

## AT32 Work Bench 用户手册

## 前言

本手册用于介绍 AT32 Work Bench 使用。AT32 Work Bench 通过对 MCU 的图形化配置，生成初始化 C 代码和对应 IDE 项目，以减少开发人员的工作量、时间和成本。

AT32 Work Bench 有以下主要特点：

1. 支持外设初始化配置
2. 支持 PIN MUX 配置，并可自定义 PIN 脚标签
3. 支持系统时钟自动化配置
4. 支持在线代码预览，所见即所得
5. 支持 add user code 功能（客户已有代码不会被新生成工程覆盖）
6. 支持 Keil、IAR、AT32 IDE 等常见 IDE 工程自动生成
7. 支持最近编辑的设计记录
8. 支持配置 PDF 报告的生成
9. 支持中英文切换
10. 支持 Windows、Linux 等多平台
11. 支持软件的在线升级，以及 MCU 固件的在线下载

## 目录

<b>1</b>	<b>引言 .....</b>	<b>5</b>
1.1	环境要求 .....	5
<b>2</b>	<b>安装 .....</b>	<b>6</b>
2.1	Windows 系统安装 .....	6
2.2	Linux 系统安装 .....	6
<b>3</b>	<b>引导页 .....</b>	<b>7</b>
<b>4</b>	<b>项目配置 .....</b>	<b>8</b>
4.1	菜单栏与工具栏 .....	8
4.1.1	菜单栏 .....	8
4.1.2	工具栏 .....	9
4.2	Pin 布局和配置 .....	9
4.2.1	外设 .....	9
4.2.2	模式和配置 .....	9
4.2.3	Pin 布局 .....	13
4.2.4	GPIO 配置 .....	15
4.2.5	DMA 配置 .....	16
4.2.6	NVIC 配置 .....	17
4.3	时钟配置 .....	18
4.4	代码预览 .....	19
4.5	生成代码 .....	20
4.5.1	生成代码 .....	20
4.5.2	重新生成代码时保留用户代码 .....	23
4.6	固件包管理 .....	24
<b>5</b>	<b>文档版本历史 .....</b>	<b>25</b>

## 表目录

表 1. 外设状态 .....	9
表 2. 模式与参数显示状态 .....	11
表 3. 引脚状态显示 .....	13
表 4. 文档版本历史 .....	25

## 图目录

图 1. 终端命令安装 .....	6
图 2. 图形方式安装 .....	6
图 3. 引导页 .....	7
图 4. 项目配置 .....	8
图 5. 菜单与工具栏 .....	8
图 6. 外设的模式和配置 .....	10
图 7. 外设模式 .....	10
图 8. 模式错误提示 .....	10
图 9. 外设参数设置 .....	11
图 10. 外设 GPIO 设置 .....	11
图 11. 外设 DMA 设置 .....	12
图 12. 外设 NVIC 设置 .....	12
图 13. PIN 布局 .....	13
图 14. 引脚配置菜单 .....	14
图 15. 自定义标签 .....	14
图 16. GPIO 配置 .....	15
图 17. DMA 配置 .....	16
图 18. NVIC 配置 .....	17
图 19. 时钟配置 .....	18
图 20. 代码预览 .....	19
图 21. 生成代码 .....	20
图 22. 项目文件目录 .....	21
图 23. MDK_V5 项目文件 .....	22
图 24. 固件包管理 .....	24

# 1 引言

## 1.1 环境要求

### ■ 软件要求

#### Windows 系统

需要 Windows 7 及以上操作系统支持。

#### Linux 系统

支持 x86\_64 架构的 Ubuntu、Fedora 等发行版。

### ■ 硬件要求

建议内存最低 2GB。

建议硬盘最低 4GB

## 2 安装

### 2.1 Windows 系统安装

本软件不需要安装，只需直接运行可执行程序 AT32\_Work\_Bench.exe。

### 2.2 Linux 系统安装

本软件目前支持 Ubuntu 16.4 以上系统版本。

Linux 系统安装可以有两种方式：终端中使用 dpkg 命令安装和图形方式安装。

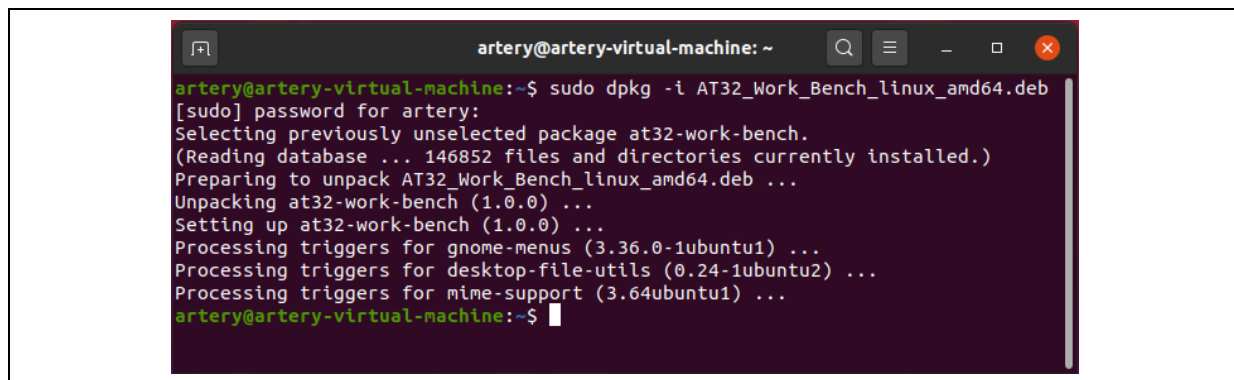
#### ■ 终端中使用 dpkg 命令安装

在终端中输入如下命令进行安装：

```
sudo dpkg -i AT32_Work_Bench_Linux-x86_64_Vx.x.xx.deb
```

如下图所示

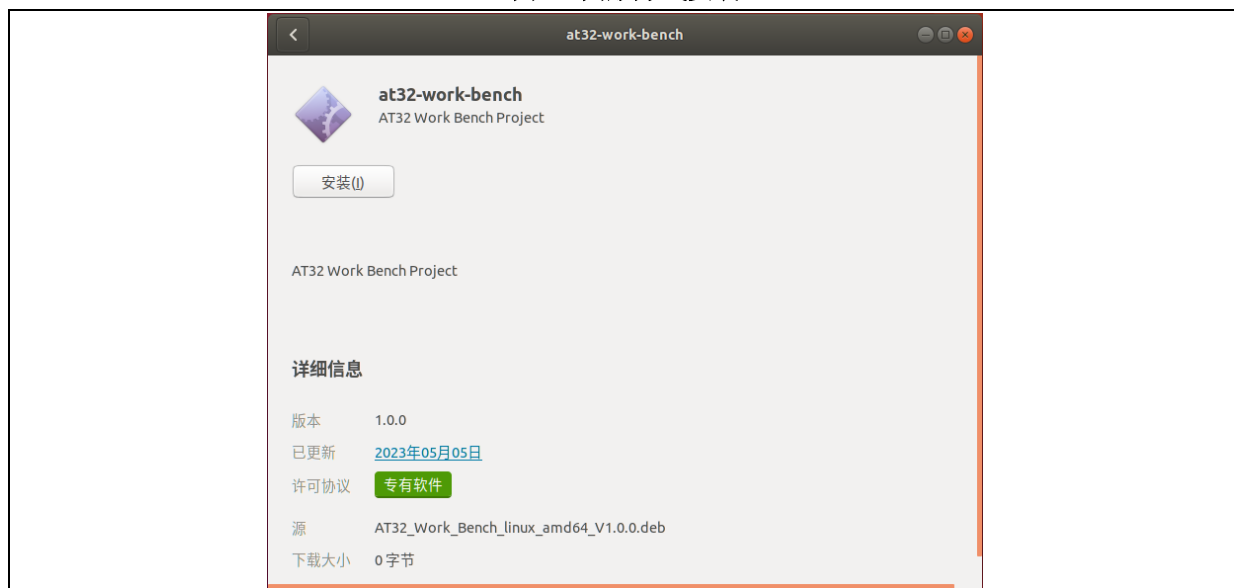
图 1. 终端命令安装



#### ■ 图形方式安装

拷贝 AT32\_Work\_Bench\_Linux-x86\_64\_Vx.x.xx.deb 到 Linux 系统中，并双击。在弹出的安装程序中，点击“安装”按钮，系统将本软件自动安装到系统中。

图 2. 图形方式安装



安装完成之后，点击左边任务栏最下方“显示所有程序”按钮，在弹出的程序列表中找到“AT32\_Work\_Bench”，并点击，将启动 AT32\_Work\_Bench。

### 3 引导页

引导页是启动 AT32 Work Bench 时打开的第一个窗口。它提供三个选项：“开始一个新的设计”、“打开一个已有的设计”和“最近使用的设计”。

#### ■ 开始一个新的设计

选择 MCU 产品线、MCU 系列、MCU 型号，封装、Flash 与 SRAM 将根据型号自动填写；然后点击“新建”，将创建一个新的项目；

#### ■ 打开一个已有的设计

可以通过资源管理器找到一个已经保存的项目，并打开该项目 (\*.ATWP 文件)；

#### ■ 最近使用的

展示了最近创建的项目列表，可选择一个项目打开。

图 3. 引导页

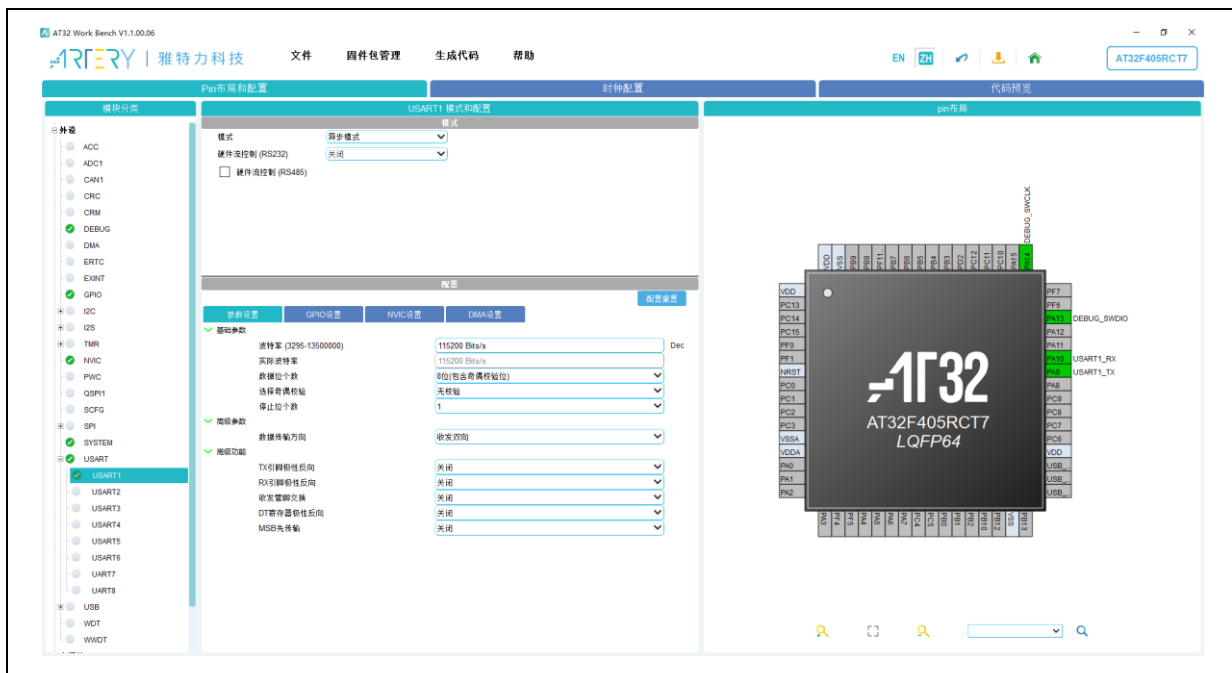


## 4 项目配置

创建新项目后，或者已经加载了之前保存的项目后，将打开项目配置页面。项目配置页面包含菜单栏、工具栏与以下一组视图：

- Pin 布局 and 配置
- 时钟配置
- 代码预览


图 4. 项目配置



在“Pin 布局 and 配置”中，可以对外设使用的引脚和参数进行配置。

在“时钟配置”中，可以对 MCU 时钟进行配置。

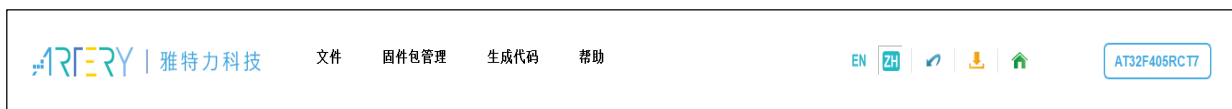
在“代码预览”中，可以预览根据当前配置自动生成的代码。

点击菜单栏的生成代码功能或者工具栏的  图标，可以生成与当前配置相对应的用户代码和项目。

### 4.1 菜单栏与工具栏

菜单栏与工具栏内容如图所示：

图 5. 菜单与工具栏



#### 4.1.1 菜单栏

##### ■ “文件” 菜单：

新建设计：

打开设计：

保存设计：

设计另存为：

生成报告：

新建一个项目。

打开已存在的项目。

保存当前项目。


将当前项目另存为。


生成当前项目 pdf 格式报告。





- 最近的设计：展示最近编辑的项目。
- “固件包管理” 菜单：  
固件包管理：安装和管理支持的固件包（BSP）。
  - “生成代码” 菜单：  
当在对应型号的操作配置界面将所期望的外设参数与时钟配置完成之后，可点击“生成代码”菜单来选择项目的存储路径并生成相应的项目文件。
  - “帮助” 菜单：  
打开手册：将打开软件的使用手册。  
新版本下载：联网进行新版本下载。  
访问雅特力官网：打开浏览器访问雅特力官网。  
版本：查看当前版本。


## 4.1.2 工具栏


 EN：选择 English 作为显示语言。

 ZH：选择简体中文作为显示语言。

：所有配置重置，将会对所有外设配置的参数、引脚、时钟配置等进行重置。

：生成代码，将打开“项目管理”窗口，然后产生代码。具体[参照节 4.5](#)。

：访问雅特力官网。

 AT32F421C8T7：显示当前正在进行配置的 MCU 型号。

## 4.2 Pin 布局和配置




“Pin 布局和配置”视图主要配置 MCU 外设使用的引脚及参数。

该视图包含有“外设”、“模式和配置”与“Pin 布局”三个窗口。每个窗口的大小可以随意调整：将鼠标悬停到窗口边框上会显示一个双向箭头，按住鼠标左键并左右移动来扩展或缩小窗口区域大小。

### 4.2.1 外设

“外设”窗口展示了所选 MCU 型号支持的外设和中间件列表。外设名称前面的图标显示了当前外设的配置状态。

表 1. 外设状态

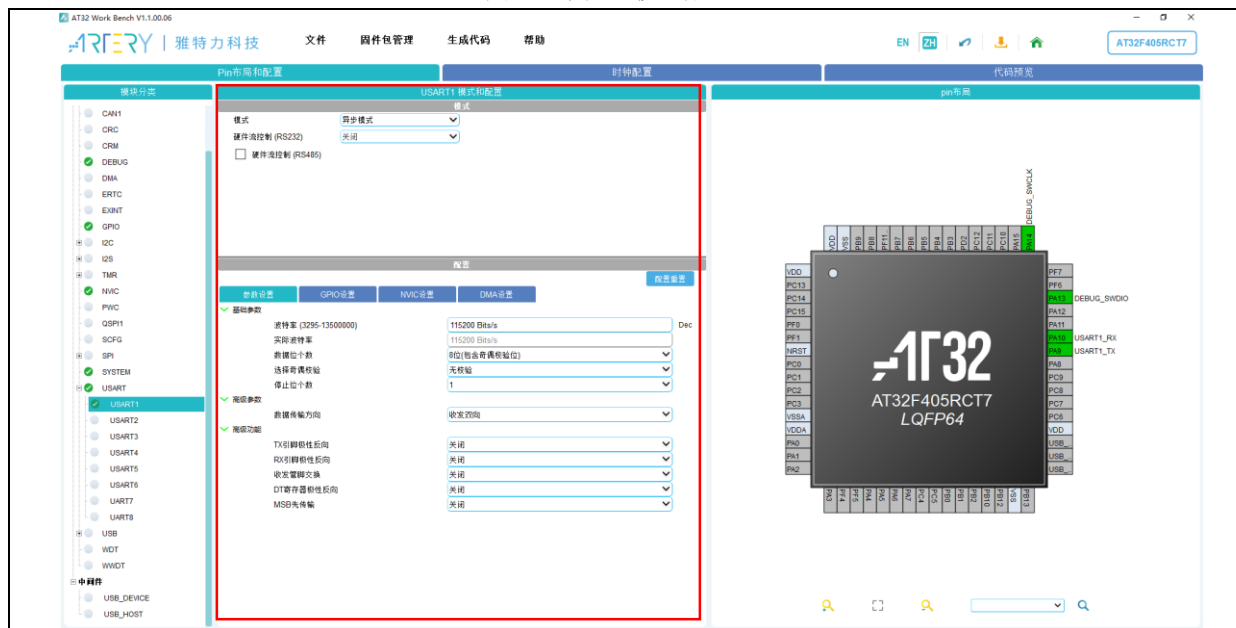
图标	状态	备注
	未配置	外设没有配置模式和参数。
	配置成功	外设已成功配置模式和参数
	配置参数有误	外设配置的模式或参数有误，请检查后重新配置。

### 4.2.2 模式和配置

从“外设”列表中选择一个外设时，将打开所选外设的“模式和配置”窗口，用户可以借此配置其功

能模式与参数。模式窗口可选择外设的工作模式和工作模式下使用的 MCU 引脚。由于 MCU 允许不同外设和多个功能使用相同的引脚，因此选择模式后，会自动配置最适合用户选择的外设的引脚排列。

图 6. 外设的模式和配置



以外设“USART1”为例进行说明：

#### ■ 外设模式

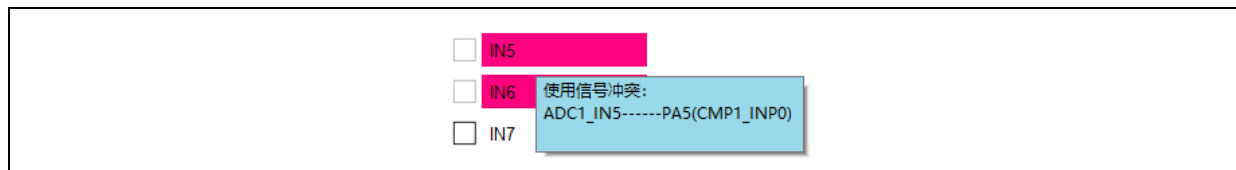
图 7. 外设模式



如上图所示，外设“USART1”的模式选择“异步模式”时，引脚布局的视图中，PA9 与 PA10 将自动映射到信号“USART1\_TX”与“USART1\_RX”；硬件流控制选择“同时 CTS/RTS”，PA11 与 PA12 将自动映射到功能“USART1\_CTS”与“USART1\_RTS”。

模式在条件不满足或者信号冲突的情况，将以不同的颜色或背景色进行显示，如表 2 所示。鼠标移动到该控件时，会弹出提示信息，用户可根据提示信息进行修改或重新配置，如下图所示。

图 8. 模式错误提示



模式与参数状态举例说明如下：

表 2. 模式与参数显示状态

图标	状态	备注
硬件流控制 <span>同时CTS/RTS</span>	可配置	
<input type="checkbox"/> 外部输出	条件不满足	需要根据提示配置其他参数。
请选择一个 反相输入选择 (Input [-])	提示需要配置对应的模式	需要选择“反相输入选择 (Input [-])”。
<input type="checkbox"/> IN5	使用信号冲突	与其他功能信号冲突。
<div><div>参数设置</div><div>GPIO设置</div><div>NVIC设置</div><div>DMA设置</div></div> <div>波特率 (3-14548) <span>115200 Baud</span></div>	参数值错误	参数值超过最大最小值范围

■ 外设参数设置

图 9. 外设参数设置

参数设置

GPIO设置

NVIC设置

DMA设置

✓ 基础参数

波特率 (1831-7500000)

115200 Bits/s

数据位个数

8位(包含奇偶校验位)

选择奇偶校验

无校验

停止位个数

1

✓ 高级参数

数据传输方向

收发双向

收发管脚交换

关闭

参数配置窗口显示了“USART1”可以配置的参数，如基础参数中的“波特率”、“数据位个数”等，高级参数中的“数据传输方向”、“收发管脚交换”等参数。

配置参数的时候，如果检测到无效设置，将会根据数据范围自动进行优化。如果用户设定的值分别小于/大于最小/最大值，则重置为最小/最大有效值。

■ 外设 GPIO 设置

图 10. 外设 GPIO 设置

参数设置

GPIO设置

NVIC设置

DMA设置

Pin名称	信号	输出电平	GPIO类型	上/下拉类型	GPIO模式	驱动能力	标签	已修改
PF0	USART1_RX	n/a	推挽	无上/下拉	复用功能模式	适中		N
PF1	USART1_TX	n/a	推挽	无上/下拉	复用功能模式	适中		N

外设 GPIO 设置窗口显示了当前外设可以配置的 GPIO，如上图可配置“USART1”中“USART1\_TX”和“USART1\_RX”的 GPIO 参数。

“GPIO 参数”的具体配置说明可以[参照节 4.2.4](#)。

#### ■ 外设 DMA 设置

图 11. 外设 DMA 设置

参数设置	GPIO设置	NVIC设置	DMA设置
DMA请求	通道	传输方向	优先级
USART1_RX	DMA1通道3	外设到内存	低
USART1_TX	DMA1通道2	内存到外设	低

添加删除

DMA请求参数

模式	普通
外设地址递增	外设地址递增关闭
内存地址递增	内存地址递增开启
外设数据宽度	字节
内存数据宽度	字节

外设 DMA 设置窗口可添加当前外设可以配置的 DMA 请求，如上图可配置“USART1”中“USART1\_TX”和“USART1\_RX”的对应的 DMA 通道以及 DMA 请求的参数。

“DMA 请求参数”的具体配置说明可以[参照节 4.2.5](#)。

#### ■ 外设 NVIC 设置

图 12. 外设 NVIC 设置

参数设置	GPIO设置	NVIC设置	DMA设置
中断表	开启	抢占优先级	子优先级
DMA1_Channel3_2_IRQ	<input type="checkbox"/>	0	0
USART1_IRQ	<input type="checkbox"/>	0	0

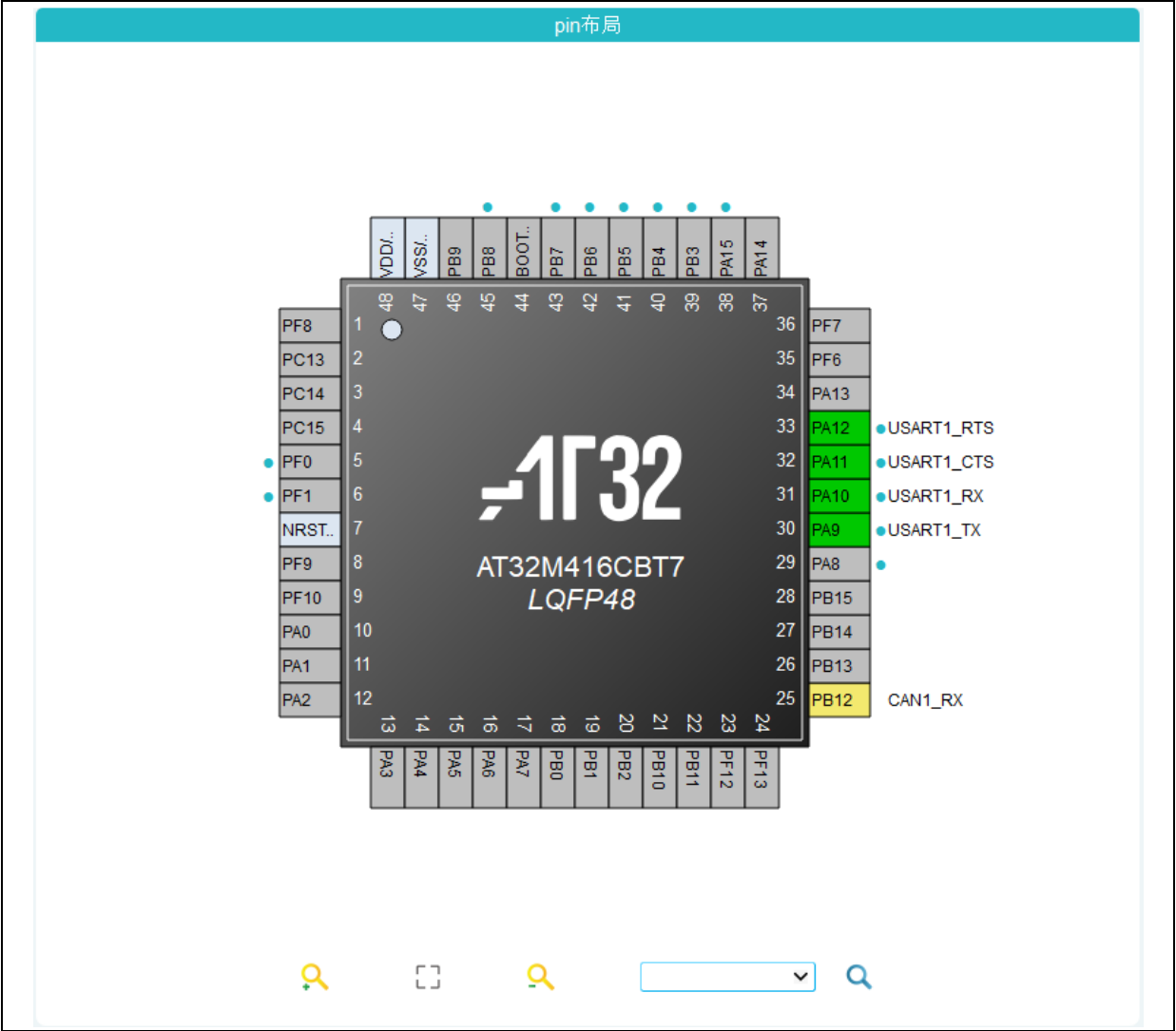
外设 NVIC 设置窗口显示了当前外设可以配置的中断，当外设对应的 DMA 通道开启后，也可配置对应 DMA 通道的中断，如上图可配置“USART1”中“USART1\_IRQ”和“DMA1\_Channel3\_2\_IRQ”。

“抢占优先级”和“子优先级”的配置需到“NVIC”总配置窗口进行统一配置，[参照节 4.2.6](#)。

4.2.3 Pin 布局

以图形方式表示所选封装的引脚布局（例如，LQFP48、QFN32、TSSOP20 等），其中每个引脚都用其名称（如 PA9）、配置状态和当前信号分配来表示。

图 13. PIN 布局



引脚根据不同的状态显示不同的颜色与功能信息，如表 3 所示。

表 3. 引脚状态显示


图标	状态	备注
	引脚未使用	可以对该引脚配置信号。
	引脚已使用	已经配置了有效的信号，信号有对应的外设模式。
	引脚无有效模式	配置了信号，但是信号无对应的外设模式。
	固定模式	固定配置，不可以分配其他功能。
	搜索到的引脚	使用下方搜索功能时，搜索到的引脚将蓝色闪烁显示。
	外设引脚标注	点击左边外设列表中的外设，可标注出该外设所有引脚位置。

Pin 布局窗口支持 Ctrl + 鼠标滚轮放大和缩小 MCU 引脚视图，并支持鼠标拖拽调整位置。


点击图示芯片内部区域，可显示或隐藏引脚号。


通过图形下方的工具栏，引脚布局的图形视图可以进行放大和缩小，便于用户查看全局配置或局部的

详细信息。也可以点击重置按钮，将引脚布局图形重置为初始大小。搜索框中选择引脚、信号、标签，按回车键，将搜索到的引脚以蓝色闪烁显示。

：放大引脚视图。

：引脚视图恢复初始大小和位置。

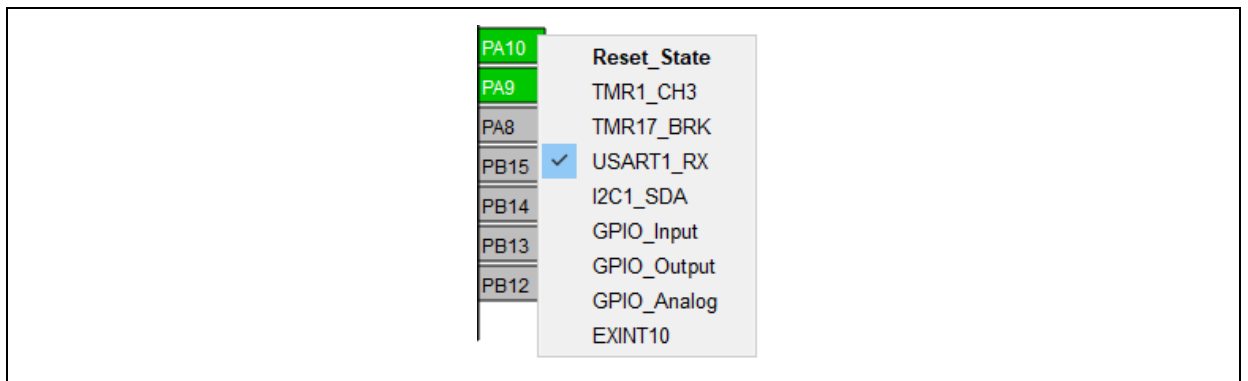
：缩小引脚视图。

：查询引脚、信号、标签在引脚视图中的位置。

同时支持 **Ctrl + 点击信号**，查找信号所在的引脚。

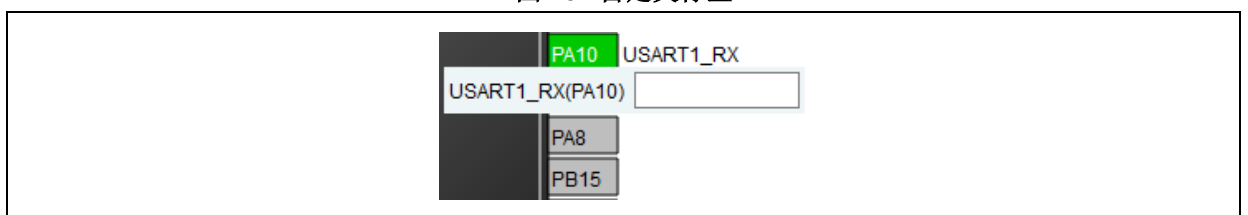
在引脚视图上，左键点击引脚(除固定模式引脚)，将弹出一个右键菜单，显示该引脚可配置的信号。当选择“Reset\_State”，将重置该引脚为未使用状态。如下图所示。

图 14. 引脚配置菜单



对于已经配置功能的引脚，右键单击将弹出“输入标签”的按钮，可以点击按钮，可以指定此信号的自定义标签。新标签将替换引脚视图中的设置的信号名称。

图 15. 自定义标签



## 4.2.4 GPIO 配置

单击“外设”窗体中的 GPIO，可以打开 GPIO 配置窗口，该窗口可用于配置 GPIO 的参数。GPIO 配置也可以在外设实例配置窗口中的专用窗口访问特定外设实例的 GPIO 设置。下拉框中可以选择显示的方式，可以显示所有或者按外设分组显示。“配置重置”按钮将重置所有已修改的 GPIO 参数。

图 16. GPIO 配置

GPIO 模式和配置

配置

显示所有

配置重置

GPIO

Pin名称	信号	输出电平	GPIO类型	上/下拉类型	GPIO模式	驱动能力	标签	已修改
PA10	USART1_RX	n/a	推挽	无上/下拉	复用功能模式	适中		N
PA11	USART1_CTS	n/a	推挽	无上/下拉	复用功能模式	适中		N
PA12	USART1_RTS	n/a	推挽	无上/下拉	复用功能模式	适中		N
PA9	USART1_TX	n/a	推挽	无上/下拉	复用功能模式	适中		N
PB12	SPI2_CS	n/a	推挽	无上/下拉	复用功能模式	适中		N
PC13	GPIO_Output	输出低电平	推挽	无上/下拉	输出模式	适中		N
PC14	GPIO_Input	n/a	n/a	无上/下拉	输入模式	n/a		N

双击可修改每个信号 GPIO 的参数，GPIO 可配置参数如下：

### ■ Pin 名称

信号所在的引脚名称。如果多个引脚有同一个信号，且引脚未被使用时，可双击切换信号到此引脚。（AT32F403A/F407/A403A/F413/F415 不支持此功能）

### ■ 输出电平

当信号为“GPIO\_Output”时，可配置“输出低电平”或“输出高电平”。

### ■ GPIO 类型

用于配置 GPIO 的类型（推挽、开漏）。

### ■ 上/下拉类型

被设置为一个默认值，可以在其他选择可用时进行配置。

### ■ GPIO 模式（模拟、输入、输出、复用功能）

在引脚布局视图中选择外设模式会自动以相关的复用功能和 GPIO 模式配置引脚。

### ■ 驱动能力

用于配置相应的 I/O 端口电流的能力（适中电流推动/吸入能力、较大电流推动/吸入能力）；

### ■ 标签

将默认名称更改为用户自定义的标签名称。引脚布局视图相应地更新。

## 4.2.5 DMA 配置

单击“外设”窗体中的 DMA，可以打开 DMA 配置窗口，该窗口用户配置 MCU 上可用的通用 DMA 控制器。DMA 接口允许在 CPU 运行时在内存和外设之间执行数据传输，以及内存到内存传输。部分外设拥有自己的 DMA 控制器，默认情况下可以在外设配置窗口进行配置。在 DMA 配置窗口中单击“添加”，在 DMA 配置窗口的最后一行添加一个新行，组合框中列出了可选的 DMA 请求名称。

图 17. DMA 配置

DMA 请求用于保留一个数据流，以便在外设与内存之间传输数据。流的优先级用于决定下一个 DMA 传输选哪个流。

可以通过 DMA 配置窗口完成对 DMA 的配置：

### ■ 传输方向

分为内存到外设、外设到内存与内存到内存。

### ■ 优先级

通道优先级分为 4 个等级，“最高优先级”、“高优先级”、“中等优先级”和“低优先级”。若有 2 个通道优先级设定相同，则较低编号的通道有较高的优先权。举例，通道 1 优先于通道 2。

### ■ 模式

包括普通模式、循环模式和外设流控制模式。

### ■ 外设地址递增

配置使能外设地址增量类型。启用后，外设地址在每次传输后都会递增。

### ■ 内存地址递增

配置使能内存地址增量类型。启用后，内存地址在每次传输后都会递增。

### ■ 外设数据宽度

配置外设数据宽度。“字节”宽度为 8 位；“半字”宽度为 16 位；“字”宽度为 32 位。

### ■ 内存数据宽度

配置内存数据宽度。“字节”宽度为 8 位；“半字”宽度为 16 位；“字”宽度为 32 位。



## 4.2.6 NVIC 配置

单击“外设”窗体中的 NVIC，可以打开 NVIC 配置窗口。

可以根据右上角的下拉框中的显示方式筛选显示的内容，包括显示所有的中断、有效的中断和已开启的中断。

图 18. NVIC 配置

中断表	开启	抢占优先级	子优先级
Reset_IRQ	<input checked="" type="checkbox"/>	0	0
NonMaskableInt_IRQ	<input checked="" type="checkbox"/>	0	0
HardFault_IRQ	<input checked="" type="checkbox"/>	0	0
MemoryManagement_IRQ	<input checked="" type="checkbox"/>	0	0
BusFault_IRQ	<input checked="" type="checkbox"/>	0	0
UsageFault_IRQ	<input checked="" type="checkbox"/>	0	0
SVCALL_IRQ	<input checked="" type="checkbox"/>	0	0
DebugMonitor_IRQ	<input checked="" type="checkbox"/>	0	0
PendSV_IRQ	<input checked="" type="checkbox"/>	0	0
SysTick_IRQ	<input type="checkbox"/>	0	0
WWDT_IRQ	<input type="checkbox"/>	0	0
PVM_IRQ	<input type="checkbox"/>	0	0
ERTC_IRQ	<input type="checkbox"/>	0	0
FLASH_IRQ	<input type="checkbox"/>	0	0
CRM_IRQ	<input type="checkbox"/>	0	0
EXINT1_0_IRQ	<input type="checkbox"/>	0	0
EXINT3_2_IRQ	<input type="checkbox"/>	0	0
EXINT15_4_IRQ	<input type="checkbox"/>	0	0
DMA1_Channel1_IRQ	<input type="checkbox"/>	0	0
DMA1_Channel3_2_IRQ	<input type="checkbox"/>	0	0
DMA1_Channel5_4_IRQ	<input type="checkbox"/>	0	0
ADC1_CMP_IRQ	<input type="checkbox"/>	0	0
TMR1_BRK_OVF_TRG_HALL_IRQ	<input type="checkbox"/>	0	0
TMR1_CH_IRQ	<input type="checkbox"/>	0	0
TMR3_GLOBAL_IRQ	<input type="checkbox"/>	0	0

表格中，中断名称为灰色的表示中断不可配置，需要开启对应的外设模式；使能状态为开启，并不可修改的为系统中断，这些中断不能禁用。

NVIC 配置包括选择优先级组、使能中断和配置中断优先级（抢占优先级和子优先级）：

### 1、选择一个优先级组

优先级组有多个位，可用于定义 NVIC 优先级。这些位被分为两个优先级组，对应两种优先级类型：抢占优先级和子优先级。优先级的位表示可以配置的优先级个数，如 0 位表示只能配置一个优先级 0，4 表示可以配置 16 个优先级(0-15)。

### 2、在中断表中，单击选择一个中断向量，在表格中进行配置：

- 使能：选中/取消选中可以启用/禁用中断。
- 抢占优先级：选择一个优先级。抢占优先级定义一个中断中断另一个中断的能力。
- 子优先级：选择一个优先级。子优先级定义中断优先级

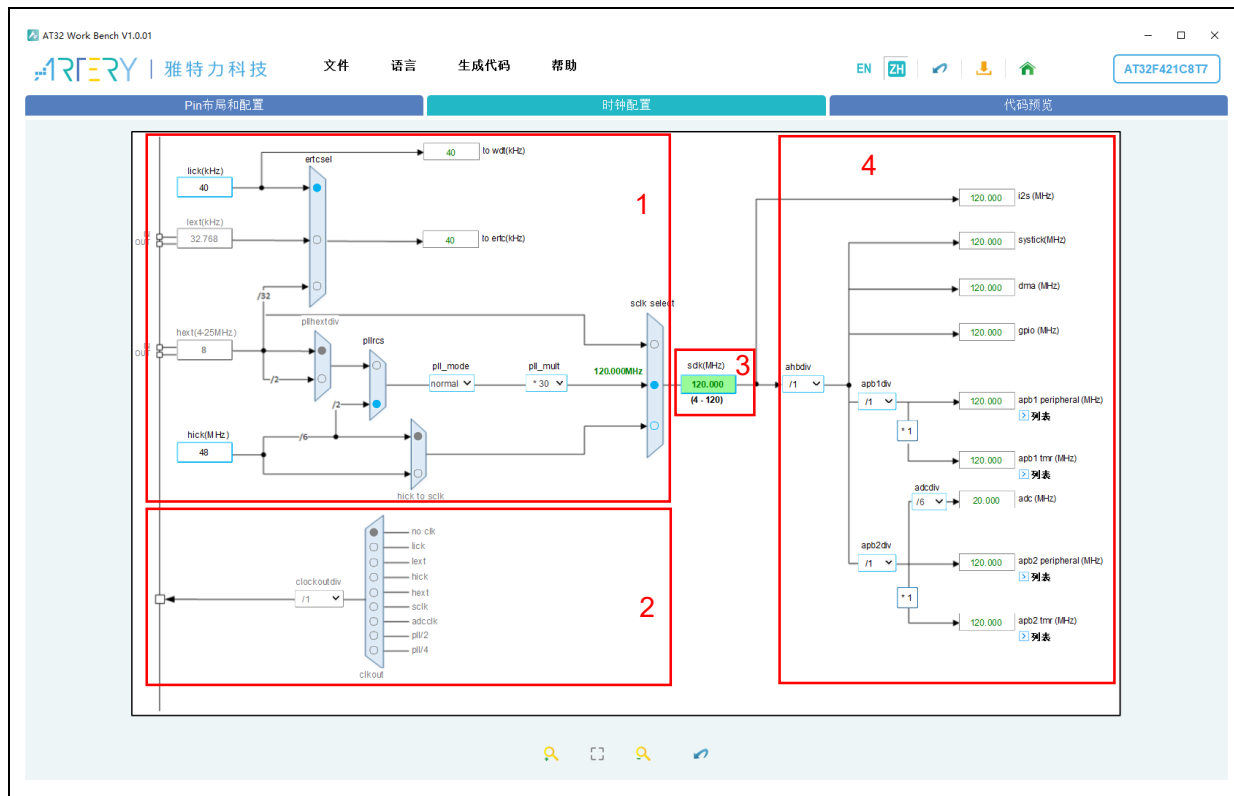
外设专用中断也可以通过“外设模式和配置”窗口中的 NVIC 设置进行配置。

## 4.3 时钟配置

时钟配置界面主要用来进行时钟路径及参数的配置。视图中的下拉菜单和输入框可用于修改实际的时钟树配置，以满足应用需求。

整个配置界面主要可以分为四个大块，如下图所示

图 19. 时钟配置



1. 配置部分：用于对时钟路径和时钟参数进行选择 and 配置，以达到期望的应用需求。
2. 输出部分：用于时钟输出（CLKOUT）的配置。
3. 在 SCLK 栏也可在选中 PLL 为系统时钟时作为输入框，可输入期望的系统时钟频率来反向自动配置出倍频参数。
4. 结果部分：用于显示当前外设所使用的时钟频率及总线上的外设。

时钟配置窗口支持 **Ctrl + 鼠标滚轮** 放大和缩小 MCU 时钟配置视图。

时钟配置窗口工具栏功能如下：

：放大时钟配置视图。

：时钟配置视图恢复初始大小和位置。

：缩小时钟配置视图。

：所有时钟配置重置。

注 1：需要使 **lxt** 为可用状态，需要在外设“CRM”的模式窗口中设置“低速外部时钟”的模式；

注 2：需要使 **hext** 为可用状态，需要在外设“CRM”的模式窗口中设置“高速外部时钟”的模式；

注 3：需要使 **clockout** 为可用状态，需要在外设“CRM”的模式窗口中勾选“时钟输出”。

## 4.4 代码预览

点击“代码预览”按钮，将自动生成当前配置的代码，并在窗口中显示代码内容。左侧显示了生成的代码文件，右侧显示选中文件的代码内容，例如：

- main.c: 主源文件，主要对各外设初始化函数进行调用。
- AT32xxxx\_wk\_config.h: 外设配置头文件，主要对各外设初始化函数进行声明。
- AT32xxxx\_wk\_config.c: 外设配置源文件，主要对各外设初始化函数进行定义。
- AT32xxxx\_int.h: 中断函数头文件。
- AT32xxxx\_int.c: 中断函数源文件。
- AT32xxxx\_conf.h: 库配置头文件。

可以实时查看当前配置生成的代码。

图 20. 代码预览



代码显示窗口，支持 **Ctrl + 鼠标滚轮** 放大和缩小功能，同时支持右键菜单，右键菜单功能如下：

- 折叠全部：将代码折叠，方便查看代码结构。
- 展开全部：将所有折叠的代码展开。
- 拷贝：拷贝选择的代码。
- 全选：选择代码全部内容。
- 查找：可在预览的代码中查找关键字。
- 编码：用于切换显示代码时的编码格式。

## 4.5 生成代码

点击菜单栏或工具栏的“生成代码”按钮，将弹出“生成代码”窗口。

### 4.5.1 生成代码

图 21. 生成代码

生成代码

项目设置

项目名称: AT32F402RCT7\_WorkBench

项目位置: D:/Test 浏览

工具链/IDE: MDK\_V5

☒ 重新生成代码时保留用户代码

☒ 所有未使用引脚设为模拟

☒ 外设初始化生成单独的“.c/.h”文件

☐ 添加所有的固件库文件到项目

☒ 只添加当前必须的固件库文件到项目

链接设置

堆最小值 (0x): 0x200

栈最小值 (0x): 0x400

MCU固件包

☒ 复制库文件到项目文件夹

固件包版本: V2.1.0 固件包管理

固件包位置: D:/WorkBench\_BSP/WK\_AT32F402\_405/AT32F402\_405\_Firmware\_Library\_V2.1.0

确定 取消

项目管理中配置项目生成时所需要参数，具体配置选项和操作如下：

#### ■ 项目设置

- 项目名称：用于创建项目的名称。
- 项目位置：项目文件夹的存储目录。
- 工具链/IDE：生成选定工具链/IDE 类型的工程项目。
- 重新生成代码时保留用户代码：[参照节 4.5.2](#)。
- 所有未使用引脚设为模拟：生成的代码中将未使用引脚 GPIO 设置为模拟，以节省功耗。
- 外设初始化生成单独的“.c/.h”文件：生成代码时，将外设初始化代码配置到单独的“.c/.h”文件中。以 USART 为例，将单独生成“usart.c”和“usart.h”。
- 添加所有的固件库文件到项目：将所有的固件库文件添加到项目文件中。
- 只添加当前必须的固件库文件到项目：将当前使用的外设的固件库文件添加到项目文件中。

#### ■ 链接设置

- 堆最小值和栈最小值：为应用分配的最小堆和栈的值。对于堆大小和栈大小，建议的默认值

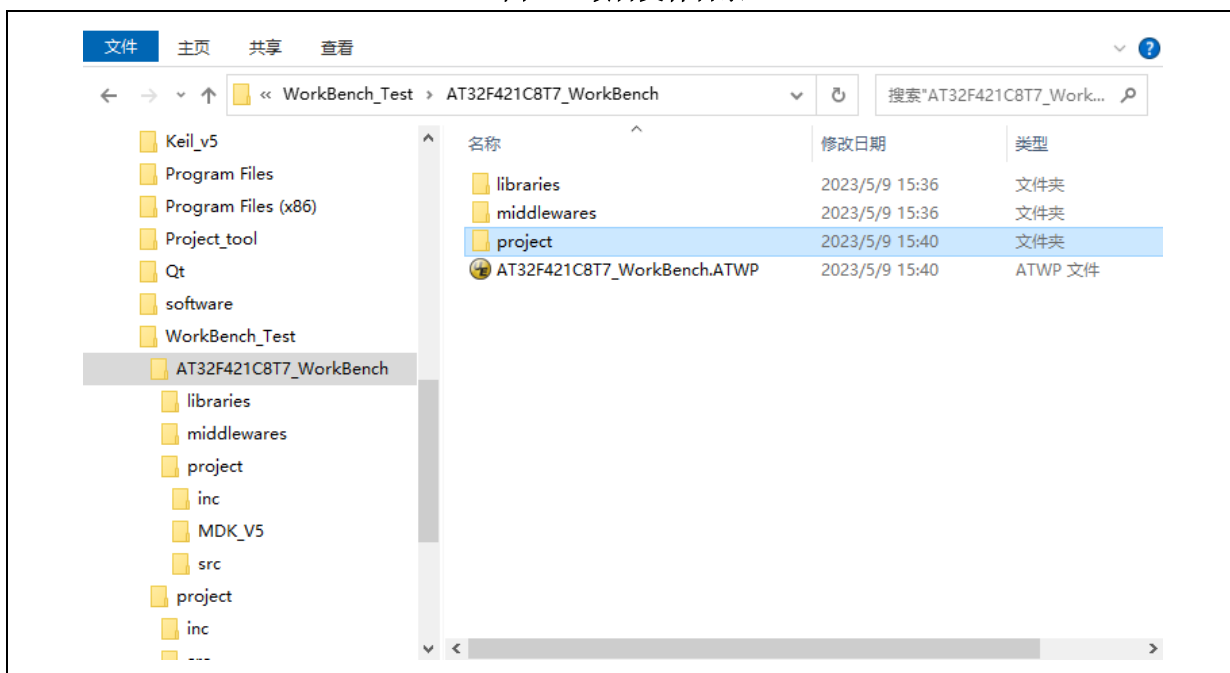
分别为 0x200 和 0x400。当应用使用中间件栈时，这些值可能需要增加。

#### ■ MCU 固件包

- 复制库文件到项目文件夹：选中后，生成代码时将固件包内的库文件自动复制到项目文件夹中。
- 固件包版本：选择已安装的固件包版本。如果未安装固件包，请点击“固件包管理”进行安装。
- 固件包管理：安装与管理软件支持的固件包。[参照节 4.6](#)。
- 固件包位置：显示已选择的固件包所在位置。

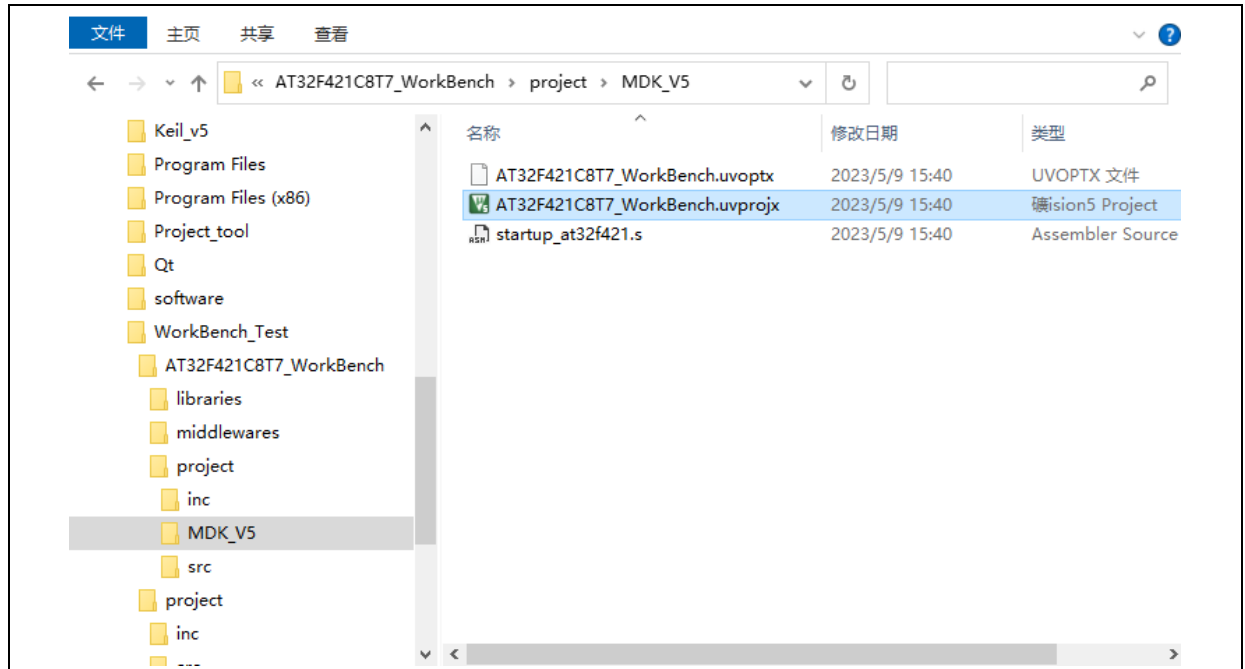
配置完项目选项后，点击“确定”按钮。将自动生成用户代码和选定 IDE 的项目文件。生成的项目文件结构如下图：

图 22. 项目文件目录



文件目录中包含了从固件包中复制的库文件 **libraries** 与中间件文件夹 **middlewares**，以及生成的项目文件夹 **project**。Projects 文件夹中包含生成的头文件夹 **inc**，源代码文件夹 **src** 和项目类型文件夹 **MDK\_V5**(根据所选中的工具链/IDE 生成)，MDK\_V5 中为 MDK 具体的项目文件，可以使用 Keil 开发环境打开该项目。

图 23. MDK\_V5 项目文件



### 4.5.2 重新生成代码时保留用户代码

AT32 Work Bench 生成的 C 代码提供用户添加自定义代码。用户自定义代码需插入到软件定义位置并可在下一次生成 C 代码时保留。定义的位置如下所示：

```
/* add user code begin ..... */
```

```
/* add user code end ..... */
```

重新生成代码时，不会对软件定义位置的用户代码进行移动或重命名，将会保留软件定义位置的用户代码，用户创建的未在软件定义位置的代码将被忽略并丢弃。

注意：

1. 不支持用户自定义 **add user code** 标签，仅能在软件定义的标签中添加自定义代码。
2. 文件中代码注释不能写成与软件定义的 **add user code** 标签相同。

以 main.c 中 main 函数为例，软件定义的可添加用户代码的位置如下黑体部分区域内：

```
int main(void)
{
    /* add user code begin 1 */

    /* add user code end 1 */

    /* system clock config. */
    wk_system_clock_config();

    /* nvic config. */
    wk_nvic_config();

    /* add user code begin 2 */

    /* add user code end 2 */

    while(1)
    {
        /* add user code begin 3 */

        /* add user code end 3 */
    }
}
```

## 4.6 固件包管理

点击菜单栏 “文件” - “固件包管理”，将弹出“固件包管理”窗口。

图 24. 固件包管理



### ■ 固件包安装位置

需选择与确定固件包的安装位置，软件每次启动将会在此位置扫描已安装的固件包。

已安装的固件包在此安装位置以 WK\_AT32xxx 目录存在，例如：WK\_AT32F425。

注意：软件版本更新后，需重新选择固件包安装位置。

### ■ 从本地安装

从本地已下载的固件包进行安装，需手动选择已下载的固件包 (\*.ZIP)。

### ■ 从网络安装

自动从网络下载固件包并进行安装。

### ■ 删除

需勾选已安装的固件包，并进行删除。



## 5 文档版本历史

表 4. 文档版本历史

日期	版本	变更
2024/10/28	V1.06	更新部分说明内容以及图片。
2024/08/13	V1.05	更新部分说明内容以及图片。
2024/03/05	V1.04	更新部分说明内容。
2024/01/26	V1.03	新增“固件包管理”章节。
2023/11/28	V1.02	更新部分图片。
2023/09/08	V1.01	更新部分说明内容。
2023/05/09	V1.00	初始版本。

**重要通知 - 请仔细阅读**

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途(及其依据任何司法管辖区的法律的对应情况)，或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：(A) 对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；(B) 航空应用；(C) 航天应用或航天环境；(D) 武器，且/或(E) 其他可能导致人身伤害、死亡及财产损害的应用。如果采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险及法律责任仍将由采购商单独承担，且采购商应独力负责在前述应用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2024 雅特力科技 保留所有权利