

CH32V20x 评估板说明及应用参考

版本：V1.0

<http://wch.cn>

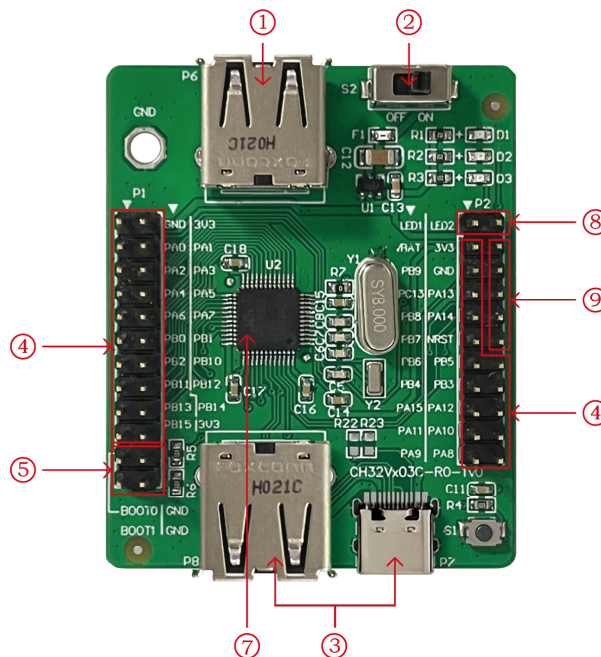
一、概述

本评估板应用于 CH32V20x 芯片的开发，IDE 使用 MounRiver 编译器，可选择使用板载或独立的 WCH-Link 进行仿真和下载，并提供了芯片资源相关的应用参考示例及演示。

二、评估板硬件

评估板的原理图请参考 CH32V20xSCH.pdf 文档

CH32V203评估板 \ CH32V203Evaluation



模块说明 \ Descriptions

- | | | |
|----------|-------------|----------|
| 1. 电源开关 | 4. MCU I/O口 | 7. MCU |
| 2. USB接口 | 5. 启动模式配置 | 8. LED |
| 3. USB接口 | 6. 复位按键 | 9. SWD接口 |

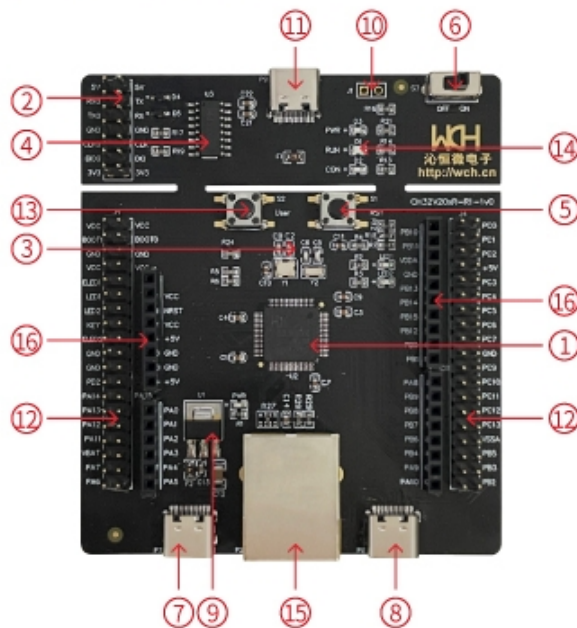
以上 CH32V203 评估板配有以下资源：

主板 - CH32V203EVT

1. 开关 S2：用于切断或连接外部 5V 供电或 USB 供电
2. USB 接口 P6：主芯片的 USB 通讯接口 PB6、PB7
3. USB 接口 P7、P8：主芯片的 USB 通讯接口 PA11、PA12
4. MCU I/O 口 P1、P2：主控 MCU 的 I/O 引出接口
5. 启动模式配置 P3：通过配置 B00T0/1 来选择芯片上电时的启动模式
6. 按键 S1：复位按键，用于外部手动复位主 MCU
7. 主控 MCU：CH32V203C8T6

8. LED：通过插针连接主芯片 IO 口进行控制
9. 调试接口：用于下载、仿真调试

CH32V203评估板 \ CH32V203 Evaluation



模块说明 \ Descriptions

- | | | | |
|----------------|--------------|---------------|----------------|
| 1、主控MCU | 5、复位按键 | 9、稳压芯片 | 13、USER按键 |
| 2、SDI&UART接口 | 6、电源开关 | 10、Download接口 | 14、WCH-Link指示灯 |
| 3、LED | 7、USB2.0全速接口 | 11、WCH-Link接口 | 15、网口 |
| 4、WCH-Link MCU | 8、USB2.0高速接口 | 12、MCU I/O | 16、ARDUINO接口 |

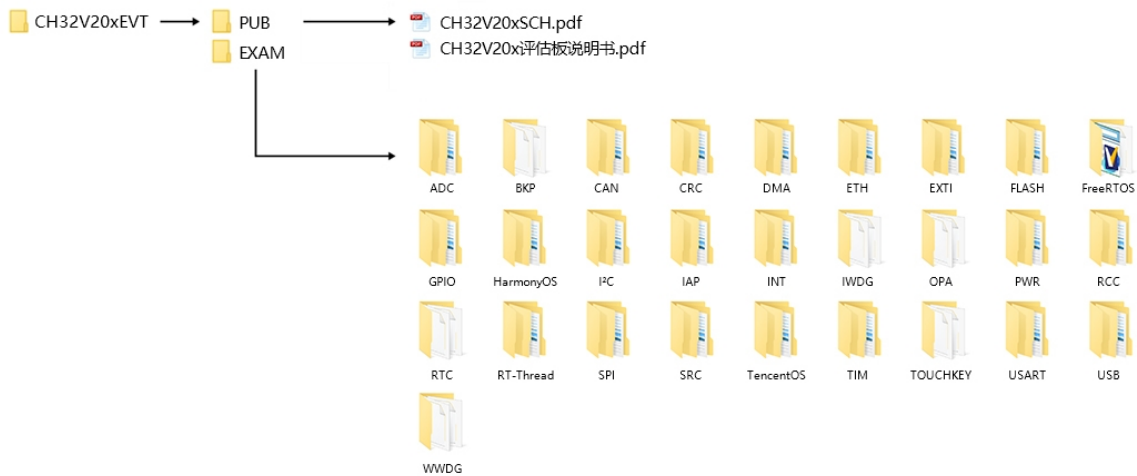
上图 CH32V203V 评估板配有以下资源：

主板 - CH32V203EVT

1. 主控 MCU：CH32V203RBT6
2. SDI&UART 接口：用于下载、仿真调试，需跳线选择是否使用板载 WCH-Link
3. LED：通过 J3 插针连接主控 MCU 的 IO 口进行控制
4. WCH-Link MCU：实现 WCH-Link 功能的 MCU
5. 按键 S1：复位按键，用于外部手动复位主控 MCU
6. 开关 S3：用于切断或连接外部 5V 供电或 USB 供电
7. USB type-C 接口 P7：连接主芯片 USB2.0 全速通信接口
8. USB 接口 P6：连接主芯片 USB2.0 高速通信接口
9. 稳压芯片 U1：用于实现将 5V 电压转成芯片可用的 3.3V 电源电压
10. Download 接口 J1：当 J1 跳线短接时，可用于实现 WCH-Link 固件更新
11. WCH-Link 接口：用于连接 PC 和 WCH-Link 功能模块
12. MCU I/O 口：主控 MCU 的 I/O 引出接口
13. USER 按键 S2：通过 J3 插针连接主控 MCU 的 IO 口进行按键控制
14. WCH-Link 指示灯：包括 D1、D2 和 D3 三个 LED 灯，指示 WCH-Link 运行状态
15. 网口：主芯片的网络通讯接口
16. ARDUINO 接口：方便连接 ARDUINO 接口的开发板

三、软件开发

3.1 EVT 包目录结构



说明：

PUB 文件夹：提供了评估板说明书、评估板原理图。

EXAM 文件夹：提供了 CH32V20x 控制器的软件开发驱动及相应示例，按外设分类。每类外设文件夹内包含了一个或多个功能应用例程文件夹。

3.2 IDE 使用 - MounRiver

下载 MounRiver_Studio，双击安装，安装后即可使用。（MounRiver_Studio 使用说明详见，路径：MounRiver\MounRiver_Studio\MounRiver_Help.pdf 和 MounRiver_ToolbarHelp.pdf）

3.2.1 打开工程

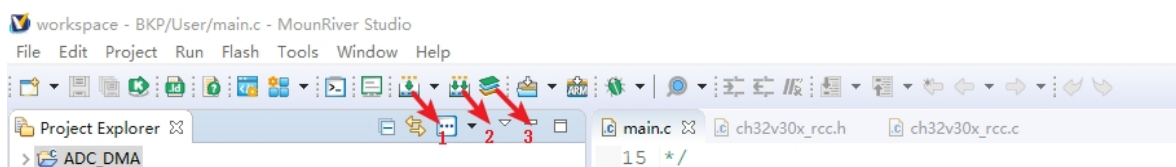
➤ 打开工程：

1) 在相应的工程路径下直接双击.wvproj 后缀名的工程文件；

2) 在 MounRiver IDE 中点击 File，点击 Load Project，选择相应路径下.project 文件，点击 Confirm 应用即可。

3.2.2 编译

MounRiver 包含三个编译选项，如下图所示：



编译选项 1 为增量编译，对选中工程中修改过的部分进行编译；

编译选项 2 为 ReBuild，对选中工程进行全局编译；

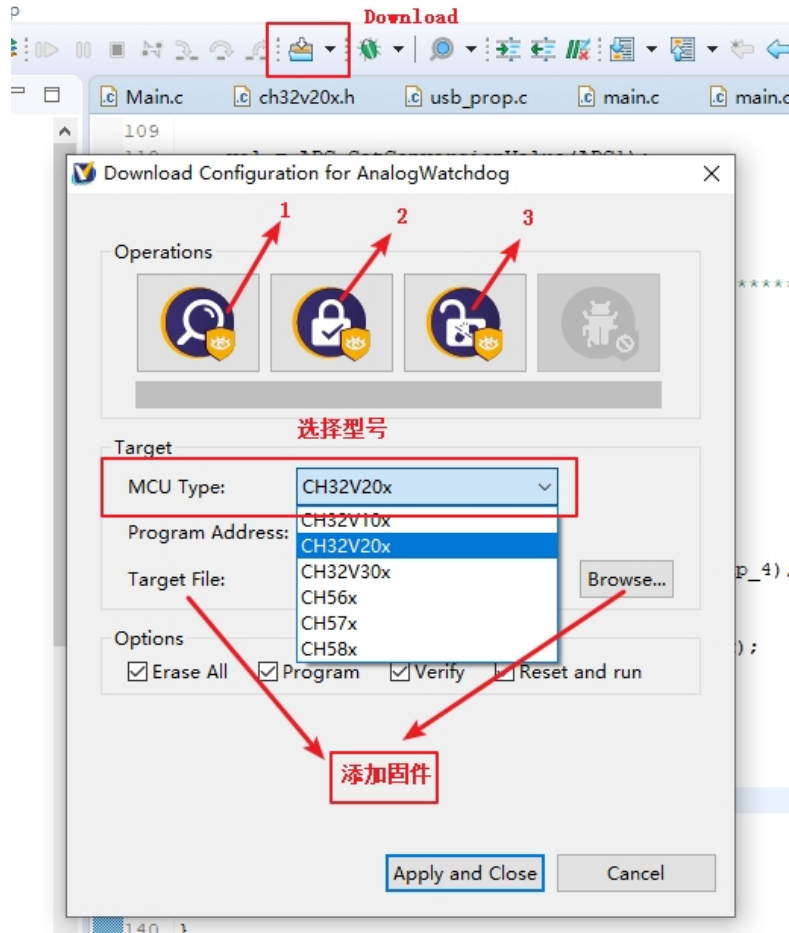
编译选项 3 为 All Build，对所有的工程进行全局编译。

3.2.3 下载/仿真

➤ 下载

1) 调试器下载

通过 WCH-Link 连接硬件（WCH-Link 使用说明详见，路径：MounRiver\MounRiver_Studio\WCH-Link 使用说明.pdf），点击 IDE 上 Download 按钮，在弹出的界面选择下载，如下图所示：



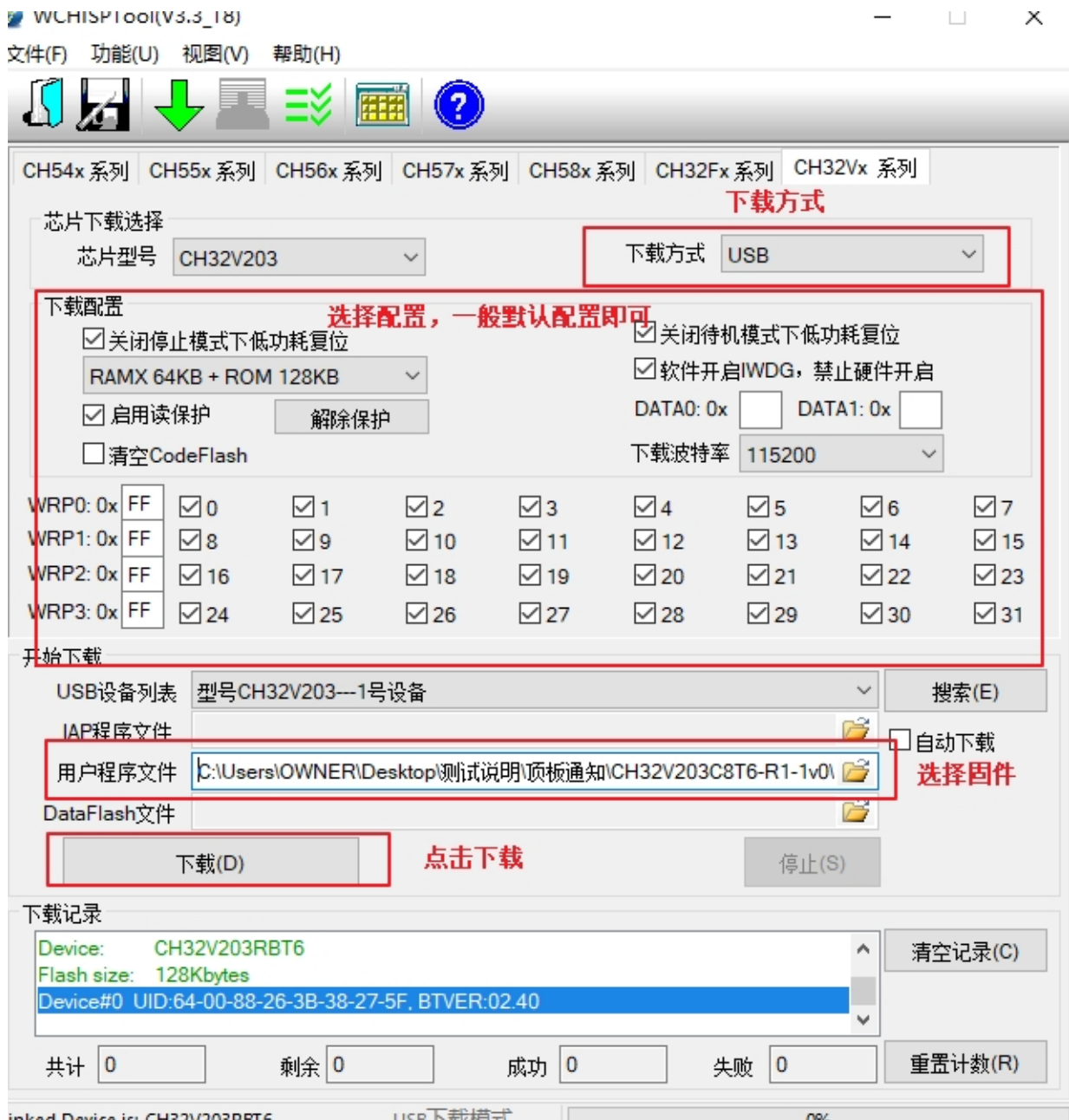
- 1 为查询芯片读保护状态；
- 2 为设置芯片读保护，重新上电配置生效；
- 3 为解除芯片读保护，重新上电配置生效；

2) WCHISPTool 下载

使用 WCHISPTool 工具对芯片进行下载，支持 USB 和串口两种下载方式。USB 管脚为 PA11（DM）、PA12（DP）或 PB6（DM）、PB7（DP），串口管脚为 PA9（TX）、PA10（RX）。下载流程为：

- (1) B00T0 接 VCC，B00T1 接地，通过串口或者 USB 连接 PC；
- (2) 打开 WCHISPTool 工具，选择相应下载方式，选择下载固件，勾选芯片配置，点击下载；
- (3) B00T0 接地，重新上电，运行 APP 程序。

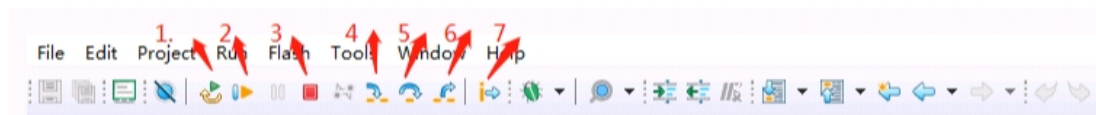
WCHISPTool 工具界面如图所示：



➤ 仿真

1) 工具栏说明

点击菜单栏的调试按键进入下载，见下图所示，下载工具栏



详细功能如下：

- (4) 复位 (Restart)：复位之后程序回到最开始处。
- (5) 继续：点击继续调试。
- (6) 终止：点击退出调试。
- (7) 单步跳入：每点一次按键，程序运行一步，遇到函数进入并执行。
- (8) 单步跳过：跳出该函数，准备下一条语句。
- (9) 单步返回：返回所跳入的函数

(10) 指令集单步模式：点击进入指令集调试（需与 4、5、6 功能配合使用）。

2) 设置断点

双击代码左侧可设置断点，再次双击取消断点，设置断点如下图所示：

```

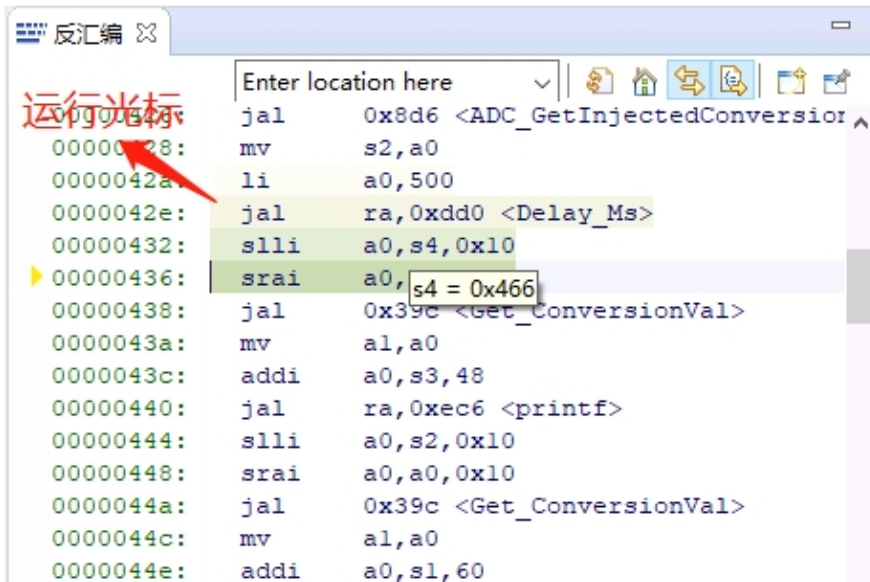
133 int main(void)
134 {
135     ul6 adc_val;
136     ul6 adc_jval;
137     Delay_Init();
138     USART_Printf_Init(115200);
139     printf("SystemClk:%d\r\n", SystemCoreClock);
140
141
142     ADC_Function_Init();
143     printf("CalibrattionValue:%d\n", Calibrattion_Val);
144
145     while(1)
146     {

```

3) 界面显示

(1) 指令集界面

点击指令集单步调试可进入指令调试，以单步跳入为例，点击一次，可运行一次，运行光标会发生移动，以查看程序运行，指令集界面如下图所示：



```

反汇编
Enter location here
00000434: jal    0x8d6 <ADC_GetInjectedConversionValue>
00000438: mv     s2, a0
00000428: li     a0, 500
0000042e: jal    ra, 0xdd0 <Delay_Ms>
00000432: slli  a0, s4, 0x10
00000436: srai  a0, s4, 0x466
00000438: jal    0x39c <Get_ConversionVal>
0000043a: mv     a1, a0
0000043c: addi  a0, s3, 48
00000440: jal    ra, 0xec6 <printf>
00000444: slli  a0, s2, 0x10
00000448: srai  a0, a0, 0x10
0000044a: jal    0x39c <Get_ConversionVal>
0000044c: mv     a1, a0
0000044e: addi  a0, s1, 60

```

(2) 程序运行界面

可与指令集单步调试配合使用，仍以以单步跳入为例，点击一次，可运行一次，运行光标会发生移动，以查看程序运行，程序运行界面如下图所示：

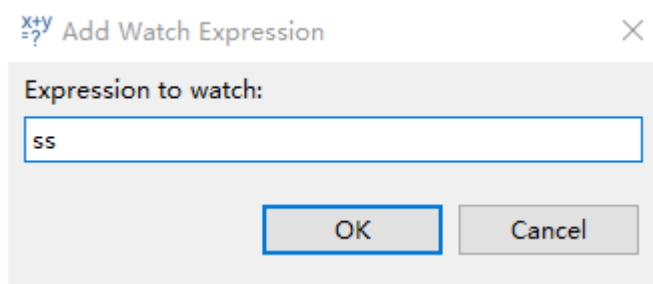
```

143     printf("CalibrattionValue:%d\n", Calibrattion_Val);
144
145     while(1)
146     {
147         ADC_SoftwareStartConvCmd(ADC1, ENABLE);
148         while( !ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC) );
149         adc_val = ADC_GetConversionValue(ADC1);
150         adc_jval = ADC_GetInjectedConversionValue(ADC1, ADC_InjectedChannel_1);
151         Delay_Ms(500);
152         printf( "val:%04d\r\n", Get_ConversionVal(adc_val));
153         printf( "jval:%04d\r\n", Get_ConversionVal(adc_jval));
154         Delay_Ms(2);
155     }
156 }
157

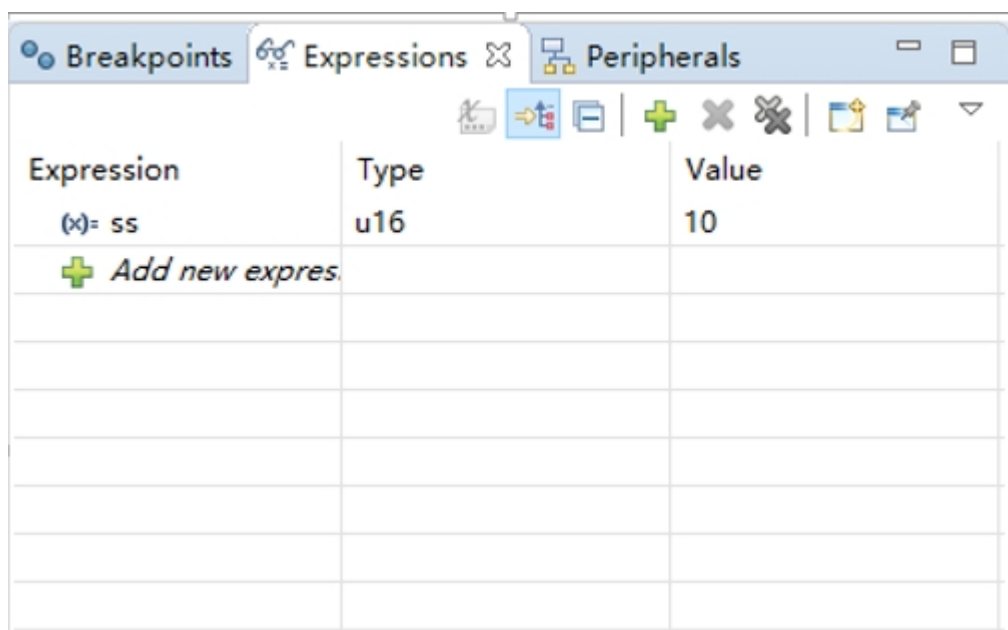
```

4) 变量:

鼠标悬停在源码中变量之上会显示详细信息，或者选中变量，然后右键单击 add watch expression

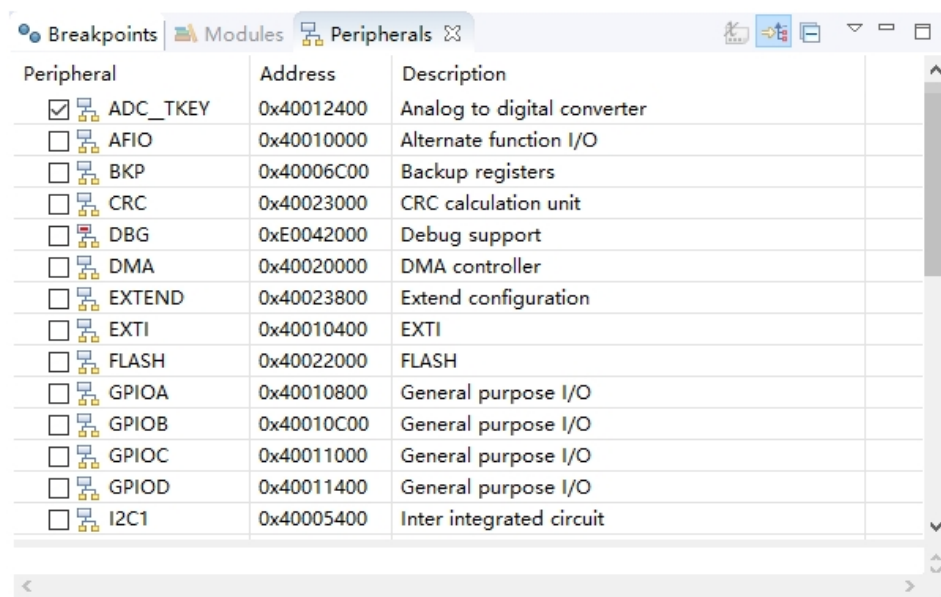


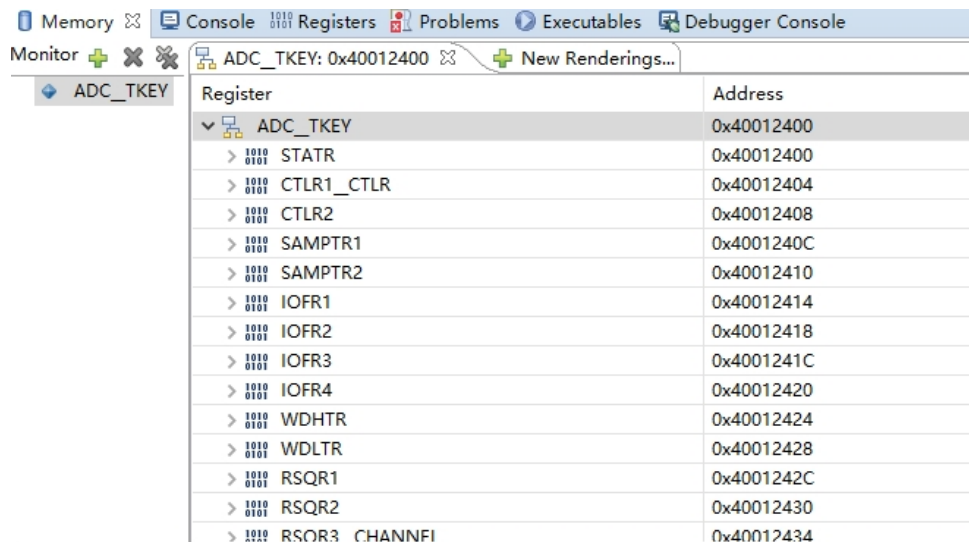
填写变量名，或者直接点击 OK，将刚才选中的变量加入到弹出的：



5) 外设寄存器

在 IDE 界面左下角 Peripherals 界面显示有外设列表，勾选外设则在 Memory 窗口显示其具体的寄存器名称、地址、数值。





注明：(1)调试时，点击右上角图标可进入原始界面。

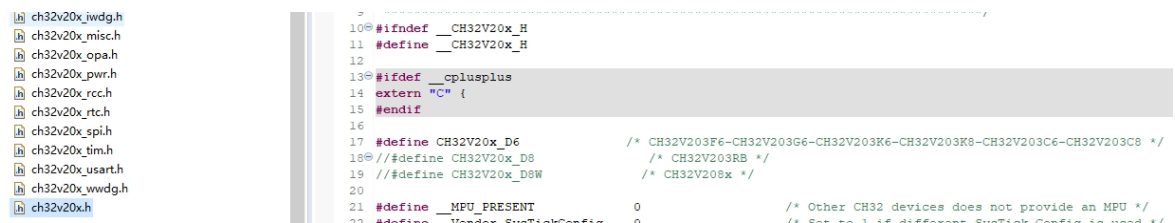


(2)有关文档进入编译器，点击 F1 可进入帮助文档，可查看详细说明。

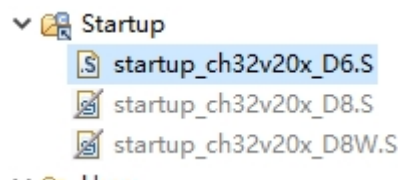
➤ 工程芯片选型

在历程工程中有多种芯片选型，以 CH32V20x 开发板所用芯片 CH32V203C8T6 为例进行工程芯片选择编译，已实现不同外设功能，步骤如下：

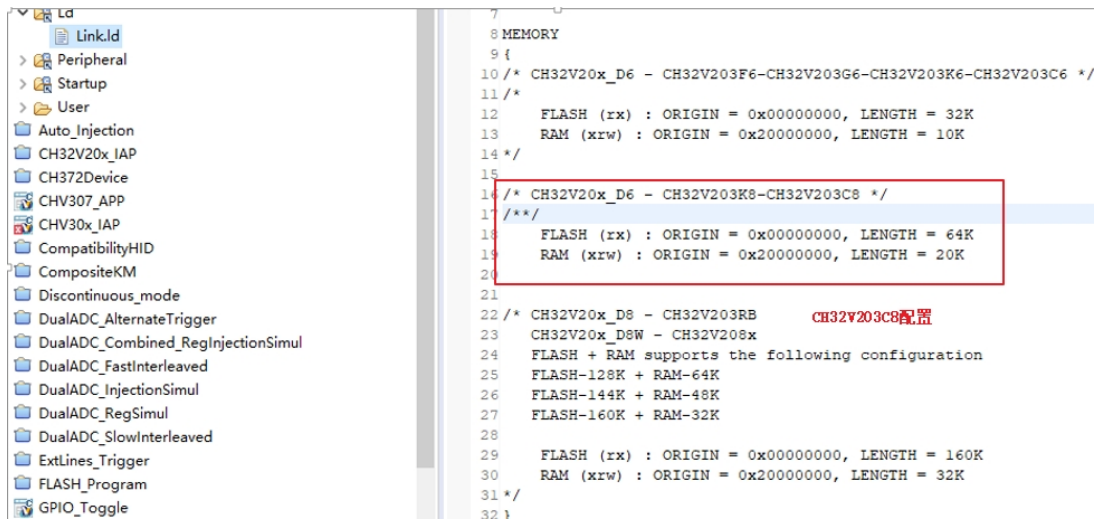
- 1) 点击 Peripheral → inc 文件夹下的 ch32v30x.h 文件进行芯片类型的勾选，如下图所示，因选用芯片类型为 CH32V203C8T6，因此选用 CH32V20x_D6。（注：不同的芯片根据注释，选用不同的声明）。



- 2) 选择启动文件，点击 Startup 文件夹，选用定义所对应的启动文件，如下图所示，因步骤 1 选用 CH32V20x_D6，因此选用 startup_ch32v20x_D6.S 文件（注：根据步骤 1 中的不同的声明，选用不同的启动文件。）



- 3) 更改起始文件内存，点击 LD 文件，如下图所示，依据芯片类型选用不同的内存配置，如使用芯片类型为 CH32V203C8T6，因此选用 CH32V20x_D6 的 FLASH 与 RAM 配置



注明: 在工程不使用的文件中, 如在文件夹内文件点击删除后, 会导致文件直接删除至回收网站, 再次使用时需重新编译才可, 因此对无使用的文件建议使用参与/排除编译功能, 以对 Startup 文件夹中编译/排除进行简要说明。首先选用工程点击右键, 选用参与/排除编译功能按键, 以图中 startup_ch32v20x_D8.S 文件为例, 如工程状态为排除编译, 点击本功能即可为参与编译。反正, 如工程状态为参与编译, 点击本功能即可为排除编译。(注: 文件夹也可适用)。

