

## UM0012

AT32 Work Bench User Manual

## Introduction

This user manual gives an overview of AT32 Work Bench. The AT32 Work Bench can generate initialization C code and corresponding IDE project through MCU graphical configuration, so as to reduce the development workload, time and cost.

AT32 Work Bench has the following features:

- 1. Support peripheral initialization configuration
- 2. Support PIN MUX configuration and customized PIN label
- 3. Support automatic system clock configuration
- 4. Support online code view
- 5. Support "add user code" function (existing code is not overwritten by the new project)
- 6. Support automatic project generation in Keil, IAR and AT32 IDE
- 7. Record recent designs
- 8. Generate configuration report (.pdf)
- 9. Support simplified Chinese & English menu
- 10. Support Windows and Linux
- 11. Support middleware such as FREERTOS, USB\_DEVICE and USB\_HOST
- 12. Support online software upgrade

# Contents

1	Introduction				
	1.1	Environmental requir	ements	5	
2	Insta	lation		6	
	2.1	Set up AT32 Work Be	ench on Windows	6	
	2.2	Set up AT32 Work Be	ench on Linux	6	
3	Getti	g started	······································	7	
4	Proje	ct configuration		8	
	4.1	Venu bar and toolba		8	
		4.1.1 Menu bar		8	
		4.1.2 Toolbar		9	
	4.2	Pin out configuration		9	
		4.2.1 Peripherals		9	
		4.2.2 Mode and conf	guration1	0	
		4.2.3 Pin layout		3	
		4.2.4 GPIO mode an	d configuration1	5	
		4.2.5 DMA mode and	configuration1	6	
		4.2.6 NVIC mode and	d configuration1	7	
	4.3	4.3 Clock configuration		9	
	4.4	Code view2			
	4.5		1		
		4.5.1 Project manage	er2	1	
		4.5.2 Keep user code	when re-generating2	4	
	4.6	Package manager		5	
5	Revi	ion history		7	



# List of tables

Table 1. Peripheral status	9
Table 2. Status of mode and parameters	11
Table 3. Status of pins	14
Table 4. Document revision history	27



# List of figures

Figure 1. dpkg command in Linux	6
Figure 2. Graphical installation	6
Figure 3. Getting started page	7
Figure 4. Project configuration	8
Figure 5. Menu bar and toolbar	8
Figure 6. Peripheral mode and configuration	10
Figure 7. USART1 mode and configuration	10
Figure 8. Mode error prompt	11
Figure 9. Parameters settings	11
Figure 10. GPIO settings	12
Figure 11. DMA settings	12
Figure 12. NVIC settings	13
Figure 13. PIN layout	13
Figure 14. Pin configuration menu	14
Figure 15. Custom label	15
Figure 16. GPIO mode and configuration	15
Figure 17. DMA mode and configuration	16
Figure 18. NVIC mode and configuration	
Figure 19. Clock configuration	19
Figure 20. Code view	20
Figure 21. Project manager	21
Figure 22. Project file directory	
Figure 23. MDK_V5 project files	
Figure 24. Package Manager window	25



## 1 Introduction

## **1.1 Environmental requirements**

#### Software

#### Windows

Windows 7 and above is required.

#### Linux

Ubuntu or Fedora that supports AMD x86\_64.

Hardware
 At least 2GB RAM.
 At least 4GB hard drive space.



## 2 Installation

## 2.1 Set up AT32 Work Bench on Windows

Run the executable AT32\_Work\_Bench.exe directly, without the need for installation.

## 2.2 Set up AT32 Work Bench on Linux

This software supports Ubuntu 16.4 and above.

Installation methods on Linux: dpkg command and graphical installation.

dpkg command

As shown in Figure 1, enter the following command in the terminal for setup:

sudo dpkg -- i AT32\_Work\_Bench\_linux\_amd64\_V1.0.0.deb





Graphical installation

Copy the AT32\_Work\_Bench\_linux\_amd64\_V1.0.0.deb to Linux and double click. Then, click on the "Install", as shown in Figure 2.



<	at32-work-bench	•••
	<b>t32-work-bench</b> T32 Work Bench Project	
Install		
AT32 Work Ben	ich Project	
Details		
Version Updated License	1.0.0 05/05/2023 Proprietary	
Source Download Size	AT32_Work_Bench_linux_amd64.deb 0 bytes	

After the installation is complete, click the "All Programs" button at the bottom of the left taskbar, find and click on the "AT32 Work Bench" in the program list to start AT32 Work Bench.





## **3 Getting started**

The getting started page is the first window that opens when AT32 Work Bench is started. It includes three options, i.e., "Start a New Design", "Open an Existing Design" and "Recent Designs".

■ Start a new design

Select the MCU serials and model, and the package, Flash and SRAM are filled in automatically according to the model. Click on "New" to create a new project.

Open an existing design

Users can find a saved project through File Explorer and then open it (\*.ATWP file).

Recent designs

It displays a list of recently created projects. Users can select one to open.

#### Figure 3. Getting started page

is device features
16+6 KB SRAM
er, and another KB. In addition, the
ARTs, 3x SPIs/I'Ss
general-purpose tv and almost all of
,
ng LQFP64,





#### **Project configuration** 4

After a new project is created or a saved file is loaded, the project configuration page opens. It contains a menu bar, a toolbar, and the following set of views:

- Pin out configuration
- **Clock configuration**
- Code view



Figure 4. Project configuration

Pin out configuration: Users can configure peripheral pins and relevant parameters. Clock configuration: Users can configure MCU clock as needed. Code view: Users can view the code automatically generated according to the current configuration.

Click on the "Generate code" in the menu bar or the 📥 icon in the toolbar to generate the

corresponding user code and project.

#### 4.1 Menu bar and toolbar

Figure 5 shows the menu bar and toolbar.

```
Figure 5. Menu bar and toolbar
```

lenu bar		
I File		
New design	Create a new	project.
Open design	Open a saved	l project.
Save design	Save the curr	ent project.
Design save as	Save the curr	ent project as…



Generate report Recent design

Display recent projects. I**er** 

Package Manager Package Manager

Install and manage BSP.

Generate a project report (.pdf).

Generate code

After the peripheral parameters and clock are configured as needed, click on "Generate code" to select the project location and generate the corresponding project file.

#### ■ Help

Open manual	Open the user manual.
New version download	Download the new version online.
Go to Artery home page	Open browser to visit the official website of Artery.
Version	View the current version.

### 4.1.2 Toolbar

EN: Select English as the display language.

ZH: Select simplified Chinese as the display language.

✓ Reset. Reset all peripheral parameters, pins and clock.

4. Generate code. Open "Project Manager" window and generate the corresponding code. Refer

to Section 4.5.

The second state of the second state of a state of a

AT32F421C8T7 : Display the current MCU model.

## 4.2 Pin out configuration

Pin out configuration: Configure pins and parameters of MCU peripherals. It contains the "Peripherals", "Mode and Configuration" and "Pin layout" windows. Users can adjust the window size as follows: hover the mouse over the window border and then a two-way arrow appears; then, hold down the left mouse button and move to expand or shrink the window size.

## 4.2.1 Peripherals

The "Peripherals" window displays a list of available peripherals and middleware for the selected MCU model. The icon in front of peripheral name indicates the peripheral configuration status.

lcon	Status	Description		
	Not configured	Mode and parameters are not configured.		
0	Configured successfully	Mode and parameters are configured successfully.		
8	Configuration parameter error	The mode or relevant parameters are incorrect. Please		
		confirm and re-configure as needed.		



### 4.2.2 Mode and configuration

Select a peripheral from the list of peripherals, and then the corresponding "Mode and Configuration" window opens. Users can select peripheral mode and the corresponding MCU pins in the "Mode". For AT32 MCUs, same pins can be used for different peripherals and multiple features; therefore, once the mode is selected, the optimum pin layout will be configured automatically.

	Pin.out Configuration	Clock Configuration	Code View
Module Categories	USA	RT1 Mode and Configuration	Pin layout
Parighensk ACC ACC ACC ACC ACC ACC ACC ACC ACC AC	Mode         Asynchro           Hardware Flow Control (RS22)         Deable           □         Hardware Flow Control (RS45)         Deable           Parameters Setting         CP/O Setting         MVC           ✓ Bark Flowmeters         Bark Data (255-1550000)         Rail Bark Rate           Data Bark Rate         Data Breameters         Data Breameters           ✓ Advanced Features         Data Direction         STOP bit num           ✓ Advanced Features         To painty reverse         To painty reverse           The painty reverse         The painty particle painty reverse         MSB transmit frot	Unde           Oranization           V           V           Status           115200 Bits/s           115200 Bits/s           Dec           8 Bit (roding Darky)           None           V           Raceire Only           Dashe           Dashe           Dashe           Dashe	



Take the USART1 as an example:

USART1 mode

#### Figure 7. USART1 mode and configuration

USART1 Mode and Configuration			
Mode			
Mode	Disable	~	
Hardware Flow Control	Disable	$\sim$	

As shown in Figure 7, when the USART1 is set to asynchronous mode, PA9 and PA10 are mapped to "USART1\_TX" and "USART1\_RX automatically; when "CTS/RTS" is selected for hardware flow control, PA11 and PA12 are mapped to "USART1\_CTS" and "USART1\_RTS" automatically. When the condition is not satisfied or in case of signal conflict, the corresponding content is displayed in a different color or background color, as shown in Table 2. Move the mouse over the corresponding content, and a message appears to prompt users to modify or re-configure, as shown in Figure 8.



#### Figure 8. Mode error prompt



Examples of status of mode and parameters:

Table 2.	Status	of mode	and	parameters
----------	--------	---------	-----	------------

Icon	Status	Description
Hardware Flow Control CTS/RTS	Configurable	
Clock Output	Condition not satisfied	Configure other parameters as
		prompted.
Please select a Inverting input selection (Input [-])	Configure the	Select "Inverting input selection
	corresponding mode	(Input[-])".
IN5	Signal conflict	There is a conflict with other function
		signals.
Parameters Settings GPI0 Settings NWC Settings DMA Settings     Settings	Parameter error	The parameter is beyond the required
Baud Rate (3-14648) 115200 Bits/s		value range.

#### Parameters settings

#### Figure 9. Parameters settings

Parameters Settings	GPIO Settings	NVIC Settings	DMA Settings	
✓ Basic Parameters				
Baud Rate (1831-7500000)		115200 Bits/s		
Data bit num		8 Bits (including Pari	ty)	~
Parity selection	n	None		~
STOP bit num		1		~
<ul> <li>Advanced Parameters</li> </ul>				
Data Direction	I	Receive and Transmi	t	~
TX and RX Pir	is Swapping	Disable		~

This window displays configurable USART1 parameters, including the "baud rate" and "data bit number" in the "Basic Parameters" and "data direction" and "Tx and Rx pins swapping" in the "Advanced Parameters".

If an invalid setting is detected, the corresponding parameter will be optimized automatically according to the value range. For example, when the user-defined value is lower than the minimum value (or greater than the maximum value), it is automatically reset to the minimum (or maximum) value.

GPIO settings



#### Figure 10. GPIO settings

This window displays configurable GPIOs, for example, GPIOs for "USART1\_TX" and "USART1\_RX" in Figure 10.

For details about GPIO settings, refer to <u>Section 4.2.4</u>.

DMA settings

#### Figure 11. DMA settings

Parameters Settings	GPIO Settings	NVIC Settings	DMA Settings	
DMA Request	Channel	Direction	Priority	
USART1_RX	DMA1 Channel 3	Peripheral To Memory	Low	
USART1_TX	DMA1 Channel 2	Memory To Peripheral	Low	
Add Dele DMA Request Param	te eters Mode	N		
	Peripheral Increment Memory Increment	P	ormal  veripheral Inc Disable  veripheral Inc Enable  veripherable  veri	

This window allows users to add configurable DMA requests for the current peripheral, for example, the corresponding DMA channels and DMA requests for "USART1\_TX" and "USART1\_RX" in Figure 11.

For details about DMA settings, refer to <u>Section 4.2.5</u>.

NVIC settings



#### Figure 12. NVIC settings

NVIC Interrupt Table	Enabled	Dreemption Priority	Sub Priority
DMA1 Channel3 2 JPO			
		0	0
USARIT_IRQ		0	0

This window displays configurable interrupts for the current peripheral. When the corresponding DMA channel is enabled, the corresponding DMA channel interrupt can be configured, for example, the "USART1\_IRQ" and "DAMA1\_Channel3\_2\_IRQ" in Figure 12.

The "preemption priority" and "sub priority" are configured in the "NVIC Mode and configuration" window. Refer to <u>Section 4.2.6.</u>

## 4.2.3 Pin layout

The pin layout of the selected package (such as LQFP48, QFN32 and TSSOP20) is displayed in graphic. Each pin is represented by its name (such as PA9), configuration status, and current signal distribution.



Figure 13. PIN layout



Pins are displayed in different colors and have different functions according to their status, as shown in Table 3.

lcon	Status	Description
	Pin not in use	Users can configure signals for this pin.
	Pin in use	A valid signal is configured, and it has a corresponding
		peripheral mode.
	Pin no mode	A signal is configured, but it has no corresponding peripheral
		mode.
	Fixed mode	Fixed configuration, not support other functions.
	Pin searched	Click on the "Search" icon, and the searched pins are
		displayed in blinking blue.
•	Peripheral pin label	Click on a peripheral in the peripheral list on the left to label all
		pin positions for that peripheral.

#### Table 3. Status of pins

Users can press the Ctrl + scroll mouse wheel to adjust the pin layout window size, and move the mouse to relocate its position.

Click on the chip internal area to show or hide the pin number.

In addition, users can zoom in/out the pin layout by using the below toolbar to view the global configuration or local details. Click on the "Reset" icon to restore the pin layout to its initial size. Select the corresponding pin, signal or label in the search box and press Enter; then, the searched pins are displayed in blinking blue.

- Soom into the pin layout.
- : Restore to the initial size and position.
- <sup>Q</sup>: Zoom out the pin layout.

Search. Find the positions of pins, signals and labels in the pin layout.

User can press the Ctrl + click on the signal to search for the pin where the signal is located. In the pin layout, left click on the pin (except for pins in fixed mode), and then a right-click menu pops up, showing configurable signals for the pin. Select "Reset\_State" to reset this pin to the unused status, as shown in Figure 14.

	_	
PA10		Reset_State
PA9		TMR1_CH3
PA8		TMR17_BRK
PB15	~	USART1_RX
PB14		I2C1_SDA
PB13		GPIO_Input
PB10		GPIO_Output
PB12		GPIO_Analog
I		EXINT10

#### Figure 14. Pin configuration menu



Right click on the configured pin, and then an "Enter label" button pops up. Users can click on the button to specify a custom label for this signal. The new label replaces the signal name set in the pin layout.

#### Figure 15. Custom label



## 4.2.4 GPIO mode and configuration

Click on "GPIO" under "Peripherals" to open the GPIO Mode and Configuration window. Users can also configure GPIO for specific peripherals in a dedicated window for GPIO configuration. Click on the drop-down box to select "Show All" or "Groups by Peripherals". Click on "Reset Configuration" button to reset all GPIO parameters.

Show All	Pin     Output level       RX     n/a       TS     n/a       TS     n/a       TX     n/a       TX     n/a       Output level low	GPIO type Push Pull Push Pull Push Pull Push Pull Push Pull	Config Pull type Pull-none Pull-none Pull-none Pull-none Pull-none	GPIO mode Mux function mode Mux function mode Mux function mode Mux function mode	Driver capability Moderate Moderate Moderate Moderate	Rese	et Configura Modif N N
Show All       GPIO       Pin Name     Signal on I       PA10     USART1_R       PA11     USART1_C       PA12     USART1_R       PA9     USART1_T       PB12     SPI2_CS       PC13     GPI0_Outp       PC14     GPI0_Input	Pin Output level RX n/a TS n/a RTS n/a TX n/a TX n/a tx n/a	GPIO type Push Pull Push Pull Push Pull Push Pull Push Pull	Pull type Pull-none Pull-none Pull-none Pull-none	GPIO mode Mux function mode Mux function mode Mux function mode Mux function mode	Driver capability Moderate Moderate Moderate Moderate	Rese Label	et Configura Modif N N N
GPIO       Pin Name     Signal on P       PA10     USART1_R       PA11     USART1_C       PA12     USART1_R       PA9     USART1_T       PB12     SPI2_CS       PC13     GPI0_Outp       PC14     GPI0_Inp	Pin Output level RX n/a TS n/a RTS n/a TX n/a TX n/a ty Output level low	GPIO type Push Pull Push Pull Push Pull Push Pull Push Pull	Pull type Pull-none Pull-none Pull-none Pull-none	GPIO mode Mux function mode Mux function mode Mux function mode	Driver capability Moderate Moderate Moderate Moderate	Label	Modit N N
Pin Name         Signal on P           PA10         USART1_R           PA11         USART1_C           PA12         USART1_R           PA9         USART1_T           PB12         SPI2_CS           PC13         GPI0_Outp           PC14         GPI0_Input	Pin         Output level           RX         n/a           CTS         n/a           RTS         n/a           RTX         n/a           CTS         n/a           CTS         n/a	GPIO type Push Pull Push Pull Push Pull Push Pull Push Pull	Pull type Pull-none Pull-none Pull-none Pull-none	GPIO mode Mux function mode Mux function mode Mux function mode Mux function mode	Driver capability Moderate Moderate Moderate	Label	Modif N N N
PA10         USART1_R           PA11         USART1_C           PA12         USART1_R           PA9         USART1_T           PB12         SPI2_CS           PC13         GPI0_Outp           PC14         GPI0_Input	RX n/a CTS n/a CTS n/a CTS n/a CTX n/a CTX n/a CTX n/a	Push Pull Push Pull Push Pull Push Pull Push Pull	Pull-none Pull-none Pull-none Pull-none	Mux function mode Mux function mode Mux function mode Mux function mode	Moderate Moderate Moderate Moderate		N N N
PA11     USART1_C       PA12     USART1_R       PA9     USART1_T       PB12     SPI2_CS       PC13     GPI0_Outp       PC14     GPI0_Input	CTS n/a CTS n/a TX n/a TX n/a out Output level low	Push Pull Push Pull Push Pull Push Pull	Pull-none Pull-none Pull-none	Mux function mode Mux function mode Mux function mode	Moderate Moderate Moderate		N
PA12     USART1_R       PA9     USART1_T       PB12     SPI2_CS       PC13     GPI0_Outp       PC14     GPI0_Input	n/a           TX         n/a           S         n/a           Dut         Output level low	Push Pull Push Pull Push Pull	Pull-none Pull-none Pull-none	Mux function mode Mux function mode	Moderate Moderate		N
PA9         USART1_T           PB12         SPI2_CS           PC13         GPI0_Outp           PC14         GPI0_Input	TX n/a	Push Pull Push Pull	Pull-none Pull-none	Mux function mode	Moderate		
PB12         SPI2_CS           PC13         GPI0_Outp           PC14         GPI0_Input	n/a	Push Pull	Pull-none				N
PC13 GPIO_Outp PC14 GPIO_Inpu	put Output level low			Mux function mode	Moderate		N
PC14 GPIO_Inp	put output level low	Push Pull	Pull-none	Output mode	Moderate		N
	ut n/a	n/a	Pull-none	Input mode	n/a		N

Double click to modify GPIO parameters for each signal. The configurable GPIO parameters include:

#### Pin Name

The name of the pin where the signal is located. If several pins have the same signal and the pin is not used, double click to switch signal to this pin.

(This function is not supported on AT32F403A/F407/A403A/F413/F415 devices)



Output level

When the current signal is "GPIO\_Output", it can be configured as "Output level low" or "Output level high".

- GPIO type
- Configure the GPIO type (push-pull/open-drain).
- Pull type

It is set to a default value, which can be configured accordingly.

GPIO mode (analog, input, output and MUX function)

Select a mode in the pin layout, and the corresponding pin is configured according to the MUX function and GPIO mode.

Driver capability

Configure the I/O port drive capability (moderate sourcing/sinking strength and stronger sourcing/sinking strength).

Label

Change the default name to a user-defined name, and the pin layout changes accordingly.

## 4.2.5 DMA mode and configuration

Click on "DMA" under "Peripherals" to open the DMA Mode and Configuration window to configure available general-purpose DMA controller. DMA interface supports data transfer between memory and peripherals and memory-to-memory data transfer when CPU is running.

Some peripherals have a dedicated DMA controller, which by default can be configured in the "DMA Mode and Configuration" window.

Click on "Add" to add a new line in the last row of the table in "Configuration" window. Optional DMA requests are listed in the combo box.

		Config	uration			
DMA1	MemToMem	1				
DMA Request	Channel	Direction	Priority			
USART1_RX	DMA1 Channel 3	Peripheral To Memory	Low			
USART1_TX	DMA1 Channel 2	Memory To Peripheral	Low			
Select						
MEMTOMEM USART2_RX						
Add Delet	e					
Add Delet	e tters Mode		Normal		~	
Add Delet	e etters Mode Peripheral Increi	nent	Normal Peripher.	al Inc Disable	<b>v</b>	
Add Delet	e sters Mode Peripheral Increm Memory Increme	nent ənt	Normal Periphera Memory	I Inc Disable	> > >	
Add Delet	e eters Mode Peripheral Increm Memory Increme Peripheral Data	ment ant Alignment	Normal Peripher: Memory Byte	al Inc Disable Inc Enable	<b>&gt;</b> <b>&gt;</b> <b>&gt;</b>	

#### Figure 17. DMA mode and configuration

DMA request is used to reserve a data flow for data transfer between peripherals and memory. The priority determines which flow to select for the next DMA transfer.

Users can configure DMA in the "DMA Mode and Configuration" window.

Direction



Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfers.

Priority

There are four levels, including very high priority, high priority, medium priority and low priority. If the two channels have the same priority level, then the channel with lower number will get priority over the one with higher number. For example, channel 1 has priority over channel 2

Mode

There are normal mode, circular mode and peripheral flow control mode.

Peripheral increment

Configure peripheral increment type and enable peripheral increment. Once enabled, the peripheral address increments after each transfer.

Memory increment

Configure memory increment type and enable memory increment. Once enabled, the memory address increments after each transfer.

Peripheral data alignment

Configure peripheral data bit width (byte: 8-bit, half-word: 16-bit, word: 32-bit).

Memory data alignment

Configure memory data bit width (byte: 8-bit, half-word: 16-bit, word: 32-bit).

## 4.2.6 NVIC mode and configuration

Click on "NVIC" under "Peripherals" to open the NVIC Mode and Configuration window. Click on the "Show" drop-down box to select to display all interrupts, available interrupts and enabled interrupts.

Configuration						
ty Group 4 bits for pre-emption priority, 0 bits t	Group 4 bits for pre-emption priority, 0 bits for subpriority 💙 Show All interrupts					
NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority			
Reset_IRQ	$\checkmark$	0	0			
NonMaskableInt_IRQ	$\checkmark$	0	0			
HardFault_IRQ	$\checkmark$	0	0			
MemoryManagement_IRQ	$\checkmark$	0	0			
BusFault_IRQ	$\checkmark$	0	0			
UsageFault_IRQ	$\checkmark$	0	0			
SVCall_IRQ	$\checkmark$	0	0			
DebugMonitor_IRQ	$\checkmark$	0	0			
PendSV_IRQ	$\checkmark$	0	0			
SysTick_IRQ		0	0			
WWDT_IRQ		0	0			
PVM_IRQ		0	0			
ERTC_IRQ		0	0			
FLASH_IRQ		0	0			
CRM_IRQ		0	0			
EXINT1_0_IRQ		0	0			
EXINT3_2_IRQ		0	0			
EXINT15_4_IRQ		0	0			
DMA1_Channel1_IRQ		0	0			
DMA1_Channel3_2_IRQ		0	0			
DMA1_Channel5_4_IRQ		0	0			
ADC1_CMP_IRQ		0	0			
TMR1_BRK_OVF_TRG_HALL_IRQ		0	0			
TMR1_CH_IRQ		0	0			

#### Figure 18. NVIC mode and configuration

As shown in Figure 18, interrupts in gray are not configurable and the corresponding peripheral mode needs to be enabled; interrupts with the "Enabled" check box ticked and cannot be modified are system interrupts, and these interrupts cannot be disabled.

NVIC configuration includes "Priority Group", "Enabled" (check box) and interrupt priority (preemption priority and sub priority).

1. Priority Group

The priority group has multiple bits that can be used to define the NVIC priority. These bits are divided into two priority groups, corresponding to two priority types, i.e., preemption priority and sub priority. Priority bits indicate the number of priorities that can be configured, for example, 0: only one priority0, 4: 16 priorities (0-15).

- 2. Click on an interrupt in the "NVIC Interrupt Table" to configure:
  - Enabled: tick/untick to enable/disable this interrupt.
  - Preemption priority: Select a preemption priority. It defines the ability of one interrupt to interrupt another.
  - Sub priority: Select a sub priority. It defines the interrupt priority.

Dedicated peripheral interrupts also can be configured in the NVIC Mode and Configuration window.



## 4.3 Clock configuration

Users can configure the clock path and parameters in the Clock Configuration window, and use drop-down menus and input boxes to modify the actual clock tree configuration to meet application requirements.

As shown in Figure 19, the Clock Configuration window mainly includes four blocks.



#### Figure 19. Clock configuration

- 1. Configuration: Select and configure the clock path and parameters as needed.
- 2. Output: Configure the clock output (CLKOUT).
- 3. SCLK: It is an input box when PLL is used as the system clock. Users can input the desired system clock frequency to automatically configure the frequency multiplication factor.
- 4. Result: Display the clock frequency of the current peripheral, and peripherals on bus.

Users can press the Ctrl + scroll mouse wheel to adjust the MCU clock configuration window size. The toolbar of the Clock Configuration window has the following functions:

Community into the clock configuration view.

: Restore the clock configuration view to the initial size and position.

- Com out the clock configuration view.
- : Reset clock configuration.

Note 1: Set the "LEXT" mode in the "CRM Mode and Configuration" window to enable LEXT. Note 2: Set the "HEXT" mode in the "CRM Mode and Configuration" window to enable HEXT. Note 3: Tick "Clock Output" in the "CRM Mode and Configuration" window to enable clockout.



### 4.4 Code view

Click on the "Code View" to generate code automatically. Code files are listed in the left, and the corresponding code is shown in the right window.

- main.c: main source file, which is used to call peripheral initialization functions.
- AT32xxxx\_wk\_config.h: header file for peripheral configuration, which is used for declaration of each peripheral initialization function.
- AT32xxxx\_wk\_config.c: source code for peripheral configuration, which is used to define each peripheral initialization function.
- AT32xxxx\_int.h: header file of interrupt functions.
- AT32xxxx\_int.c: source file of interrupt functions.
- AT32xxxx\_conf.h: header file for library configuration.

Users can view the code generated automatically according to the current configuration.

IN-ALL PRIME	10.00.0			
Pin ou	I_Configuration	Clock Configuration	Code View	
oject	mainie 🛛 💌			
<ul> <li>nc</li> <li>et 21402,405,corfh</li> <li>et 21402,405,corfh</li> <li>et 21402,405,inh</li> <li>et 21402,405,inh</li> <li>et argument</li> <li>et argument</li></ul>	1 /* add user code begin Header */ 2 /** 4 /** 5 /** 5 /** 6 /** 6 /** 6 /** 7 /* dd user code and Plaster Journess, ddaw 7 /* for Lucks Textre Pearl Textre Journess, ddaw 7 /* of there is an experiment of the second	cc & Disclaimer S59) that is made available to a is the copyrighted work of Artery. Gogy, and distribute the S59 on for the purpose of design and ry microcontroliner. Use of the th notice and the following disclaimer. BASIS WITFOUN MEMORYLINE, NHON-INFERIMENTING, PULLD WARAITES, DWIELD ON ArtESS ON REPRESIVATIONS, PULLD WARAITES, ON REPRESIVATIONS, PULLD WARAITES, ON REPRESIVATIONS, PULL WARAITES, PULL WARAI		

#### Figure 20. Code view

Users can press the Ctrl + scroll mouse wheel to adjust the code view size, and use the right click menu to:

- Fold all: fold code to view its structure;
- Expand all: expand all folded code;
- Copy: cope the selected code;
- Select all: select all of the code.
- Find: Search keywords in the preview code;
- EnCoding: Switch the encoding format to display code.



### 4.5 Generate code

Click on the "Generate code" in the menu bar or the 🦶 icon in the toolbar, and then the "Project Manager" window pops up.

### 4.5.1 **Project manager**

Project Name	AT32F402RCT7_WorkBench
Project Location	D:/Test Browse
Toolchain/IDE	MDK_V5
Keep User Code	when re-generating
Set all free pins a	as analog
Generate periphe	eral initialization as a pair of '.c/.h' files per peripheral
Add all firmances 15	and firs into the environt
Add all firmware lit Add only the neces	ranes files into the project
ker Settings	
Minimum Heap Size (	0x) 0x200
Minimum Stack Size	(0x) 0x400
CU and Firmware Pac	kage
	o the project folder
Copy libraries int	
Copy libraries int	
Copy libraries int	V2.1.0   Package Manager
Copy libraries int Package Version Firmware Relative Pa	V2.1.0   Package Manager  th D:WorkBench BSP/WK AT32E402 405/AT32E402 405 Firmware Library V2.1.0
Copy libraries int Package Version Firmware Relative Pa	V2.1.0  Package Manager th D:/WorkBench_BSP/WK_AT32F402_405/AT32F402_405_Firmware_Library_V2.1.0

Figure 21. Project manager

Users can configure required parameters as follows:

#### Project Settings

- Project name: create the project name.
- Project location: storage directory for the project folder.
- Toolchain/IDE: Generate a project of the selected toolchain/IDE.
- Keep User Code when re-generating code. Refer to <u>Section 4.5.2</u> for details.
- Set all free pins as analog (to optimize the power consumption).
- Generate peripheral initialization as a pair of ".c/.h" files per peripheral: The peripheral initialization code is configured to a separate ".c/.h" file when the code is generated. For example, "usart.c" and "usart.h" are generated for USART.
- Add all firmware libraries files into the project.
- Add only the necessary firmware libraries files into the project.



#### Linker Settings

- Minimum heap size and minimum stack size. Default values are 0x200 and 0x400, respectively. Users can modify the value when middleware stack is used.

#### MCU and Firmware Package

- Copy libraries into the project folder: Tick this check box, and libraries in the firmware package are copied automatically into the project folder when generating code.
- Package Version: Select the firmware package version. If the package is not setup, click on "Package Manager" to install.
- Package Manager: Install and manage firmware package. Refer to <u>Section 4.6</u> for details.
- Firmware Relative File: Display the path of the selected firmware package.

Finally, click on "OK" to generate user code and project of the selected IDE automatically. The generated project file structure is shown in Figure 22.

Organize 🔻 🗦 Open Inc	lude in	library ▼ Share with ▼ New folder			
▲ ★ Favorites	-	Name	Date modified	Туре	Size
🧮 Desktop		📔 libraries	5/10/2023 10:12 AM	File folder	
\rm Downloads	≡	📔 middlewares	5/10/2023 10:12 AM	File folder	
🕮 Recent Places		퉬 project	5/10/2023 10:12 AM	File folder	
		AT32F421C8T7_WorkBench.ATWP	5/10/2023 10:12 AM	ATWP File	
4 🥽 Libraries					
Documents					
🖻 🎝 Music					
Pictures					
🖻 🛃 Videos					
4 🜉 Computer					
4 💒 Local Disk (C:)					
⊳ 퉲 Keil_v5					
PerfLogs					
Program Files					
4 📗 Users					
Dublin	÷ .				

Figure 22. Project file directory

The file directory contains the "*libraries*" and "*middleware*" folders copied from the firmware package, as well as the generated project folder "*Project*". The "*Project*" contains a header file folder "*inc*", source code folder "*src*" and "*MDK\_V5*" (generated according to the selected toolchain/IDE). The *MDK\_V5* folder contains MDK project files that can be opened in Keil.





#### Figure 23. MDK\_V5 project files



### 4.5.2 Keep user code when re-generating

The AT32 Work Bench generated C code allows users to add custom code. The custom code needs to be inserted into the software-defined location and can be reserved for the next generation of C code. The software-defined location is as follows:

/\* add user code begin ..... \*/

/\* add user code end ..... \*/

When re-generating code, user codes in the software-defined position are reserved and not moved or renamed, while user codes not in the software-defined location are ignored and discarded. *Note 1: User-defined "add user code" tag is not supported. Custom code can be added to software-defined tags only.* 

Note 2: Code annotation cannot be the same as the software-defined "add user code" tag.

Take the main function in *main.c* as an example. User code can be added to the software-defined positions (in bold).

int main(void)	
{	
/* add user code begin 1 */	
/* add user code end 1 */	
/* system clock conlig. */	
wk_system_clock_config();	
/* nvic config_*/	
when when config():	
wk_nvic_conng(),	
/* add user code begin 2 */	
/* add user code end 2 */	
while(1)	
{	
/* add user code begin 3 */	
/* add user code end 3 */	
}	
}	



## 4.6 Package manager

Click on "File" – "Package Manager", and the Package Manager window pops up, as shown below.

Figure 24. Package Manager window

🜆 Package Manager	×			
Package Installation Location D:/WorkBench_BSP Brow				
Firmware Package	Creation Date			
✓ AT32F402_405	0			
AT32F402_405_Firmware_Library_V2.0.3	2024-01-18 11:42:32			
✓ AT32F403A_407	0			
AT32F403A_407_Firmware_Library_V2.1.7	2024-01-24 15:07:22			
AT32F413	•			
AT32F415	•			
AT32F421	•			
AT32F423	•			
AT32F425	•			
AT32A403A	•			
AT32WB415	•			
Install From Local Install From Network	Remove Close			

#### Package installation location

Select and confirm the package installation path. The software scans for the installed package at this location each time it starts up.

The installed package is stored in the "WK\_AT32xxx" directory, such as WK\_AT32F425. *Note: Users need to select the installation location after software update.* 

#### Install from local

Install the firmware package that has been downloaded. This local package (\*.ZIP) should be selected manually.

#### Install from Network

Download firmware package from Network and then set up automatically.



#### Remove

Tick the installed package and then remove it.

# 5 Revision history

Date	Version	Revision note
2024/12/30	V1.08	Updated some descriptions.
2024/12/25	V1.07	Updated some descriptions.
2024/10/28	V1.06	Updated some descriptions and figures.
2024/08/15	V1.05	Updated some descriptions and figures.
2024/03/05	V1.04	Updated some descriptions.
2024/01/26	V1.03	Added Section 4.6 "Package manager".
2023/11/28	V1.02	Updated some figures.
2023/09/08	V1.01	Updated some descriptions.
2023/05/09	V1.00	Initial release.

#### Table 4. Document revision history

#### **IMPORTANT NOTICE – PLEASE READ CAREFULLY**

Purchasers are solely responsible for the selection and use of ARTERY's products and services; ARTERY assumes no liability for purchasers' selection or use of the products and the relevant services.

No license, express or implied, to any intellectual property right is granted by ARTERY herein regardless of the existence of any previous representation in any forms. If any part of this document involves third party's products or services, it does NOT imply that ARTERY authorizes the use of the third party's products or services, or permits any of the intellectual property, or guarantees any uses of the third party's products or services or intellectual property in any way.

Except as provided in ARTERY's terms and conditions of sale for such products, ARTERY disclaims any express or implied warranty, relating to use and/or sale of the products, including but not restricted to liability or warranties relating to merchantability, fitness for a particular purpose (based on the corresponding legal situation in any unjudicial districts), or infringement of any patent, copyright, or other intellectual property right.

ARTERY's products are not designed for the following purposes, and thus not intended for the following uses: (A) Applications that have specific requirements on safety, for example: life-support applications, active implant devices, or systems that have specific requirements on product function safety; (B) Aviation applications; (C) Aerospace applications or environment; (D) Weapons, and/or (E) Other applications that may cause injuries, deaths or property damages. Since ARTERY products are not intended for the above-mentioned purposes, if purchasers apply ARTERY products to these purposes, purchasers are solely responsible for any consequences or risks caused, even if any written notice is sent to ARTERY by purchasers; in addition, purchasers are solely responsible for the compliance with all statutory and regulatory requirements regarding these uses.

Any inconsistency of the sold ARTERY products with the statement and/or technical features specification described in this document will immediately cause the invalidity of any warranty granted by ARTERY products or services stated in this document by ARTERY, and ARTERY disclaims any responsibility in any form.

© 2024 ARTERY Technology – All Rights Reserved