

# 开发环境的搭建

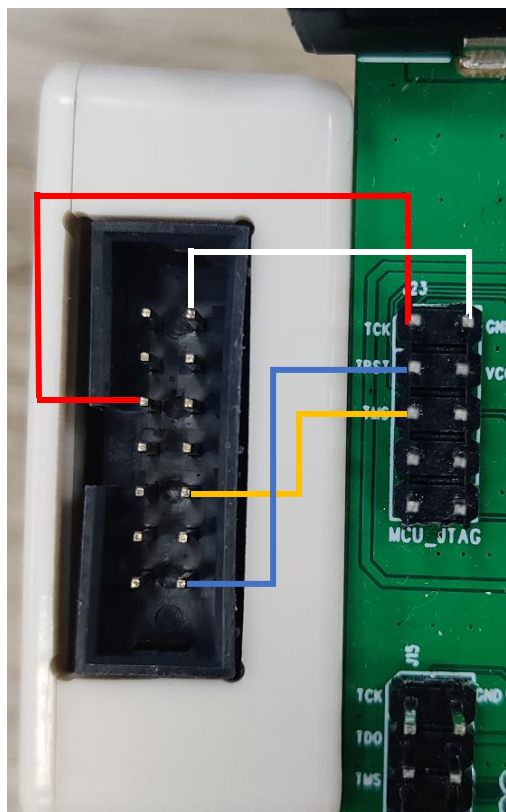
## 历史版本

版本号	作者	日期	修改内容
V1.0		2020.09.01	初版建立

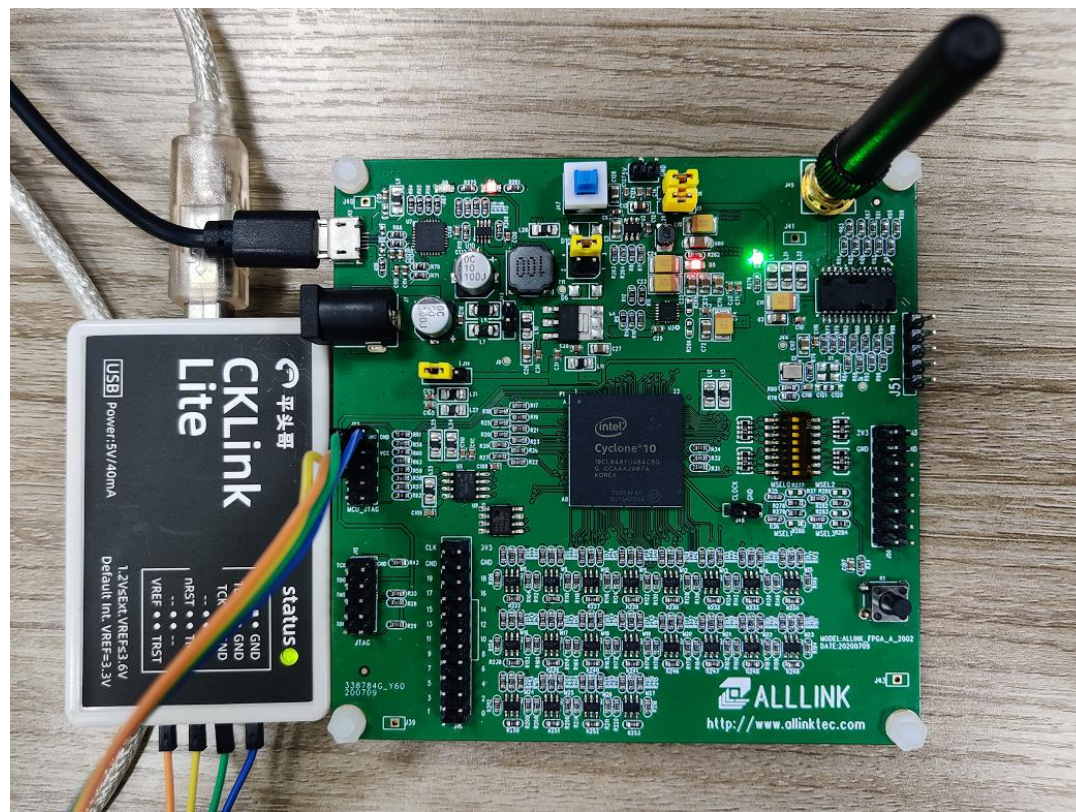
# 目录

1.	开发板接线 .....	4
2.	CDK软件安装与使用 .....	5
	2.1打开一个工程 .....	6
	2.2CDK软件界面介绍 .....	7
	2.3工程目录介绍.....	8
	2.4工程编译 .....	9
3.	Debug调试 .....	10
	3.1基本调试步骤.....	11
	3.2局部变量查看.....	12
	3.2查看当前变量值 .....	13
	3.2查看内存值 .....	14
4.	烧录工具.....	15
	4.1SOP16芯片烧录示例 .....	16

## 1.0 开发板接线



红色: TCK  
蓝色: TRST  
黄色: TMS  
白色: GND



1. 请根据所提供的《AK80X开发板使用说明》文档将开发板接好对应的导线，如上图
2. 若使用不同的型号的仿真器请根据具体引脚进行连接

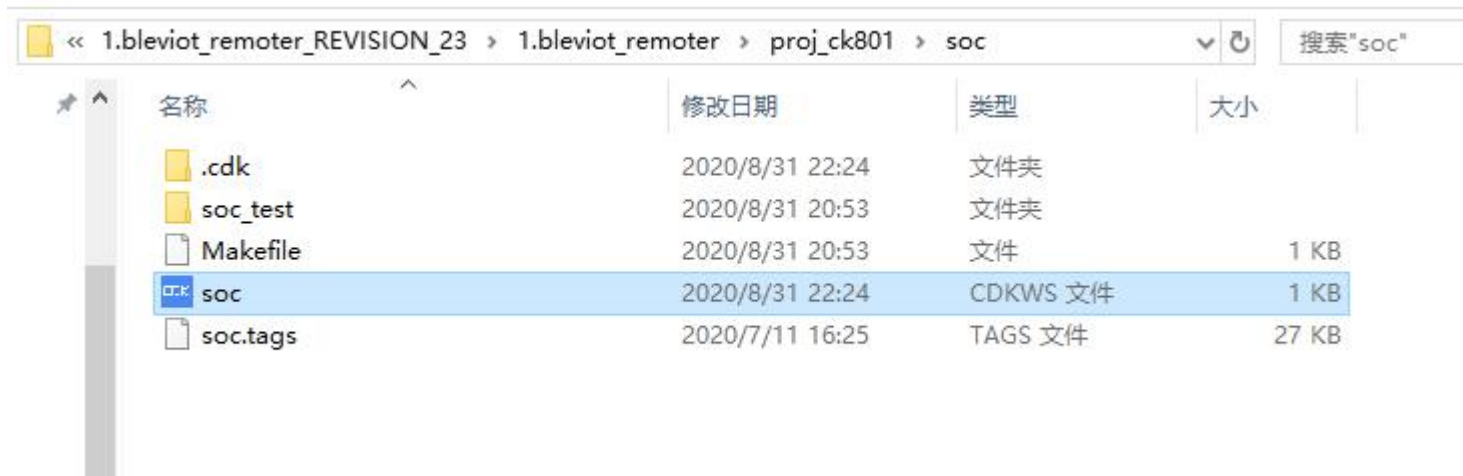
## 2.0 CDK软件安装

名称	修改日期	类型	大小
utils	2018/3/7 13:50	文件夹	
0x0409	2016/3/21 12:04	配置设置	22 KB
data1	2020/5/15 18:09	WinRAR 压缩文件	937 KB
data1.hdr	2020/5/15 18:09	HDR 文件	1,201 KB
data2	2020/5/15 18:11	WinRAR 压缩文件	512,281 KB
ISSetup.dll	2018/1/15 15:15	应用程序扩展	1,587 KB
layout.bin	2020/5/15 18:11	BIN 文件	1 KB
setup	2020/5/15 18:09	应用程序	919 KB
setup	2020/5/15 18:09	配置设置	3 KB
setup.inx	2020/5/15 18:09	INX 文件	274 KB



1. 获取到安装包后，将其解压得到左边的文件列表，双击“setup”，根据提示进行安装；
2. 安装成功后桌面出现“CDK”软件的图标，双击该图标即可运行CDK软件。

## 2.1 打开一个工程



1.获取到相应的SDK压缩包，将其解压后得到以上文件列表。

2.找到解压包路径

“X:\XX\1.bleviot\_remoter\_REVISION\_23\1.bleviot\_remoter\_REVISION\_23\1.bleviot\_remoter\proj\_ck801\soc”

3.双击名称为“soc”文件即可使用CDK软件打开该工程

**\*注：存放工程的路径名称不能有中文**

## 2.2 CDK软件界面





## 2.3 工程目录介绍



1.不同的SDK目录文件有所差异，请以实际收到的SDK为准



## 2.4 工程编译

The screenshot displays an IDE interface for a project named 'soc\_test'. The Project View on the left shows a tree structure with 'soc\_test' expanded to 'app'. A context menu is open over 'app', with 'Rebuild' highlighted. The main editor shows the source file 'app\_main.c' with a comment block containing a table:

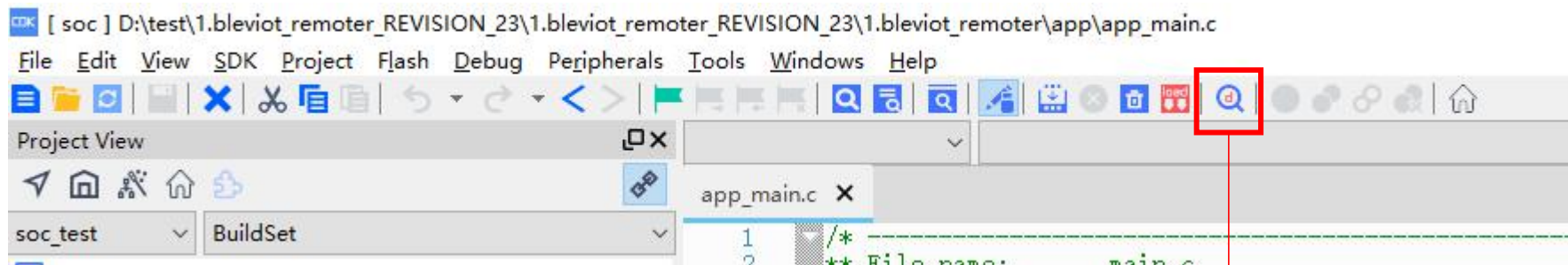
```
1  /*
2  ** File name:      main.c
3  ** Brief:
4  ** COPYRIGHT(C) ZXW 2020
5  **
6  ** Module Information:
7  **
8  ** Revision History
9  **
10 ** Date      Version  Action      Author
11 **
12 ** 2019-5-17  1.0      Created
13 **
14 **
15 */
```

The Output View at the bottom shows the compilation process:

```
compiling key_process.c...
compiling rmt_lmp_fun.c...
compiling ble_viot.c...
linking...
generating hex file...
generating bin file...
generating asm file...
size of target:
text  data  bss  dec  hex filename
7692  16   124  8832  2280 D:/test/1.bleviot_remoter_REVISION_23/1.bleviot_remoter_REVISION_23/1.bleviot_remoter/proj_ck801/soc/soc_test/Obj/soc_test.elf
checksum value of target: 0x34999F7F (559,632)
make[1]: Leaving directory 'd:/test/1.bleviot_remoter_REVISION_23/1.bleviot_remoter_REVISION_23/1.bleviot_remoter/proj_ck801/soc/soc_test'
====0 errors 6 warnings, total time : 4s793ms====
```

Red arrows point from the 'Rebuild' menu item to the text '选择“Rebuild”,将整个工程编译' and from the '6 warnings' in the output to the text '编译输出的6个警告为正常结果'.

## 3.0 Debug调试

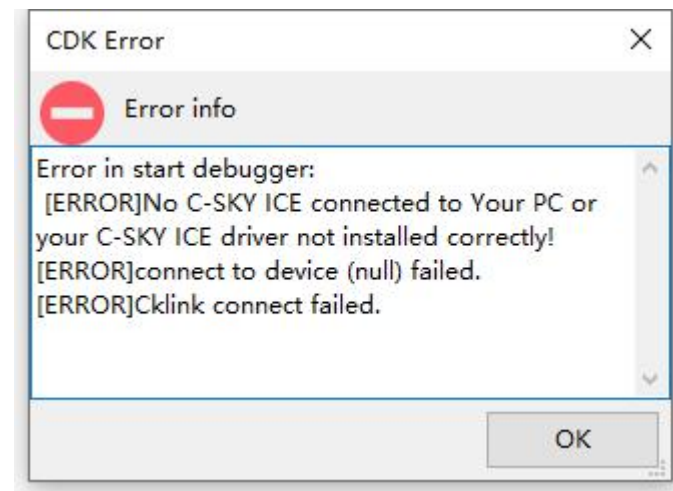


点击小放大镜进入Debug

进入Debug之前请先确认：

- 1.仿真器和开发板供电正常
- 2.仿真器和开发板的接线正确

否则会提示进入debug模式失败。



## 3.1 调试基本步骤



放置断点

清除所有断点

复位

运行

进入函数

单步运行

跳出函数

## 3.2 局部变量查看

The screenshot shows a debugger interface with a code editor and a Locals window. The code editor displays the following code:

```
43
44 void sys_soft_reset( uint8_t module )
45 {
46     uint32_t    reg_val = 0;
47
48     reg_val = read_reg( TOP_SOFT_RESET_REG );
49
50     reg_val |= module;
51     write_reg(TOP_SOFT_RESET_REG, reg_val);
52     reg_val &= ~module;
53     write_reg(TOP_SOFT_RESET_REG, reg_val);
54 }
55
56 void sys_set_clock( cpu_clock_sel_t clock )
57 {
58     uint32_t    reg_val = 0;
59
60     reg_val = read_reg( TOP_MODULE_MODE_REG );
61     reg_val &= ~TOP_CPU_CLK_SEL_MASK;
62     reg_val |= clock << TOP_CPU_CLK_SEL_SHIFT;
63     write_reg( TOP_MODULE_MODE_REG, reg_val );
64 }
```

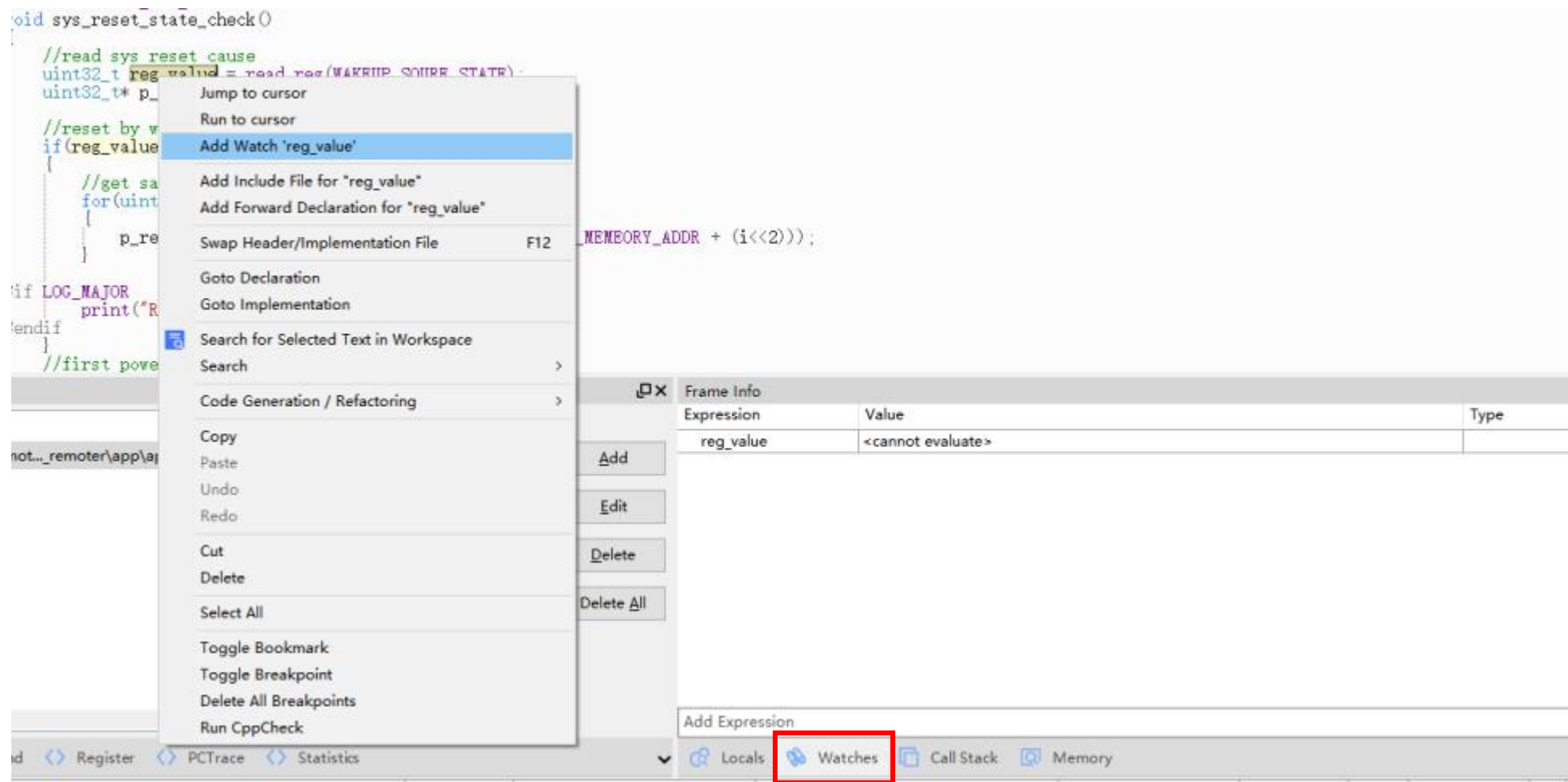
The Locals window shows the following variables:

Name	Value	Type
clock	<optimized out>	cpu_clock_sel_t
reg_val	24	uint32_t

The Locals window also includes buttons for Add, Edit, Delete, and Delete All. The Locals tab is highlighted in the bottom toolbar.

1. 执行到该函数之后，可在Locals窗口可查看该函数局部变量值和形参值

### 3.3 查看当前变量值



1.选中想要观察的变量，右键--Addwatch “xxx”即可添加到变量观察窗口，如上图所示

## 3.4查看内存值

The screenshot shows a debugger window titled "Frame Info" with a "Memory" tab selected. The "Address" field is set to "0x50000040". The "Number of columns" is set to "4". The "Width Adapt" button is visible. The memory view displays a table with columns labeled "1", "2", "3", and "4". The first column shows addresses from 0x50000040 to 0x50000110. The other columns show values, mostly 0x00000000. Red arrows point from the first column to the right, with labels: 0x5000004C, 0x50000048, 0x50000044, and 0x50000040. The "Memory" tab is highlighted with a red box.

	1	2	3	4
0x50000040	0x00000000	0x00000000	0x00000000	0x00000000
0x50000050	0x00000000	0x00000000	0x00000000	0x00000000
0x50000060	0x00000000	0x00000000	0x00000000	0x00000000
0x50000070	0x00000000	0x00000000	0x00000000	0x00000000
0x50000080	0x00000000	0x00000000	0x00000000	0x00000000
0x50000090	0x00000000	0x00000000	0x00000000	0x00000000
0x500000a0	0x00000000	0x00000000	0x00000000	0x00000000
0x500000b0	0x00000000	0x00000000	0x00000000	0x00000000
0x500000c0	0x00000000	0x00000000	0x00000000	0x00000000
0x500000d0	0x00000000	0x00000000	0x00000000	0x00000000
0x500000e0	0x00000000	0x00000000	0x00000000	0x00000000
0x500000f0	0x00000000	0x00000000	0x00000000	0x00000000
0x50000100	0x00000000	0x00000000	0x00000000	0x00000000
0x50000110	0x00000000	0x00000000	0x00000000	0x00000000

1.例如想查看0x5000040这个地址的内容，只需在Address中填入0x5000040即可定位到该地址的位置；地址是以4个字节递增的，内存查看器中的排列顺序如左图所示。



## 4.0 烧录工具



burner\_demo\_v  
2.8



1. 获取到烧录工具压缩包，解压后得到“burner\_demo\_v2.8.exe”，双击即可打开烧录工具进行配置
2. 对烧录软件的配置请参考提供的《AK801芯片烧录器使用说明》文档进行操作



## 4.1 SOP16烧录示例

