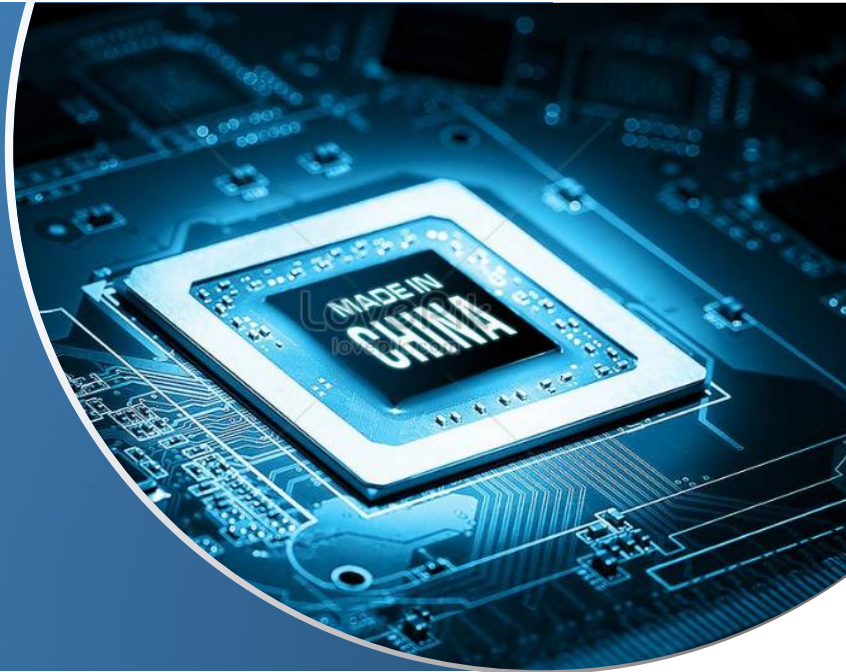


CDK IDE使用介绍

APTCHIP

APTCHIP MICROELECTRONICS



01

CDK介绍

02

CDK调试介绍

03

CDK 工程设置介绍

04

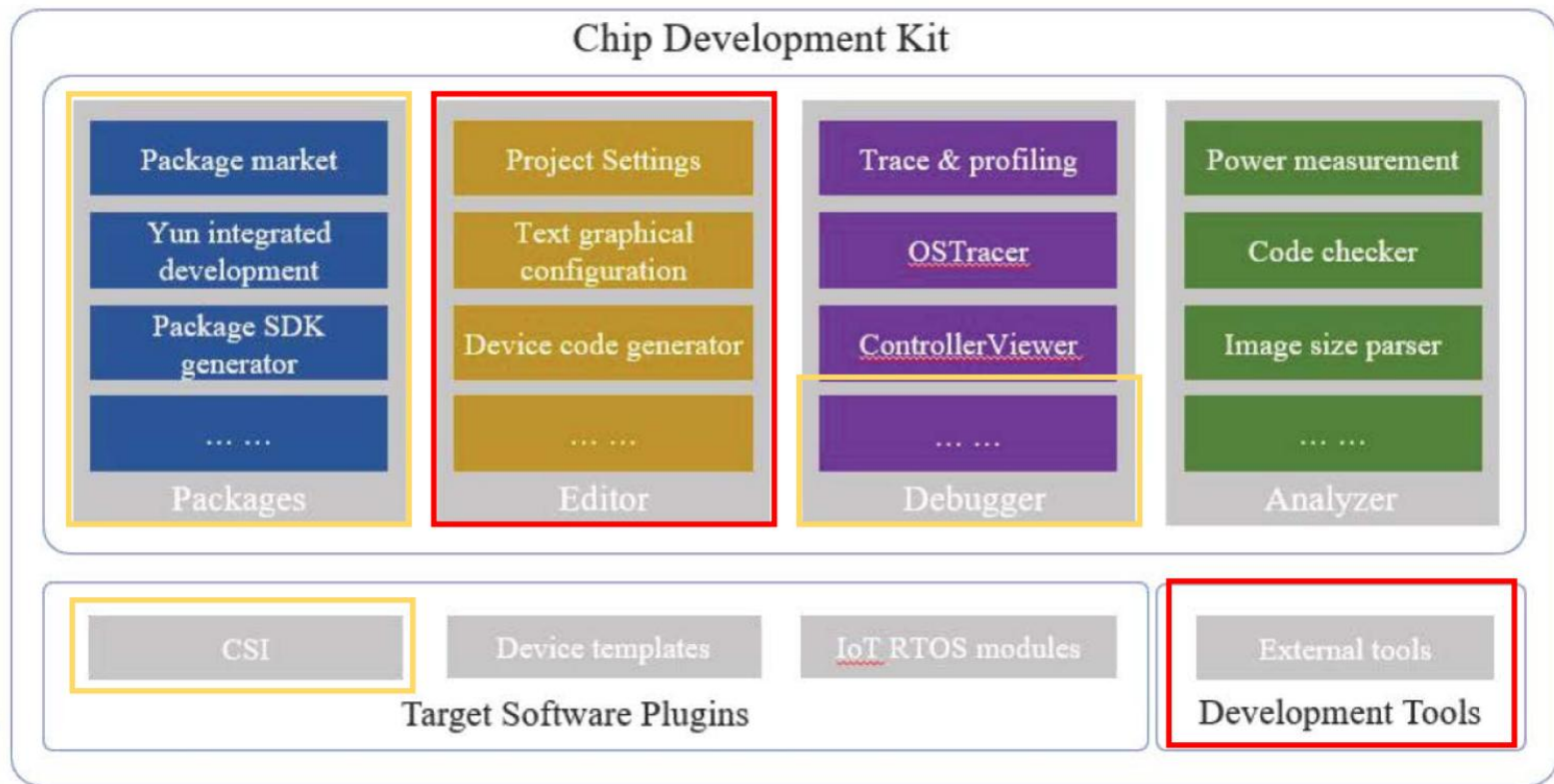
CDK使用常见问题介绍

01

CDK介绍

APT芯片开发环境-CDK

- 剑池CDK：面向IoT开发领域的集成开发环境，目前已支持平头哥自研指令集C-SKY 架构和RISC-V 架构的芯片的开发
- CDK 内部（welcome page）自动对接芯片开放平台，可以直接获取芯片开放平台上的开发资源



CDK Logo

界面-编辑模式

The screenshot displays the APT32F110x IDE interface in edit mode. The interface is divided into several key sections:

- 菜单栏 (Menu Bar):** Located at the top, it includes File, Edit, View, SDK, Project, Flash, Debug, Peripherals, Tools, Windows, and Help.
- Debug栏 (Debug Bar):** A toolbar below the menu bar containing various debugging icons.
- 工程窗口 (Project View):** On the left side, it shows a tree view of the project structure, including folders like 'arch', 'doc', 'drivers', 'FWlib', 'inc', and 'source', along with source files like 'apt32f110x_initial.c' and 'main.c'.
- 程序窗口 (Code Editor):** The central area displays the source code for 'main.c', showing C++ code for BT module initialization and a main loop with UART debugging.
- 大纲窗口 (Outline View):** On the right side, it provides a hierarchical overview of the code, listing include files, global functions and variables, function prototypes, and macros.
- log窗口 (Output View):** At the bottom, it shows the compilation and linking process, including the command line and the resulting ELF file size and checksum.

```
95 BT_Start(BT1);
96 GPIO_Set_Value(GPIOA0,11,0);
97 }
98 }
99 }
100 else
101 {
102     Key_Map_bk=0;
103     BT_Stop(BT1);
104     GPIO_Set_Value(GPIOA0,11,1);
105 }
106 }
107 //main
108 //*****
109 int main(void)
110 {
111     delay_nms(10000);
112     APT32F110x_init();
113     tk_uart_init();
114     GPIO_Init(GPIOA0,11,0);
115     GPIO_Set_Value(GPIOA0,11,1);
116 #ifdef _DEBUG
117     tk_uart_init();
118 #endif
119 while(1)
120 {
121     SYSCOM_IWDGNT_Reload();
122     tk_prgm();
123     tk_test();
124 #ifdef _DEBUG
125     //uart_send_tk_Sampling();
126     //uart_send_tk_offset();
127     uart_send_tk_keymap();
128     //uart_send_tk_trigg();
129 #endif
130     //....
131     //....
132 }
133 }
134 //***** (C) COPYRIGHT 2022 APT Chip *****END OF FILE*****
```

```
csky-elfabiv2-gcc -o Obj/Release_APT32F110.elf -Wl,--ckmap=../Lst/Release_APT32F110.map @apt32f110x.txt -mcpu=ck801 -nostartfiles -Wl,--gc-sections -T"../ckcpu.ld" -Wl,--start-group -Wl,--end-group
csky-elfabiv2-objcopy -O ihex ../Obj/Release_APT32F110.elf ../Obj/Release_APT32F110.ihex
csky-elfabiv2-objdump -S ../Obj/Release_APT32F110.elf > ../Lst/Release_APT32F110.asm
size of target:
text  data  bss  dec  hex filename
11710  158  1826  13704  3588  ./Obj/Release_APT32F110.elf
checksum value of target:0xE23F04ED (282,732)
Obj/Release_APT32F110.elf is modified at: 2022/8/8 14:32:52
====0 error(s), 0 warning(s), total time : 4s796ms====
```

界面-debug模式

The screenshot displays the APT32F110x IDE interface in debug mode. The main window is divided into several panes:

- Project View (工程窗口):** Located on the left, showing the project structure for 'apt32f110x', including folders like 'arch', 'doc', 'drivers', 'FWlib', 'inc', and 'source', and source files like 'apt32f110x_initial.c', 'apt32f110x_interrupt.c', 'apt32f110x_test.c', 'ckcpu.ld', and 'main.c'.
- Disassemble (反汇编窗口):** Located at the top center, showing assembly code for the 'main' function, including instructions like 'push', 'lwr', 'bsr', 'lwr', 'bsr', and 'movi' with their respective addresses and operands.
- Source Code (程序窗口):** Located in the center, showing the C source code for 'main.c'. The code includes headers, a 'main' function, and a loop that calls 'SYSCON_IMDCNT_Reload()', 'tk_prgm()', 'tk_test()', and various UART send functions.
- Frame Info (信息窗口):** Located on the right, displaying a table of stack frames. The table has columns for 'Expression', 'Value', and 'Type'. The current frame is 'main' at address 0x40000000.
- Output View (log窗口):** Located at the bottom center, showing the output of the build process, including the command used to compile the ELF file and the resulting checksum and size information.
- Serial Pane (串行窗口):** Located on the right, showing the serial output of the program, with a 'Send:' field and a dropdown menu for the serial port.
- Debugger (Debugger窗口):** Located at the bottom left, showing the debugger controls, including buttons for 'Breakpoints', 'Threads', 'Output', 'Command', 'PCTrace', and 'Statistics'.

CDK菜单栏介绍

File菜单

集中了对文件、程序本身的基本操作；

Edit菜单

用于文本编辑相关操作、用户CDK的基本配置；

View菜单

包含了视图相关的设置；

SDK菜单

包含了SDK组建的全部入口；

Project菜单

包含了工作空间、工程管理相关配置；

Flash菜单

包含了Flash的基本操作选项；

Debug菜单

包含了调试的基本命令及断点；

Tools菜单

介绍了CDK内置的工具，用于方便用户的开发；

Windows菜单













包含了CDK布局操作；

Help菜单

用于帮助用户详细了解CDK；
















CDK工具栏按钮介绍1

-  新建文件：点击之后，文本编辑窗口中会弹出一个未命名、未保存的文件；
-  打开文件：点击弹出文件选择对话框；
-  重载文件：点击会将文本编辑窗口中正在编辑的文本重载；
-  保存文件：将文本编辑窗口中正在编辑的文件保存
-  关闭文件：将文本编辑窗口中开启的文件关闭；
-  剪切：将文本编辑窗口中选中的文本剪切到黏贴板；
-  复制：将文本编辑窗口中选中的文本复制到黏贴板；
-  粘贴：将黏贴板的内容复制在文本编辑窗口中光标的位置；
-  恢复之前的操作：将正在编辑的文本恢复到前一个编辑状态；
-  重做之前操作：将正在编辑的文本重新恢复到之前的操作；
-  跳转至前一位置：将光标的位置设置为前一个跳转的位置；
-  跳转至后一位置：将光标的位置设置为后一个跳转的位置；



CDK工具栏按钮介绍2
















-  设置、删除光标所在行的书签;
-  跳转至下一个书签; 将光标位置设置在下一个书签的位置;
-  跳转至上一个书签; 将光标位置设置在上一个书签位置;
-  删除全部书签;
-  文件内查找替换; 在正在编辑的文件中查找、替换字符;
-  全局查找替换; 在整个工作空间中查找、替换字符;
-  查找全局资源; 在整个工作空间中查找特定的文件、符号;
-  高亮选中项; 高亮当前文本中, 与选中的字符相同的字符;
-  编译当前打开的文件;
-  工程构建按钮; 对工作空间中的活动工程进行构建;
-  工程重新构建按钮; 对工作空间中的活动工程进行重新构建;
-  停止构建; 停止正在进行的构建操作;
-  清空工程; 清空工作空间中的活动工程的构建内容;

02

CDK调试介绍



CDK工程调试按钮介绍

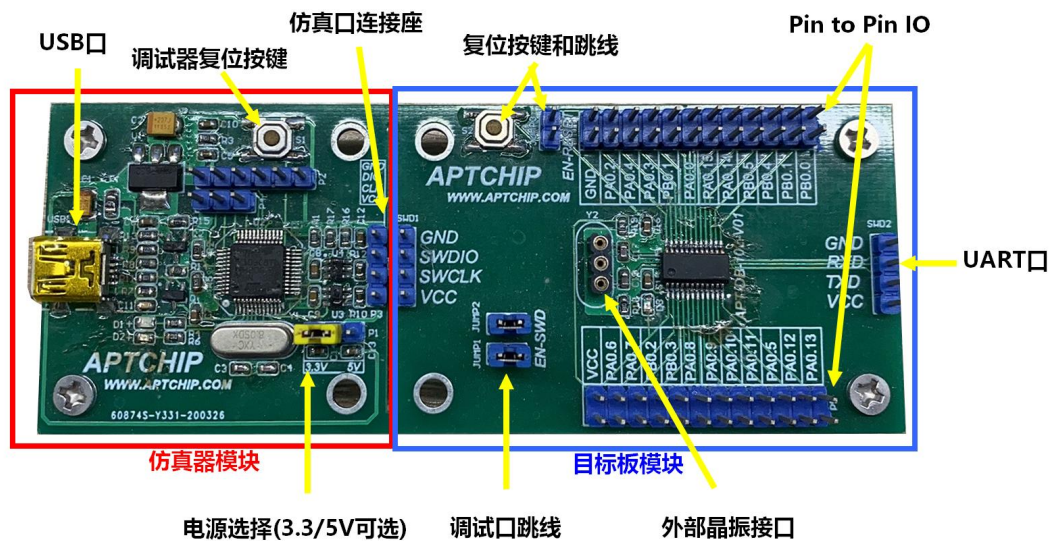
-  启动调试，CDK进入调试状态，停止本次调试操作，CDK切换到编辑状态；停止调试；
-  启动调试，不执行Load或者Flash擦除/烧写/验证流程；
-  Flash下载；点击之后，会根据用户的Flash配置进行Flash下载操作；
-  添加断点；
-  Enable所有断点；
-  Disabled所有断点；
-  删除所有断点；
-  调试系统复位；按照下拉菜单中选中的Reset方式，对调试目标进行复位操作；Soft Reset的参数需要在调试配置中设置；
-  运行程序；
-  停止程序运行；将正在运行的程序停止下来；
-  显示当前位置；点击之后，文本编辑以及反汇编会显示当前PC所在的位置；
-  C语言级单步操作；
-  C语言级下一行操作；
-  C语言级跳出当前函数操作；
-  下拉菜单按钮，快速定位调试器窗格、符号查看窗口；

开发硬件环境搭建

调试硬件

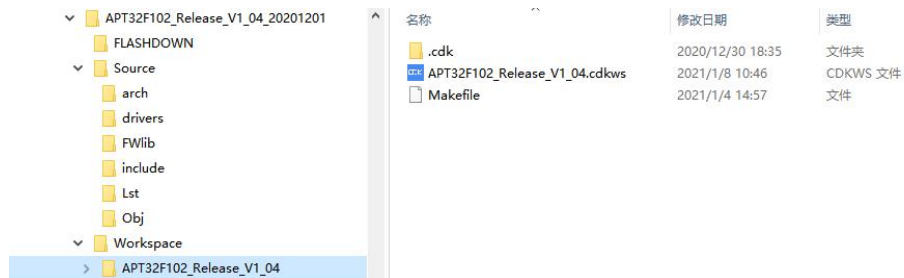
方式1：APTlink+目标板，AB口USB线，

方式2：APT学习板，mini USB线



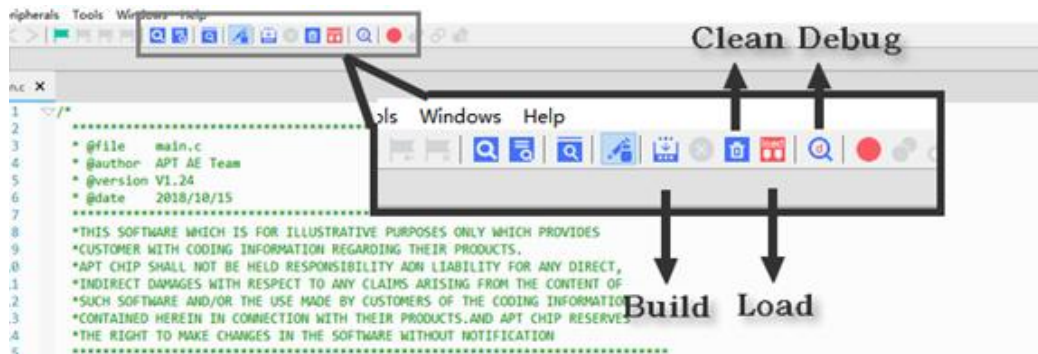


1.连接好硬件后，使用CDK打开APT库文件包



2.开启工程后，可以在CDK内显示整个工程

3.仿真常用按钮

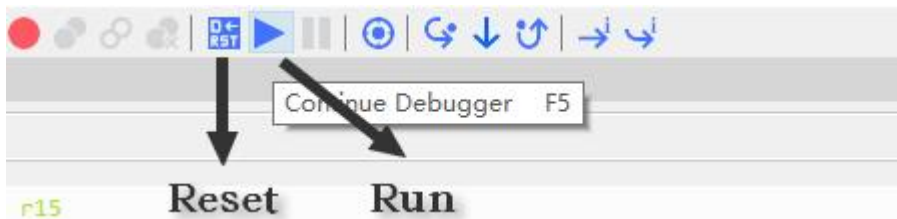


4.点击  进入调试模式

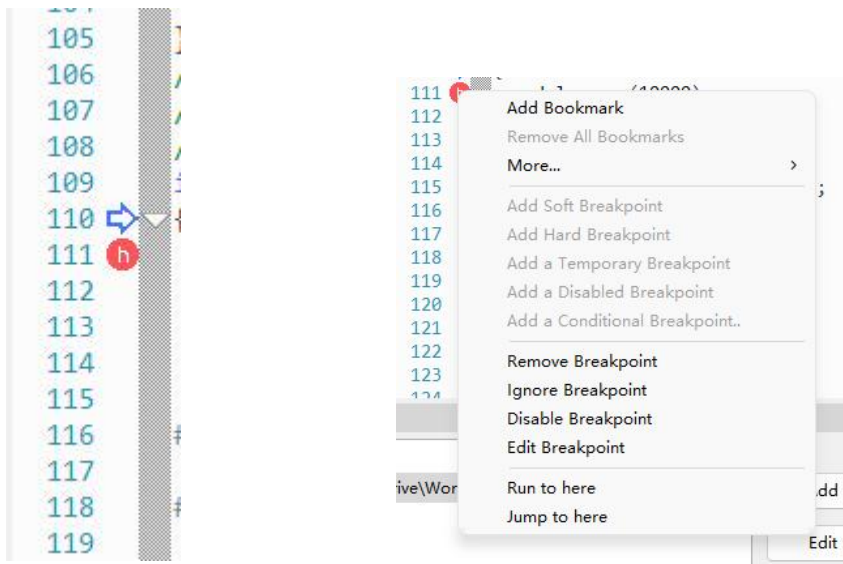


CDK编译与调试2

5. 点击  可以开始运行程序



6. 点击程序窗口的行，可设置断点，在断点上右键可对断点进行配置




7. 点击  可以暂停程序

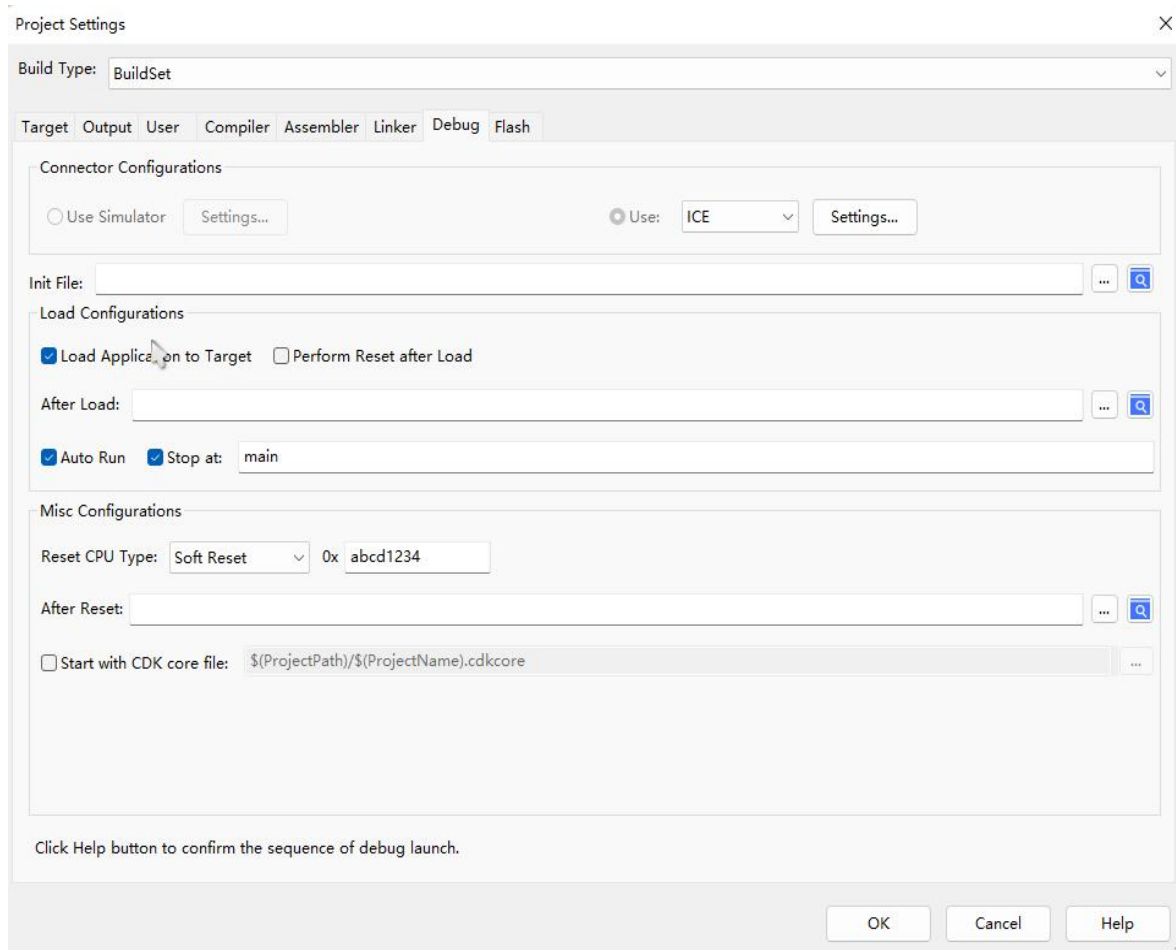
8. 点击  可以将程序下载至芯片

03

CDK 工程设置介绍



点击  按钮，可进入工程配置界面；



CDK工程配置菜单

TargetTab

描述了工程配置的基本信息以及基本属性、地址空间的配置；

OutputTab

描述了工程输出的格式，是可执行或者静态库文件，以及配置是否生成中间相关文件；

UserTab

用于配置在构建工程期间，需要执行的命令；

CompilerTab

用于对编译选项的配置；

AssemblerTab

用于对汇编选项的配置；

LinkerTab

用于对链接器选项的配置；

DebugTab

用于对debugg的配置；

FlashTab

用于对Flash选项的配置；

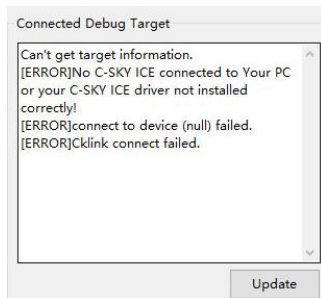
*: 对于以上Tab功能的详细说明，请参考CDK--->Help菜单

04

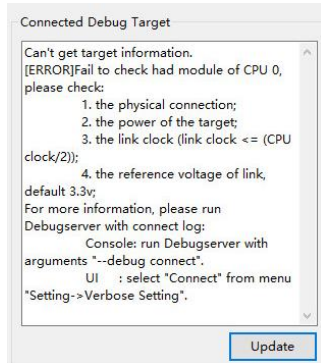
CDK使用常见问题介绍

常见问题1

● CDK无法连接

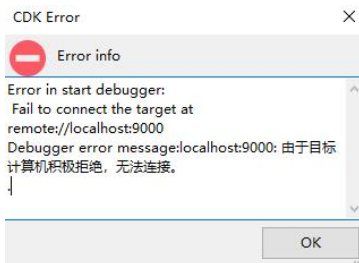


如果CDK提示如左图，说明您的的仿真器没有和PC连接。



如果CDK提示如左图，说明您的仿真器没有和芯片连接。请检查：

- ✓ 仿真器和芯片的物理连接是否稳定。
- ✓ 对于一些有多组SWD口可以选择的芯片，请检查实际连接的SWD口是否和user option中的设置一致。
- ✓ 芯片的SWD口有没有被误改为非SWD功能。如果您曾经连接成功，而后失败，很可能是这种情况

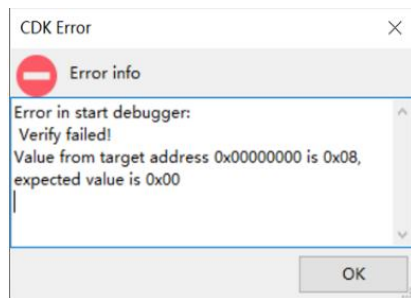


如果CDK提示如左图，说明您的仿真器处于非正常状态。通常可以通过复位仿真芯片或是给仿真器重新上电解决。

常见问题2

● Flash 下载校验失败(verify failed)

当调试器提示 校验失败时

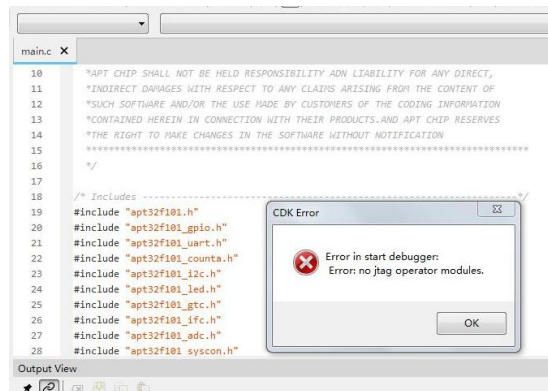


您可以尝试:

- 1、重新下载
 - 2、尝试flash erase
 - 3、检查是否加载了正确的flash算法文件。
- 如果上述尝试都不成功，请更换芯片

常见问题3

● 动态库丢失(Error in start debugger)



此错误说明CDK找不到内部DebugServer需要使用的动态库。
请确认电脑是否安装有安全卫士等软件删除了CDK的调试文件。
重新安装CDK软件并将CDK文件夹加入白名单可解决

常见问题4

● 访问错误异常

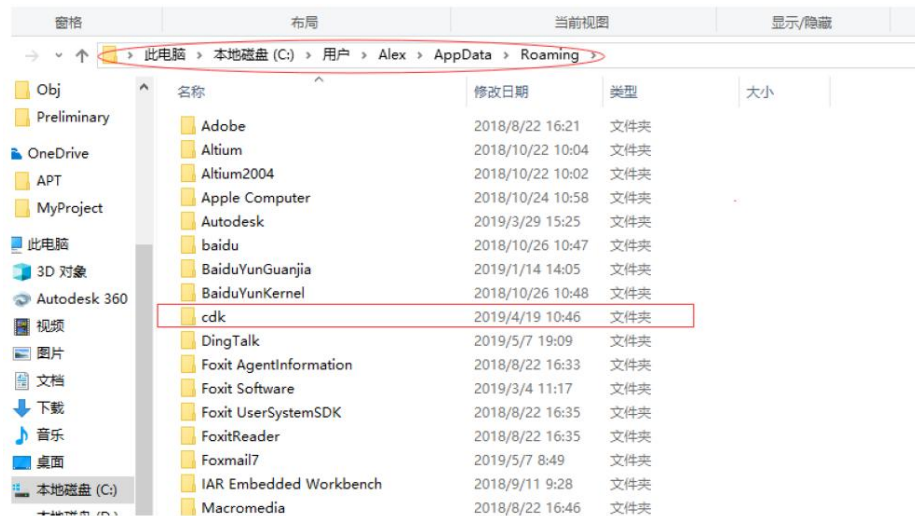
此问题为（访问错误异常）异常中断，原因和解决办法如下

- ✓ 如果客户代码在使用定义比较多结构体成员的结构体时，这个时候SRAM寻址很容易溢出，可以在watch窗口查询R14（堆栈指针）是否超出MCU SRAM地址。当SRAM寻址超出后，MCU会进入AccessErrorHandler异常中断。
- ✓ 建议在定义变量时少用结构体或定义较少的结构体成员。

常见问题5

● CDK软件出错

若出现CDK无法打开，重装后也无法使用的情况，可修改C盘->用户->客户名->AppData->Roaming，修改CDK文件名，或直接删除CDK文件夹。



常见问题6

- CDK突然无法使用

如果您的电脑使用了如360或腾讯安全管家等杀毒软件，除了在安装过程中允许CDK的操作之外，安装之后，必须将整个CDK安装目录加入到杀毒软件的白名单区，否则该类杀毒软件可能将CDK的相关文件删除，进而造成CDK无法使用。