_SCL	PWM1_CH0	SLED1_CH0	
2	PA1	MOSI	UR1_RX	I2C0_SDA	PWM1_CH1	SLED1_CH1	
3	PA2	MISO	UR0_TX		PWM1_CH2	SLED1_CH2	
4	PA3	SEL	UR0_RX		PWM1_CH3	SLED1_CH3	
5	PA4	SCK	UR1_TX	I2C1_SCL	PWM1_CH4	SLED0_CH0	
6	PA5	MOSI	UR1_RX	I2C1_SDA	PWM1_CH5	SLED0_CH1	
7	PA6	MISO			PWM1_CH6	SLED0_CH2	
8	PA7	SEL			PWM1_CH7	SLED0_CH3	
9	PD4		UR1_TX		SCTM0		
10	PD5		UR1_RX		SCTM1		

SLED1

BaudRate: 800000 bps

T0H duty: 20 %

T1H duty: 80 %

SyncState: SLED_SYNC_STATE_T0

IdleState: SLED_IDLE_STATE_LOW

ResetState: SLED_RESET_STATE_LOW

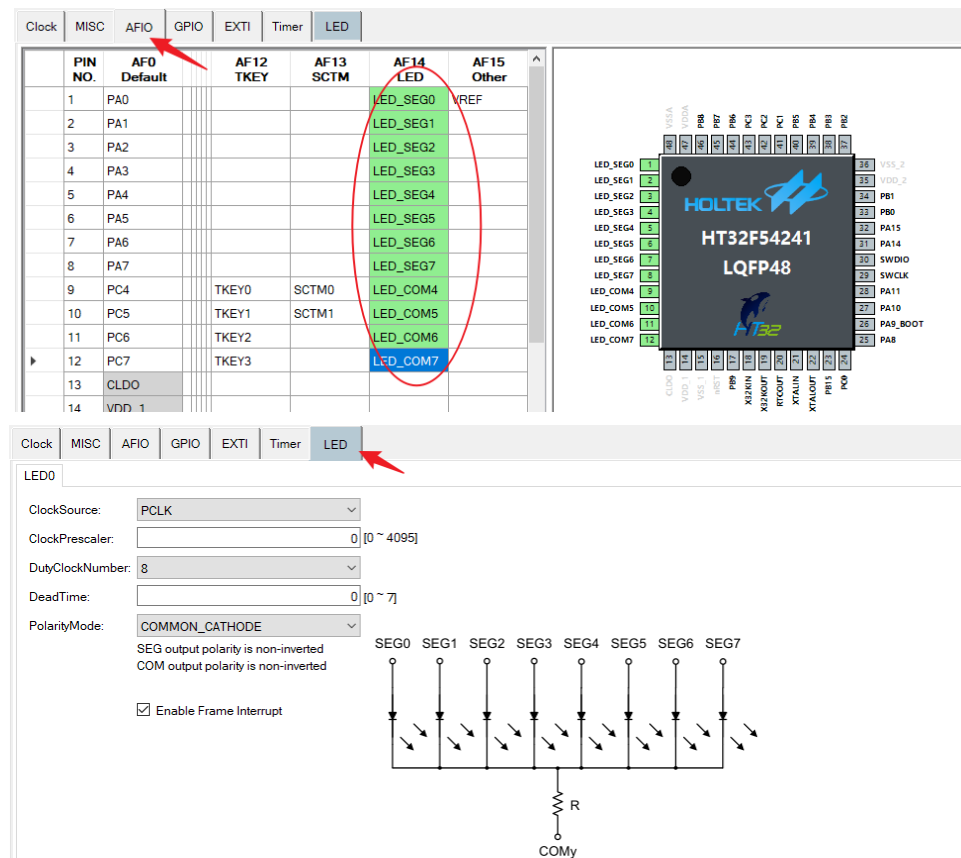
SyncMode: SLED_SYNC_MODE_DISABLE

OutputPolarity: SLED_OUTPUT_NONINVERTING

- 可以设置 SLED 的波特率，占空比，同步状态，空闲状态，复位状态，同步模式以及输出极性。
- 具体如何设置需要按选用的 SLED 而定。

21. LED 页面的设置

在 AFIO 页面里选中任意 LED 之后，本页面就会显示出来。



The screenshot displays the HT32_CodeConfig software interface. The top section shows a table of pins and their default functions. The bottom section shows the LED0 configuration settings, including clock source, prescaler, duty cycle, dead time, and polarity mode. A diagram of the LED connection is also shown.

PIN NO.	AF0 Default	AF12 TKEY	AF13 SCTM	AF14 LED	AF15 Other
1	PA0			LED_SEG0	VREF
2	PA1			LED_SEG1	
3	PA2			LED_SEG2	
4	PA3			LED_SEG3	
5	PA4			LED_SEG4	
6	PA5			LED_SEG5	
7	PA6			LED_SEG6	
8	PA7			LED_SEG7	
9	PC4	TKEY0	SCTM0	LED_COM4	
10	PC5	TKEY1	SCTM1	LED_COM5	
11	PC6	TKEY2		LED_COM6	
12	PC7	TKEY3		LED_COM7	
13	CLDO				
14	VDD1				

LED0 Configuration:

- ClockSource: PCLK
- ClockPrescaler: 0 [0 ~ 4095]
- DutyClockNumber: 8
- DeadTime: 0 [0 ~ 7]
- PolarityMode: COMMON_CATHODE
- SEG output polarity is non-inverted
- COM output polarity is non-inverted
- ☒ Enable Frame Interrupt

Diagram showing the LED connection (SEG0 to SEG7) and the common cathode (COMy) connected to ground through a resistor R.

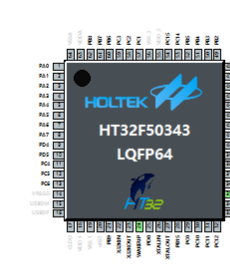
- 可以设置 LED 的时钟源、预分频、Duty Clock、Dead Time 以及 LED 的连接模式 Polarity Mode。

22. Other 页面的设置

本页面包含有 CKOUT、WAKEUP 的相关设置。
在 AFIO 页面里选中 CKOUT 或者 WAKEUP 之后，本页面就会显示出来。

ClockMISCAFIOPGIOEXTITimerOther

PIN NO.	AF0 Default	AF5 IPI	AF6 UART	AF7 I2C	AF13 SCTM	AF14 SLED	AF15 Other
22	X32KIN	SEL	UR1_TX		SCTM0	SLED0_CH0	
23	X32KOUT	SCK	UR1_RX		SCTM1	SLED0_CH1	
24	RTCOUT	MISO	UR0_RX		SCTM0		WAKEUP
25	PD0				SCTM1		
26	XTALIN		UR0_TX		PWM0_CH0	SLED0_CH2	
27	XTALOUT		UR0_RX		PWM0_CH1	SLED0_CH3	
28	PB15	SEL		I2C1_SCL	PWM0_CH2		
29	PC0	SCK		I2C1_SDA	PWM0_CH3		
30	PC10	SEL			PWM0_CH4		
31	PC11	SCK			PWM0_CH5		
32	PC12	MOSI	UR1_TX	I2C0_SCL	PWM0_CH6		
33	PC13	MISO	UR1_RX	I2C0_SDA	PWM0_CH7		
34	PA3		UR1_TX		SCTM0	SLED1_CH0	
35	PA9_BOOT	MOSI			SCTM1	SLED1_CH1	CKOUT
36	PA10	MOSI	UR1_RX		PWM0_CH4	SLED1_CH2	



ClockMISCAFIOPGIOEXTITimerOther

CKOUT

Clock Source: CK_SYS/16 3.7500MHz

WakeUp Pin

No setting required

- 在 AFIO 页面里选中 CKOUT 之后，本页面可以选择 CKOUT 的 7 种时钟源之一，并会实时显示出最终输出的时钟频率值。
- WAKEUP 引脚代码简单，没有可设置项。

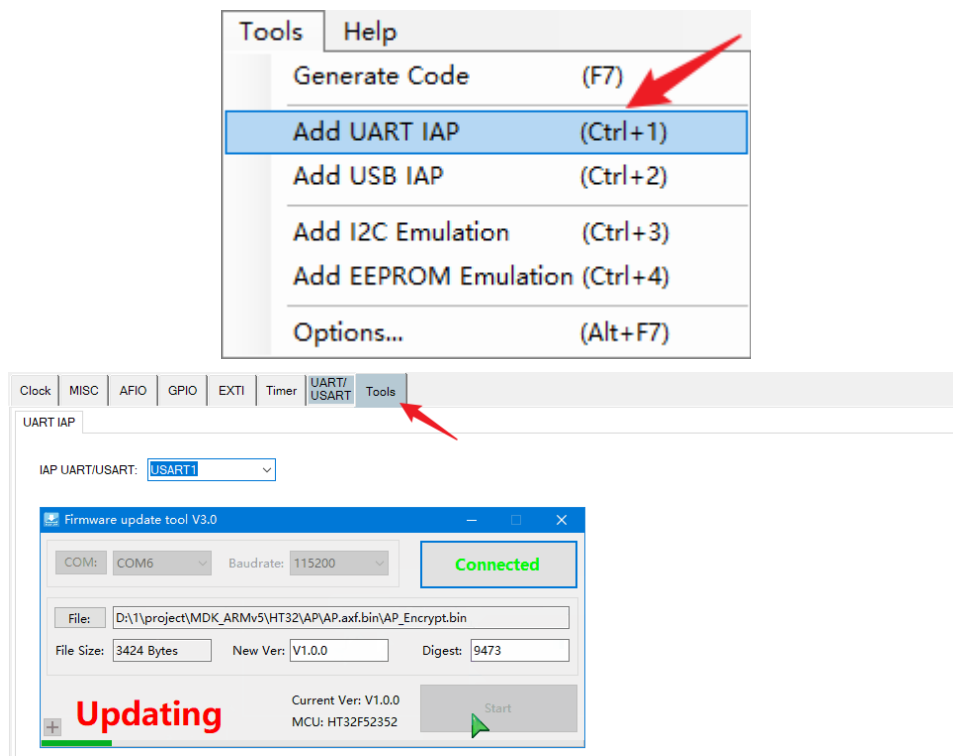
23. Tools 页面的设置

本页面包含有 UART IAP、USB IAP、I²C Emulation、EEPROM Emulation 的相关设置。

菜单栏 Tools 里选择 Add UART IAP、Add USB IAP、Add I²C Emulation、Add EEPROM Emulation 任意一项之后，本页面就会显示出来。

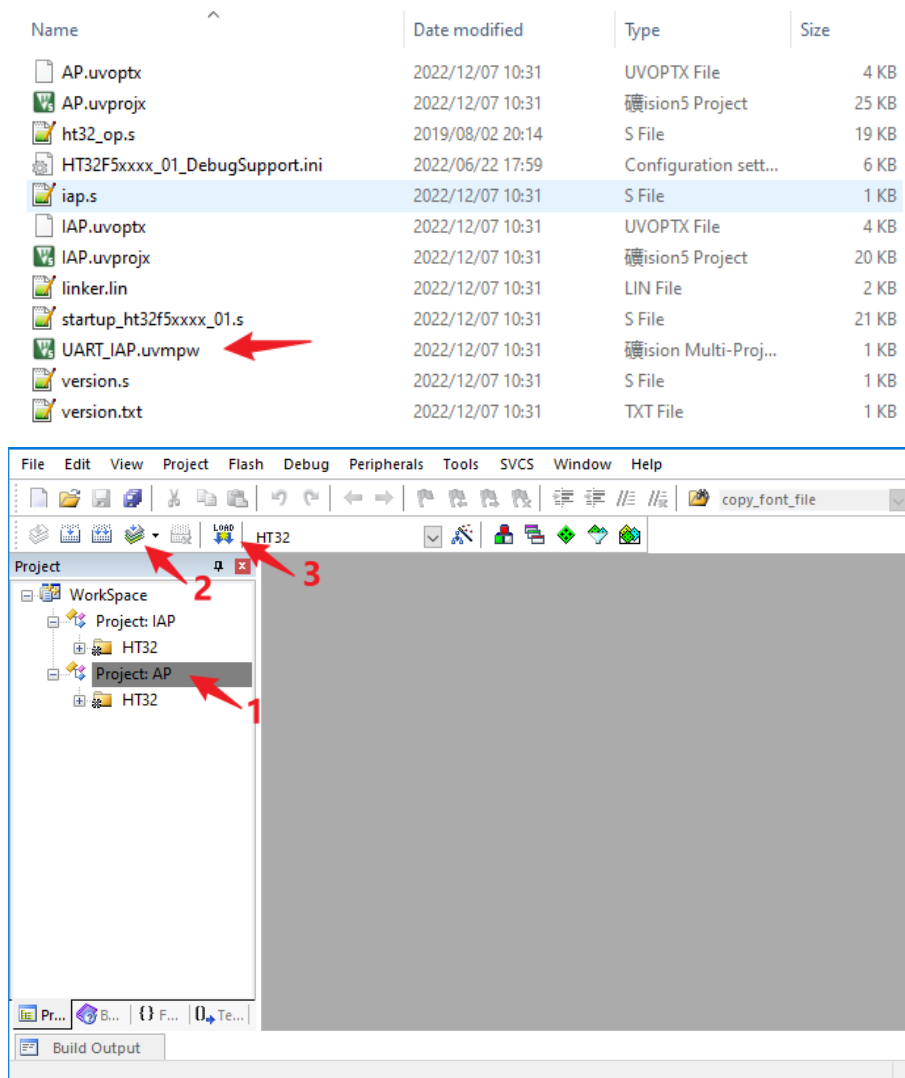
23.1 UART IAP 设置

通过菜单栏 Tools 里选择 Add UART IAP 本页面就会显示出来。注意，需要先从 AFIO 页面激活一组 UART。



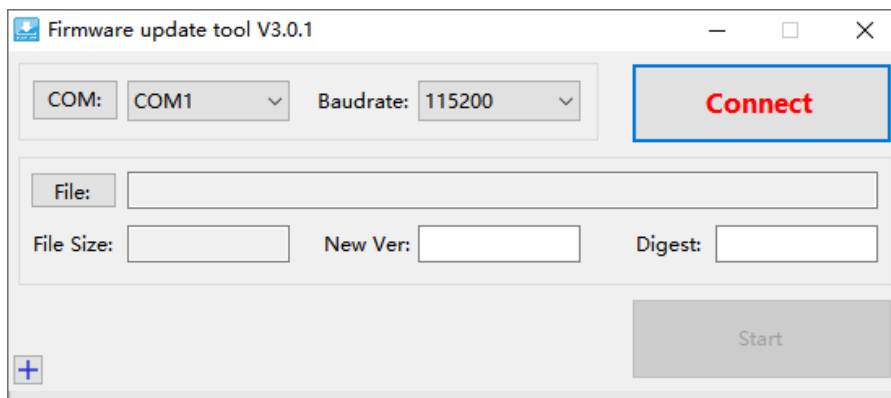
- UART IAP 是一套解决方案，包括上位机程序和固件源程序。
- 先使能一组 UART，就可以通过菜单栏 Tools→Add UART IAP 激活此功能。
- 本页面的设置只有选择哪组 UART，其它都无需设置。

- 详细使用步骤：
 - ◆ 生成的工程包含 IAP、AP 两个工程，点击 Multiproject 可以打开两个工程，如下图：



- ◆ 设置 AP 为 Active Project
- ◆ Batch Build IAP 和 AP 两个工程
- ◆ 下载到 MCU；建议下载之后稍微修改代码，例如点亮一个 LED，然后编译，但不要再通过 MDK 下载到 MCU。用于验证更新后程序功能有变化。
- ◆ 用串口线连接 MCU 到电脑

- ◆ 打开 HT32_UART_IAP_Tool.exe，选择串口号，波特率默认 115200，Connect

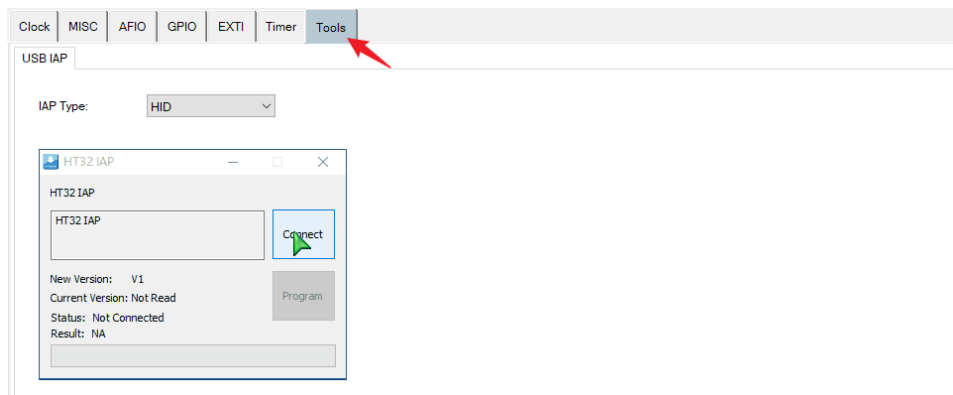


- ◆ 成功连接后会显示当前 MCU 的型号和当前的固件版本号，然后可以选择更新的 bin 文件，bin 文件一般位于 .\project\MDK_ARMv5\HT32\AP\AP.axf.bin 文件夹内。
- ◆ 文件选择正确则会显示其版本号和 checksum，点击 Start 开始下载更新，成功后会提示 Successful。
- ◆ 详细说明请查看 HT32_UART_IAP_Tool.exe 的说明文件。

23.2 USB IAP 设置

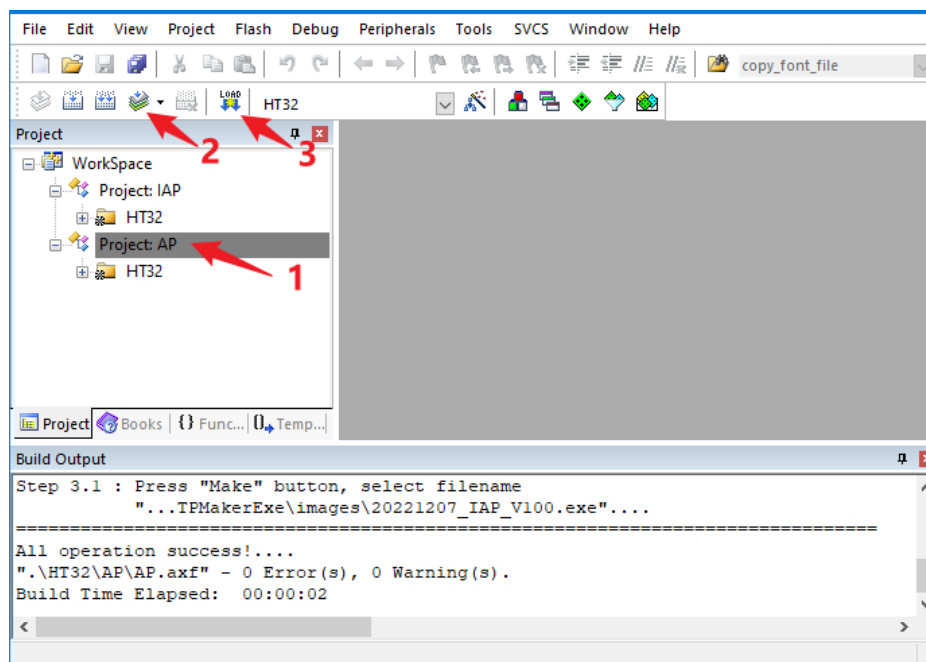
如果 MCU 具有 USB 功能，就可以通过菜单栏 Tools→Add USB IAP 激活此功能。USB IAP 分为三个类型，分别是 HID 类、Mass Storage 类和 DFU 类。下面详细介绍。

23.2.1 USB HID IAP

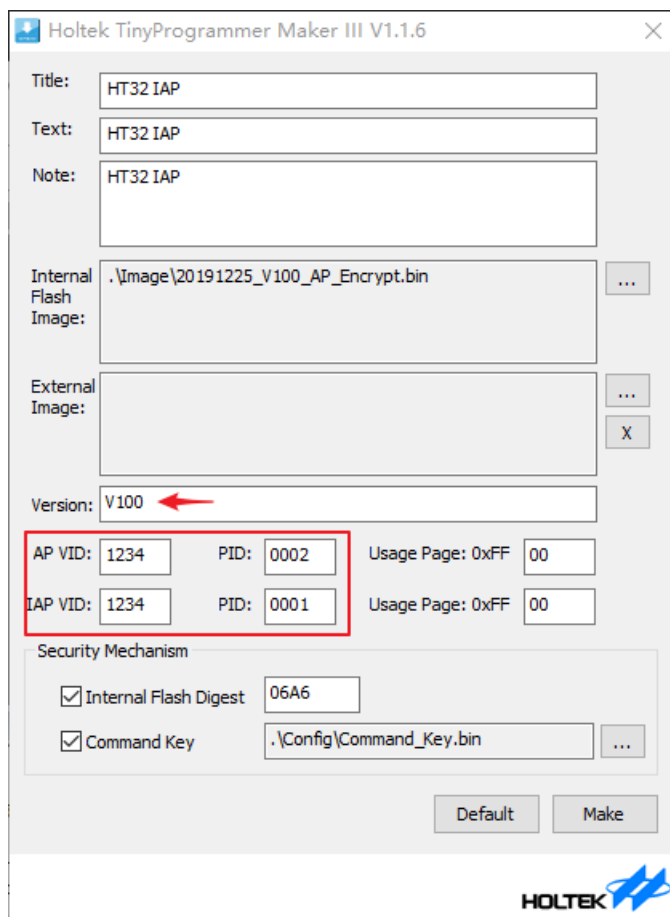


- 激活 USB IAP 功能后，在上图页面 IAP Type 选择 HID；
- 如果有 GPIO、EXTI、TM、SPI、UART、I²C 等其它需要设置的，请一并设置好，按 Build 流程输出工程。
- 生成的工程包含 IAP、AP 两个工程，点击 Multiproject 可以打开两个工程，如下图：

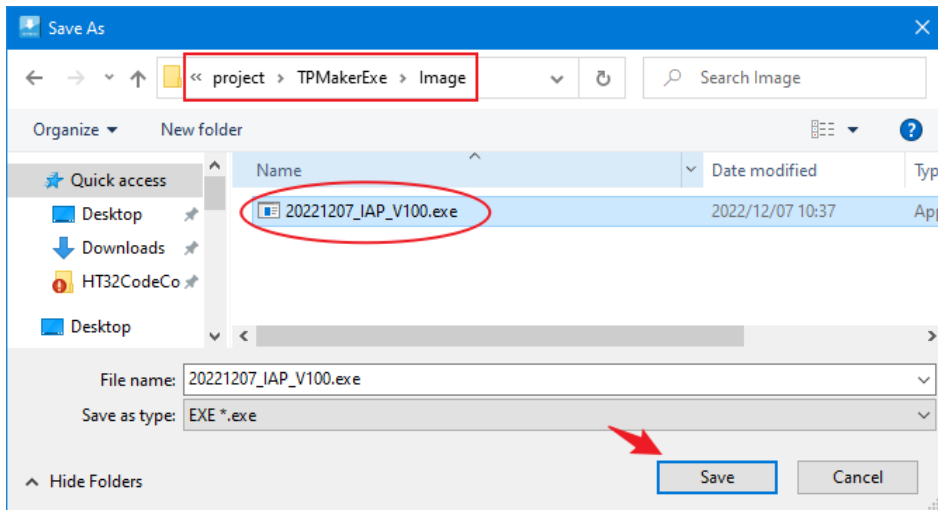
Name	Date modified	Type	Size
AP.uvoptx	2022/12/07 10:36	UVOPTX File	4 KB
AP.uvprojx	2022/12/07 10:36	碇ision5 Project	26 KB
ht32_op.s	2019/08/02 20:14	S File	19 KB
HT32F5xxx_01_DebugSupport.ini	2022/06/22 17:59	Configuration sett...	6 KB
iap.s	2022/12/07 10:36	S File	1 KB
IAP.uvoptx	2022/12/07 10:36	UVOPTX File	4 KB
IAP.uvprojx	2022/12/07 10:36	碇ision5 Project	23 KB
linker.lin	2022/12/07 10:36	LIN File	2 KB
startup_ht32f5xxx_01.s	2022/12/07 10:36	S File	21 KB
USB_HID_IAP.uvmpw	2022/12/07 10:36	碇ision Multi-Proj...	1 KB
USBBufCheck_Lib_M0p_Keil.lib	2020/09/15 16:03	Object File Library	61 KB
version.s	2022/12/07 10:36	S File	1 KB
version.txt	2022/12/07 10:36	TXT File	1 KB



- 设置 AP 为 Active Project
- Batch Build IAP 和 AP 两个工程，工程编译结束时会弹出 TinyProgrammer Maker，输入版本号，点击 Make 按钮 (VID/PID 的值，是根据固件的源代码，在 .\project\TPMakerExe\Config\make_config.bat 中设定的，如果修改请与固件源代码对应位置一起修改)

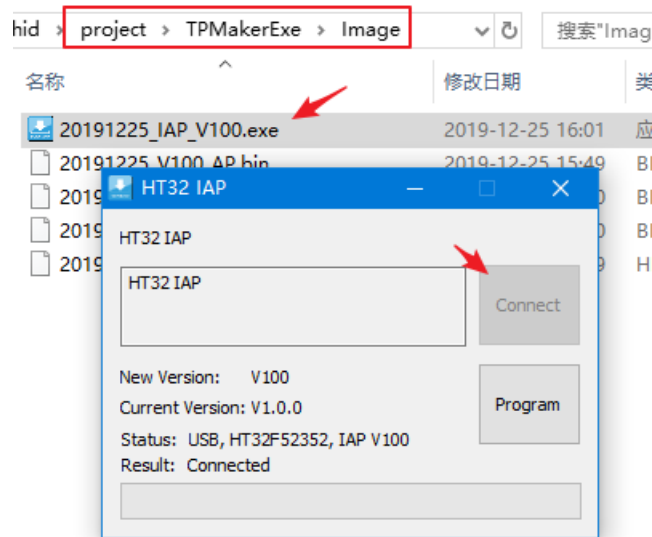


- 选择如下路径，指定已经按日期和版本号取好名字的文件，覆盖保存

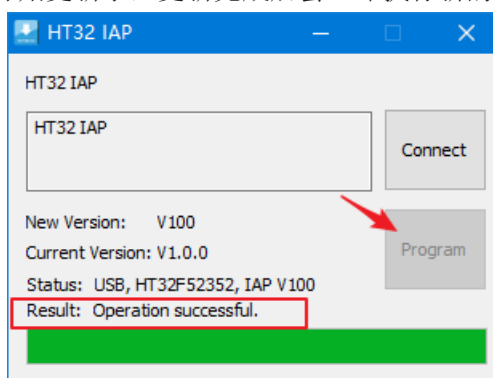


- 下载到 MCU；建议下载之后稍微修改代码，例如点亮一个 LED，然后编译，按上面的步骤重新制作一个 exe 文件，但不要再通过 MDK 下载到 MCU。用于验证更新后程序功能有变化。
- 用 USB 线连接 MCU 到电脑

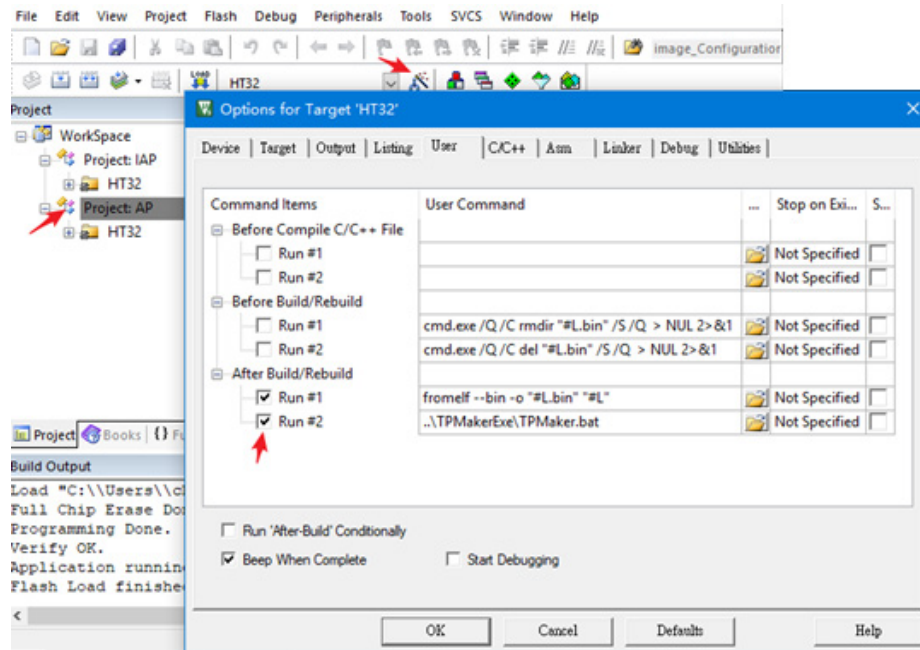
- 打开刚才生成的 exe 文件，点击 Connect，连接成功后会显示芯片的型号，和当前芯片内的固件版本号，以及本次更新的版本号，如下图。



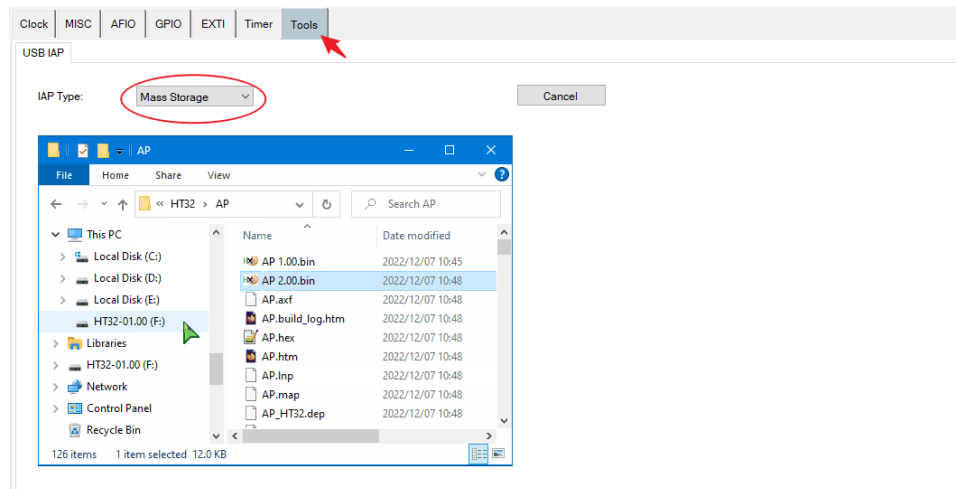
- 点击 Program 就开始更新了，更新完成后会立即执行新的固件



- 如果不希望每次编译 AP 工程都进行升级文件的制作流程，请于 AP 的 Option → User 页面取消 Run #2 的选择，如下图：

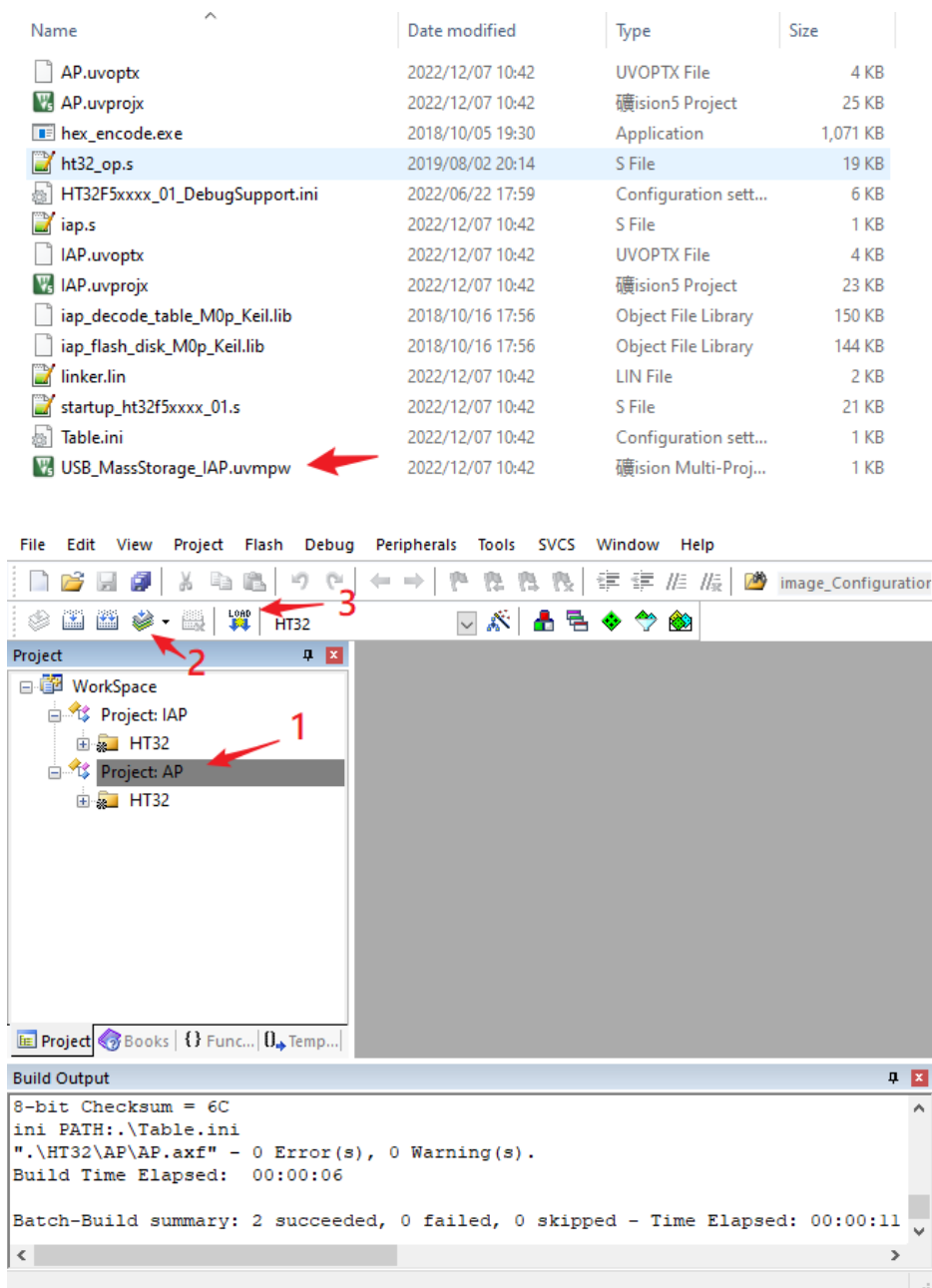


23.2.2 USB Mass Storage IAP



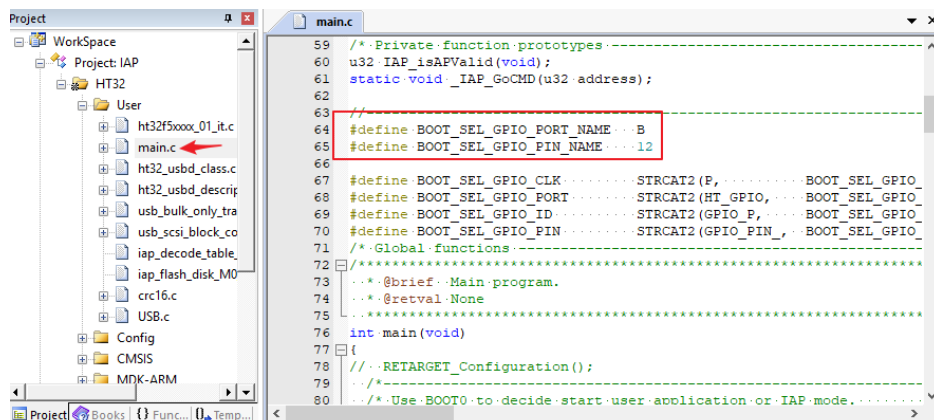
- 激活 USB IAP 功能后，在上图页面 IAP Type 选择 Mass Storage；
- 如果有 GPIO、EXTI、TM、SPI、UART、I²C 等其它需要设置的，请一并设置好，按 Build 流程输出工程。

- 生成的工程包含 IAP、AP 两个工程，点击 Multiproject 可以打开两个工程，如下图：

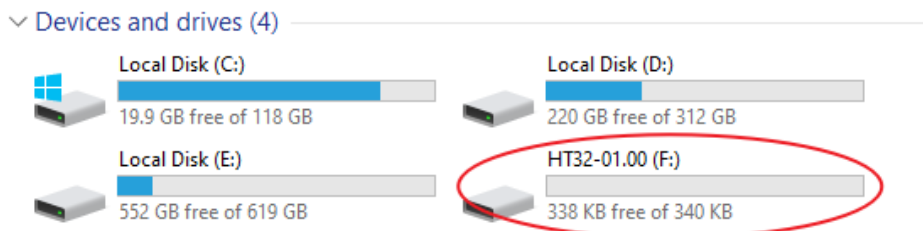


- 设置 AP 为 Active Project
- Batch Build IAP 和 AP 两个工程
- 下载到 MCU；建议下载之后稍微修改代码，例如修改磁盘名称，增加一个 LED 功能，然后编译但不要再通过 MDK 下载到 MCU。用于验证更新后程序功能的变化。
- 用 USB 线连接 MCU 到电脑

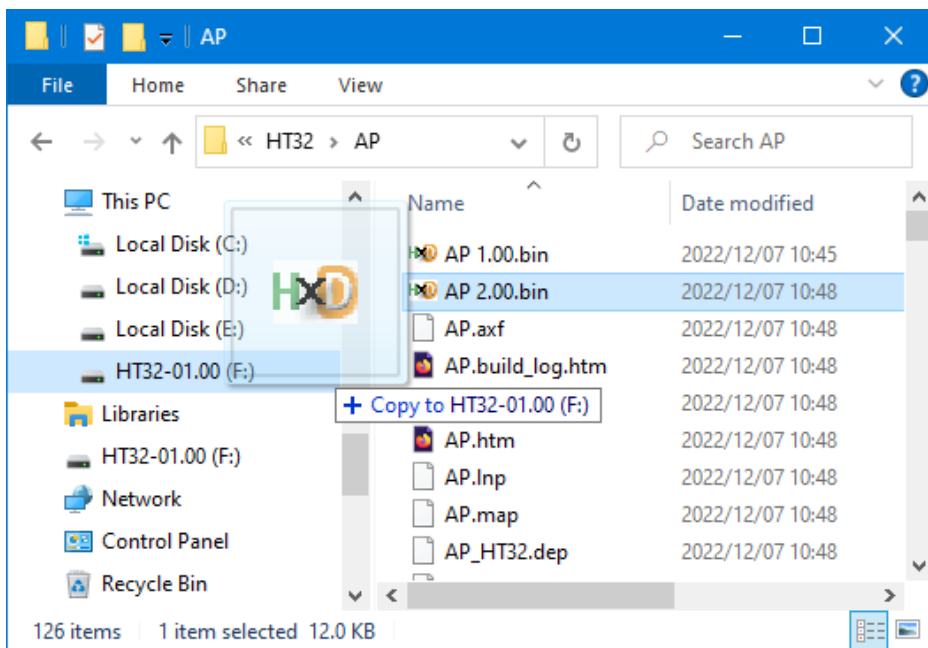
- IAP 工程里的默认设定是 PB12 拉低则程序停留在 IAP 里，也可以修改 AP 程序，通过某种触发条件，例如按某个按键，进入 IAP。这个部分太灵活，此处不深入讨论。按照 PB12 拉低复位一次使程序进入 IAP。



- 进入 IAP 程序之后，电脑会弹出一个可移动磁盘，如下图。



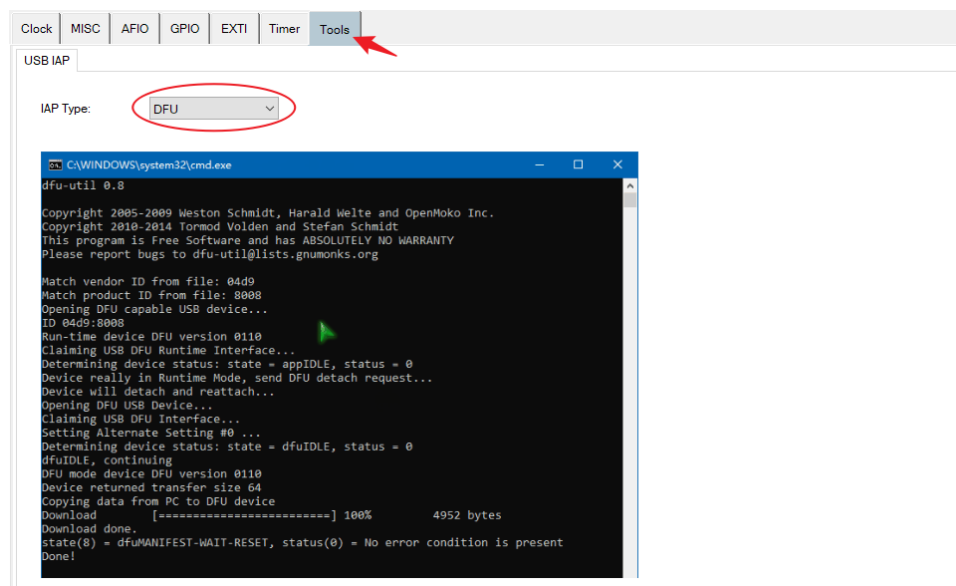
- 把需要更新的 bin 文件 (位于 .\project\MDK_ARMv5\HT32\AP\AP.bin)，拖到弹出的可移动磁盘里，更新就完成了。



- 保持 PB12 拉高，手动复位一次，开始执行新的固件。

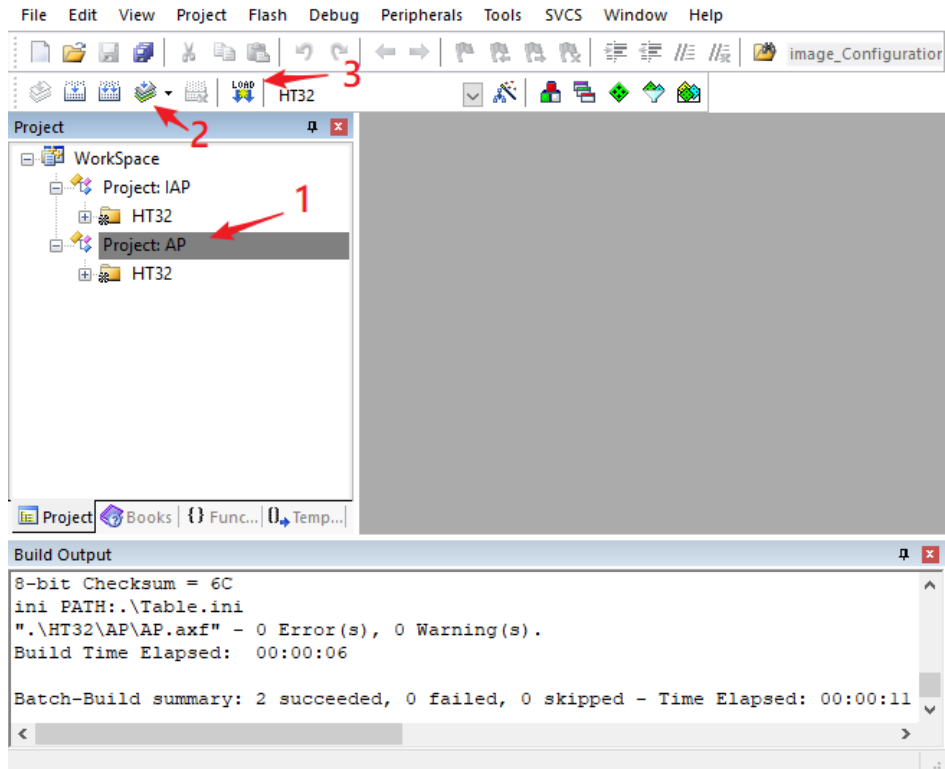
- 文件 .\project\MDK_ARMv5\Table.ini 是加密表，若不希望他人破解用于更新的 bin 文件，请自行修改 Table.ini 的数据，注意 IAP 的 main.c 里 Table1[139] 需要与此文件的数据一致。
- 这种更新方式可用在 Windows、MAC、Linux 各种操作系统上。

23.2.3 USB DFU

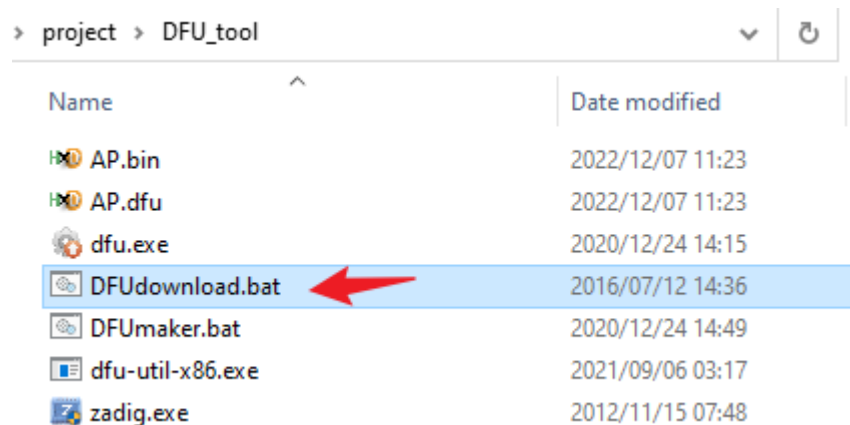


- 激活 USB IAP 功能后，在上图页面 IAP Type 选择 DFU；
- 如果有 GPIO、EXTI、TM、SPI、UART、I2C 等其它需要设置的，请一并设置好，按 Build 流程输出工程。
- 生成的工程包含 IAP、AP 两个工程，点击 Multiproject 可以打开两个工程，如下图：

project > MDK_ARMv5			
Name	Date modified	Type	Size
AP.uvoptx	2022/12/07 11:22	UVOPTX File	
AP.uvprojx	2022/12/07 11:22	MDKvision5 Project	
ht32_op.s	2019/08/02 20:14	S File	
HT32F5xxxx_01_DebugSupport.ini	2022/06/22 17:59	Configuration sett...	
iap.s	2022/12/07 11:22	S File	
IAP.uvoptx	2022/12/07 11:22	UVOPTX File	
IAP.uvprojx	2022/12/07 11:22	MDKvision5 Project	
linker.lin	2022/12/07 11:22	LIN File	
startup_ht32f5xxxx_01.s	2022/12/07 11:22	S File	
USB_DFU.uvmpw	2022/12/07 11:22	MDKvision Multi-Proj...	



- 设置 AP 为 Active Project
- Batch Build IAP 和 AP 两个工程
- 下载到 MCU；建议下载之后稍微修改代码，例如点亮一个 LED，然后编译，按上面的步骤重新制作一个 exe 文件，但不要再通过 MDK 下载到 MCU。用于验证更新后程序功能有变化。
- 用 USB 线连接 MCU 到电脑
- Keil MDK 正常编译后，会在以下路径产出 DFU 文件以及下载 DFU 的批处理文件 DFUdownload.bat。



- 点击运行 DFUdownload.bat，就可以进行 DFU 更新。

```
C:\WINDOWS\system32\cmd.exe

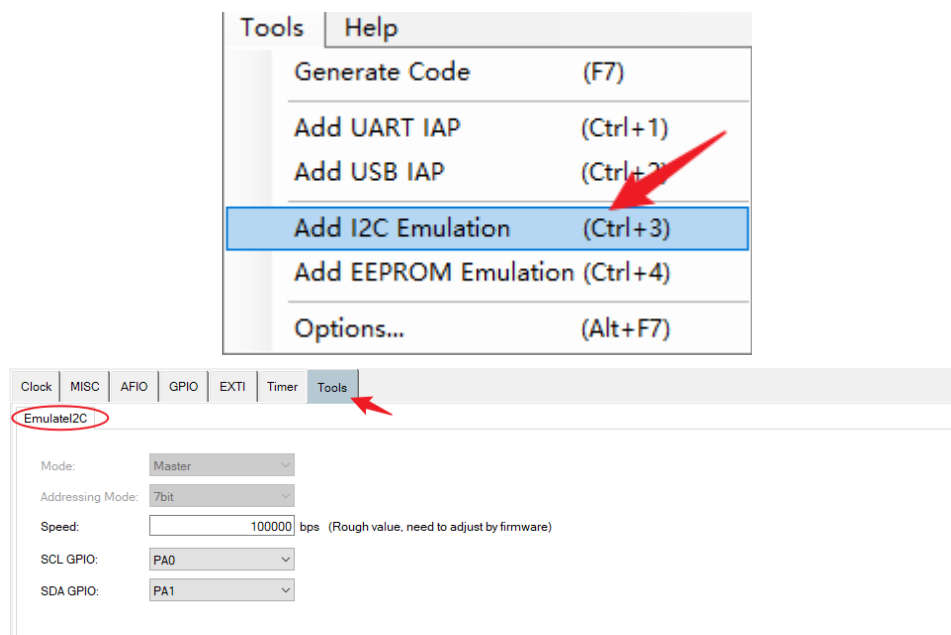
C:\Users\chen\OneDrive\Desktop\2\project\DFU_tool>dfu-util-x86 -D AP.dfu
dfu-util 0.8

Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2014 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to dfu-util@lists.gnumonks.org

Match vendor ID from file: 04d9
Match product ID from file: 8008
Opening DFU capable USB device...
ID 04d9:8008
Run-time device DFU version 0110
Claiming USB DFU Runtime Interface...
Determining device status: state = appIDLE, status = 0
Device really in Runtime Mode, send DFU detach request...
Device will detach and reattach...
Opening DFU USB Device...
Claiming USB DFU Interface...
Setting Alternate Setting #0 ...
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 0110
Device returned transfer size 64
Copying data from PC to DFU device
Download [=====] 100% 4756 bytes
Download done.
state(8) = dfuMANIFEST-WAIT-RESET, status(0) = No error condition is present
Done!
```

23.3 I²C Emulation

通过菜单栏 Tools→Add I²C Emulation 可以激活软件模拟 I²C 功能。



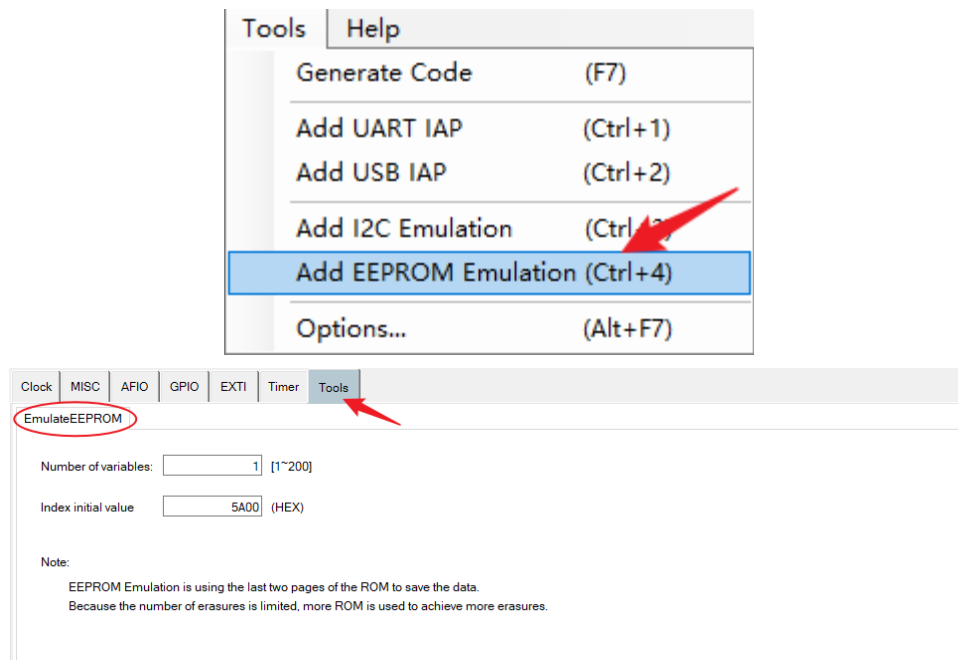
软件模拟的 I²C 只能是 Master 模式，可以任意设置使用两个 GPIO 作为 I²C 接口。当硬件 I²C 不够时，软件模拟 I²C 也是很方便，只是软件模拟的 I²C 通信时会一直占用 CPU。

如果需要多组软件模拟的 I²C，请先生成一组然后自行复制文件、修改文件再添加到工程内。

23.4 EEPROM Emulation

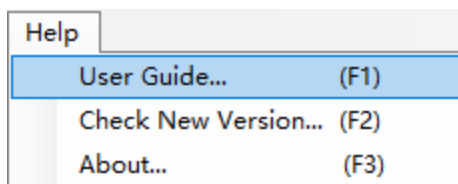
MCU 的内部 Flash 可以当成 EEPROM 用来保存用户数据，断电也不会丢失。但是内部 Flash 擦写寿命是 10K 次，为了延长擦写次数，可以用内部 Flash 的多个页面交替保存数据以达到更多擦写次数的目的。例如用户有 2 个字节需要断电保存，如果只写在固定的位置那只有 10K 次擦写；如果用最后两页保存，按每页 1024 字节，可以擦写 $1024 \times 2 \div 4 \times 10K = 5120K$ 次，即用更多的 ROM 空间换取更多的擦写次数。

通过菜单栏 Tools→Add EEPROM Emulation 激活此功能。

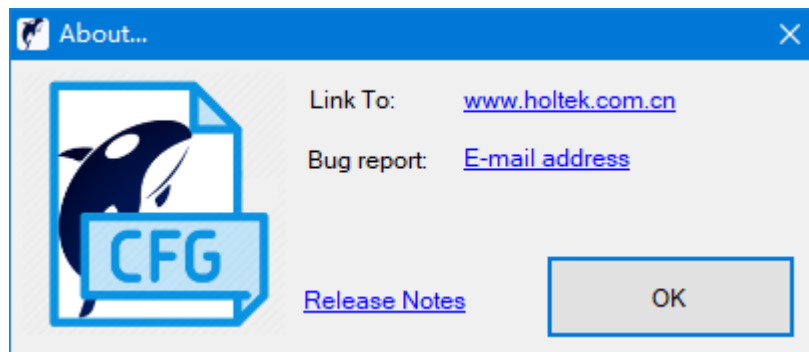


- 可以设置要保存的变量数目，范围 1~200。
- 可以设置数据索引号的初始值，不能设置为 0 或者 FFFF。

24. Help 页面



- User Guide: 打开本使用手册。
- Check New Version: 检查在线新版本，如果有新版本请按提示下载和更新。
- About: 打开官网链接或问题反馈链接。



Copyright© 2023 by HOLTEK SEMICONDUCTOR INC. All Rights Reserved.

本文件出版时 HOLTEK 已针对所载信息为合理注意，但不保证信息准确无误。文中提到的信息仅是提供作为参考，且可能被更新取代。HOLTEK 不担保任何明示、默示或法定的，包括但不限于适合商品化、令人满意的质量、规格、特性、功能与特定用途、不侵害第三方权利等保证责任。HOLTEK 就文中提到的信息及该信息之应用，不承担任何法律责任。此外，HOLTEK 并不推荐将 HOLTEK 的产品使用在会由于故障或其他原因而可能会对人身安全造成危害的地方。HOLTEK 特此声明，不授权将产品使用于救生、维生或安全关键零部件。在救生 / 维生或安全应用中使用 HOLTEK 产品的风险完全由买方承担，如因该等使用导致 HOLTEK 遭受损害、索赔、诉讼或产生费用，买方同意出面进行辩护、赔偿并使 HOLTEK 免受损害。HOLTEK (及其授权方，如适用) 拥有本文件所提供信息 (包括但不限于内容、数据、示例、材料、图形、商标) 的知识产权，且该信息受著作权法和其他知识产权法的保护。HOLTEK 在此并未明示或暗示授予任何知识产权。HOLTEK 拥有不事先通知而修改本文件所载信息的权利。如欲取得最新的信息，请与我们联系。