**Ubuntu下使用Code::Blocks+OpenOCD+GDB**

**搭建Atmel MCU開發偵錯環境**

[Nuttx\_Fan\_now](https://blog.csdn.net/oldman147)於 2019-05-25 19:51:10 發佈

閱讀量887 收藏 1

點贊數 1

分類專欄： [ARM](https://blog.csdn.net/oldman147/category_2094959.html) 文章標籤： [codeblocks](https://so.csdn.net/so/search/s.do?q=codeblocks&t=all&o=vip&s=&l=&f=&viparticle=) [OpenOCD](https://so.csdn.net/so/search/s.do?q=OpenOCD&t=all&o=vip&s=&l=&f=&viparticle=) [GDB](https://so.csdn.net/so/search/s.do?q=GDB&t=all&o=vip&s=&l=&f=&viparticle=) [Atmel](https://so.csdn.net/so/search/s.do?q=Atmel&t=all&o=vip&s=&l=&f=&viparticle=) [SAMV71](https://so.csdn.net/so/search/s.do?q=SAMV71&t=all&o=vip&s=&l=&f=&viparticle=)

版權

[ARM專欄收錄該內容](https://blog.csdn.net/oldman147/category_2094959.html)

**在**[Ubuntu](https://so.csdn.net/so/search?q=Ubuntu&spm=1001.2101.3001.7020)**下搭建Atmel MCU開發環境**

* [1、準備工作](https://blog.csdn.net/oldman147/article/details/90544936#1_1)
  + [安裝Code::Blocks](https://blog.csdn.net/oldman147/article/details/90544936#CodeBlocks_4)
  + [下載arm-none-eabi-gcc編譯器](https://blog.csdn.net/oldman147/article/details/90544936#armnoneeabigcc_14)
  + [安裝OpenOCD工具](https://blog.csdn.net/oldman147/article/details/90544936#OpenOCD_22)
* [2、組態Code::Blocks](https://blog.csdn.net/oldman147/article/details/90544936#2CodeBlocks_28)
  + [建立一個新的Debugger組態](https://blog.csdn.net/oldman147/article/details/90544936#Debugger_30)
  + [組態Compiler](https://blog.csdn.net/oldman147/article/details/90544936#Compiler_36)
* [3、start.atmel.com下載官方例程](https://blog.csdn.net/oldman147/article/details/90544936#3startatmelcom_41)
* [4、在Code::Blocks新建工程並匯入atmel官方例程](https://blog.csdn.net/oldman147/article/details/90544936#4CodeBlocksatmel_44)

**1、準備工作**

作為一個重度Linux系統使用者，平時開發相關的工作都是在Ubuntu下進行，最近有個項目需要用到Atmel 32位的Cortex-M7 MCU，所以花時間研究下如何在Ubuntu下搭建起ARM MCU的開發偵錯環境！ 如果你也是想使用 **Code::Blocks** 進行ARM MCU相關的開發工作，可以參照這篇部落格所列出的步驟指引搭建起開發偵錯環境。

**安裝Code::Blocks**

**Code::Blocks** 是作者非常喜歡和推薦的一款多平台的軟體開發IDE，尤其對C/C++的支援做的特別好，整個軟體對電腦資源的消耗很少，輕便快捷，程式設計師最看中的程式碼補齊和搜尋功能絲毫不弱於Source Insight和vs code。

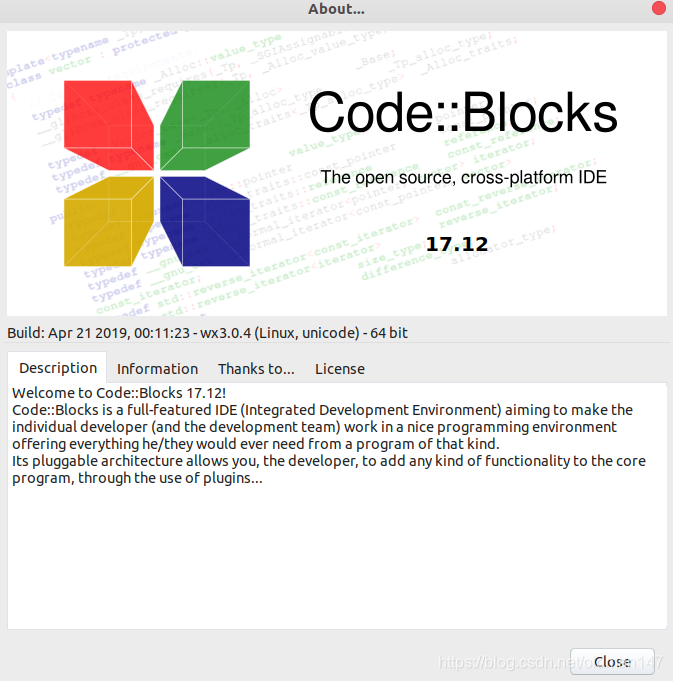
作者使用的**Code::Blocks** 版本是17.12,自己去官網下載原始碼手動編譯安裝的，動手能力強的可以嘗試手動編譯下，或者省事的點就在命令終端輸入 sudo apt-get install codeblocks. (Ubuntu默認支援的版本是16.x的舊版本)

如果需要更新版本的，可以嘗試：

$sudo add-apt-repository ppa:damien-moore/codeblocks-stable

$sudo apt update

$sudo apt install codeblocks codeblocks-contrib



**下載arm-none-eabi-gcc編譯器**

直接去ARM官網就可以下載，記得下載for Linux的版本。

<https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads>

演示時使用的是2018-q4版本：

[Version 8-2018-q4-major Linux 64-bit](https://armkeil.blob.core.windows.net/developer/Files/downloads/gnu-rm/8-2018q4/gcc-arm-none-eabi-8-2018-q4-major-linux.tar.bz2)

將下載後的編譯器工具包解壓到使用者目錄下，比如作者將工具解壓到 **/home/kevin/opt** 目錄

**安裝OpenOCD工具**

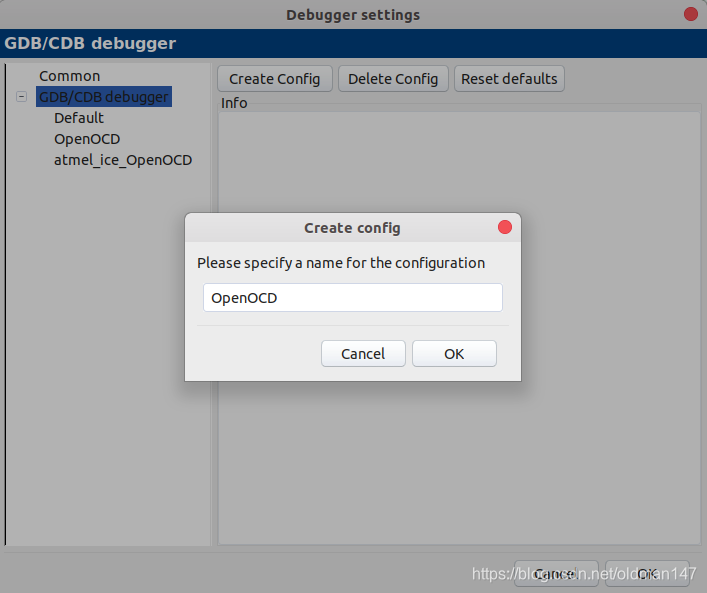
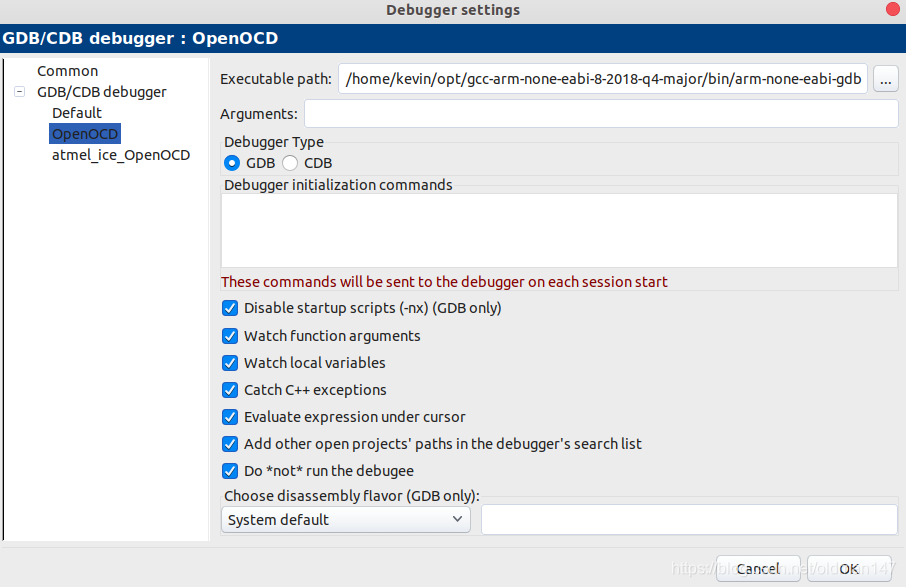
**OpenOCD（Open On-Chip Debugger）** 開源片上偵錯程式，是一款開放原始碼軟體，最初是由Dominic Rath同學還在大學期間發起的（2005年）項目。**OpenOCD** 旨在提供針對嵌入式裝置的偵錯、系統程式設計和邊界掃描功能。

**OpenOCD** 的功能是在模擬器的輔助下完成的，模擬器是能夠提供偵錯目標的電訊號的小型硬體單元。模擬器是必須的，因為偵錯主機（運行**OpenOCD** 的主機）通常不具備這種電訊號的直接解析功能。在Linux主機環境下，我們可以用**GDB + OpenOCD** 就可以進行程式碼的下載和偵錯。

說道這裡得致敬下10幾年前國內大牛做的一個非常好用的偵錯程式叫 **H-JTAG**， 可惜的是後來停止更新。在當時那個ARM偵錯工具非常稀缺的年代，H-JTAG的出現還是幫助到了很多早期ARM開發人員。

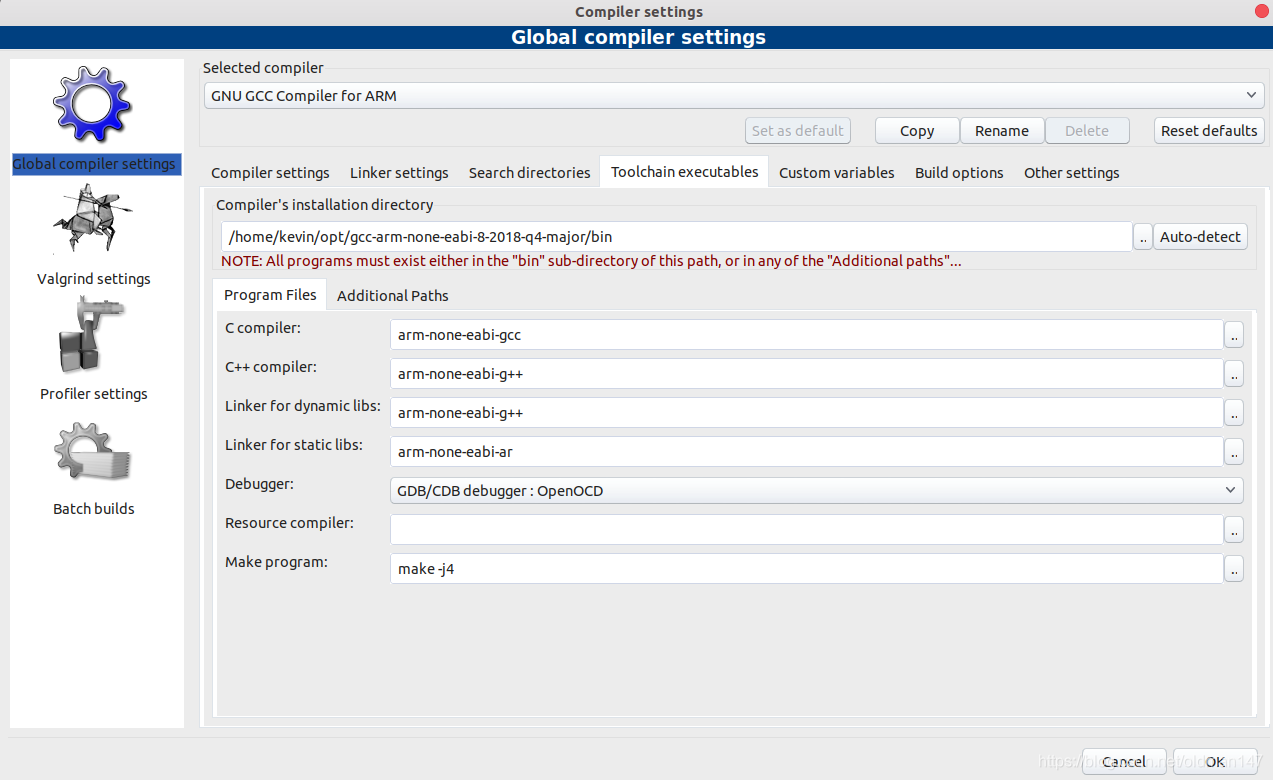
**2、組態Code::Blocks**

**建立一個新的Debugger組態**

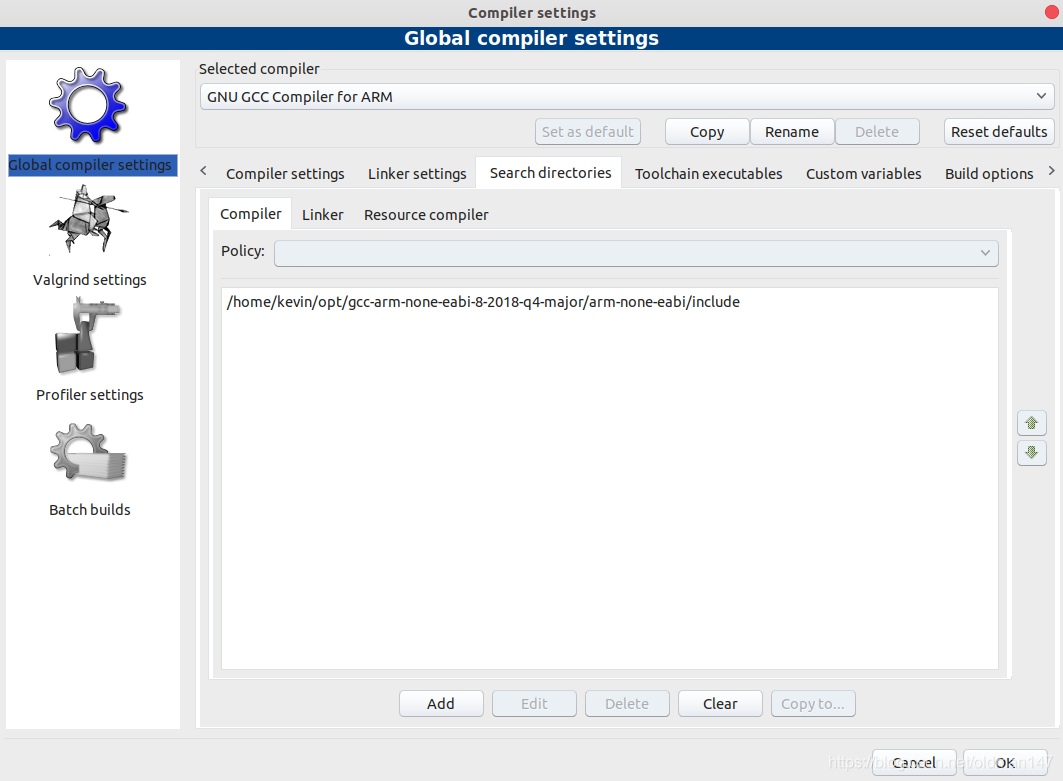
打開**Code::Blocks**，選擇**Settings** -> **Debugger.**, 然後選擇**Create Config**, 新建立一個名叫組態OpenOCD的組態，點選OK保存新的組態。  
  
接著對新建立的Debugger進行組態，需要指定GDB工具路徑：  


**組態**[Compiler](https://so.csdn.net/so/search?q=Compiler&spm=1001.2101.3001.7020)

打開**Code::Blocks**，選擇**Settings** -> **Compiler.**, 在Selected Compiler下拉框裡面選擇**GNU GCC Compiler for ARM**, 我們需要組態下arm-none-eabi-gcc編譯工具路徑等內容，Debugger選擇新建立的OpenOCD，最後點選OK保存組態。



另外還可以組態系統編譯器標頭檔尋找路徑：



**3、start.atmel.com下載官方例程**

[Atmel官網例程線上組態網站start.atmel.com](http://xn--atmelstart-zy5p244e5fnb27hlschtx2icea46cs560a.atmel.com/)，可以線上組態所需要的MCU外設，外設驅動參數等資訊，用起來還是非常方便，關鍵組態的工程最後下載支援多種整合開發環境，比如它官方的Atmel Studio 7，通用的IAR/Keil，還有Linux使用者可以用的GCC Makefile。

由於Code::Blocks直接支援匯入使用者**Makefile**檔案進行編譯（敲下小黑板劃重點-這點很重要），這樣很多開放原始碼專案可以直接通過此類的方法進行組態匯入**Code::Blocks**中進行編譯和偵錯模擬。原始碼具體的下載過程就請參考後面提供的視訊連結。

**4、在Code::Blocks新建工程並匯入atmel官方例程**

整個操作流程請參考以下優酷網視訊連結：

[視訊教學](https://v.youku.com/v_show/id_XNDE5NjM2MTM5Mg==.html?spm=a2h3j.8428770.3416059.1)