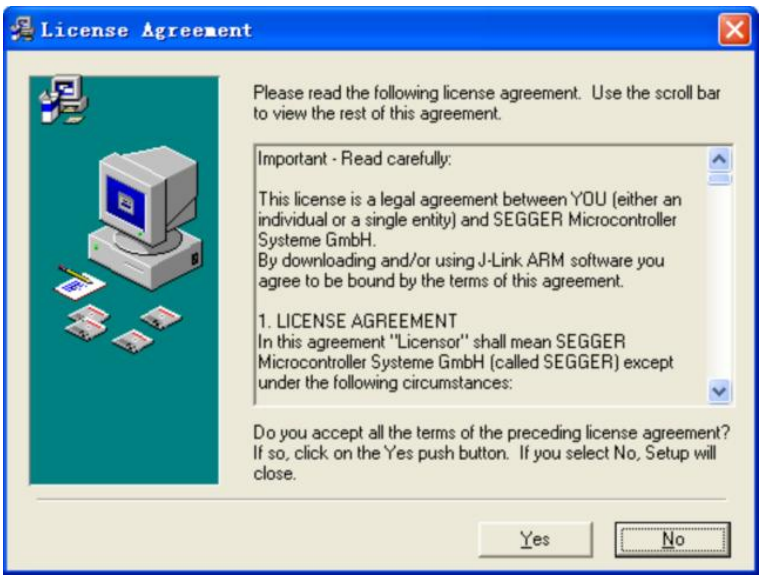
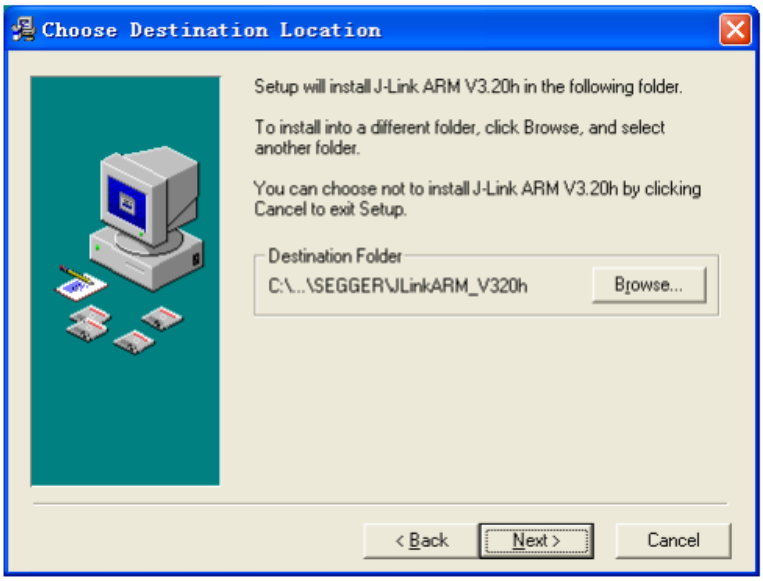
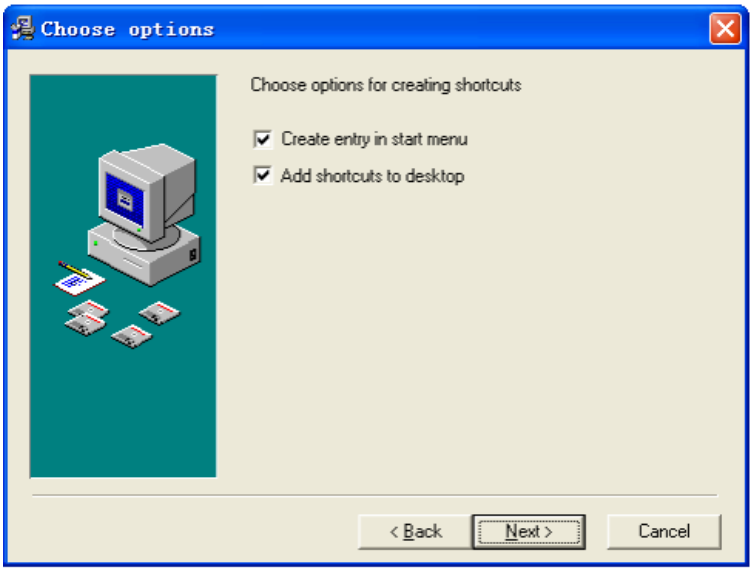
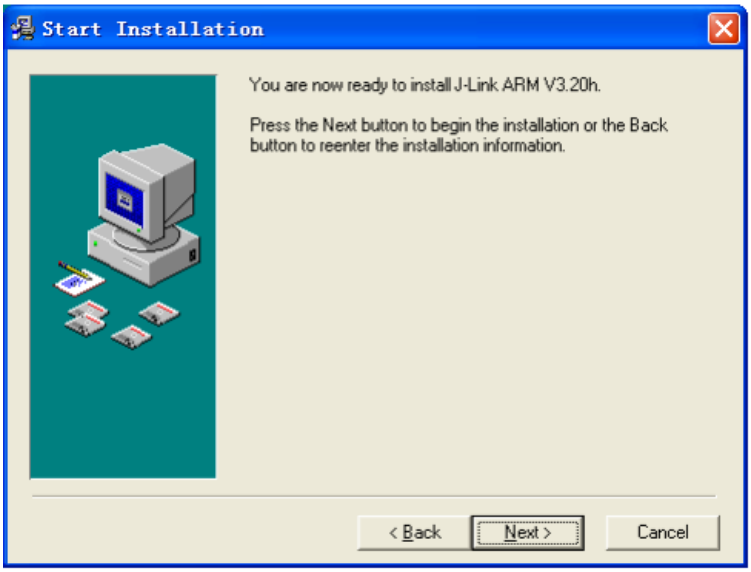
Install driver

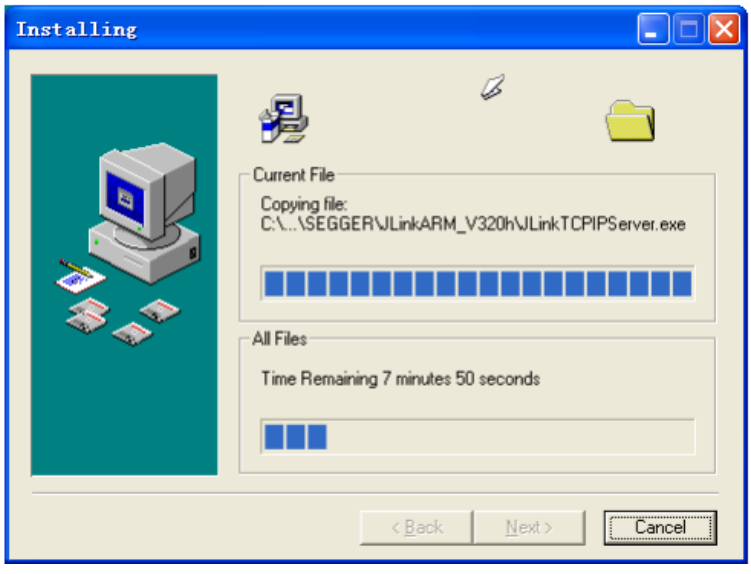
First of all, you need to download the newest driver software from there:
http://www.segger.com/download jlink.html
It named J-Link ARM software and documentation pack, includes USB driver, J-Mem, J-Link.exe and DLL for ARM, J-Flash and J-Link RDI. Then just directly installed after unzip the downloaded zip file.

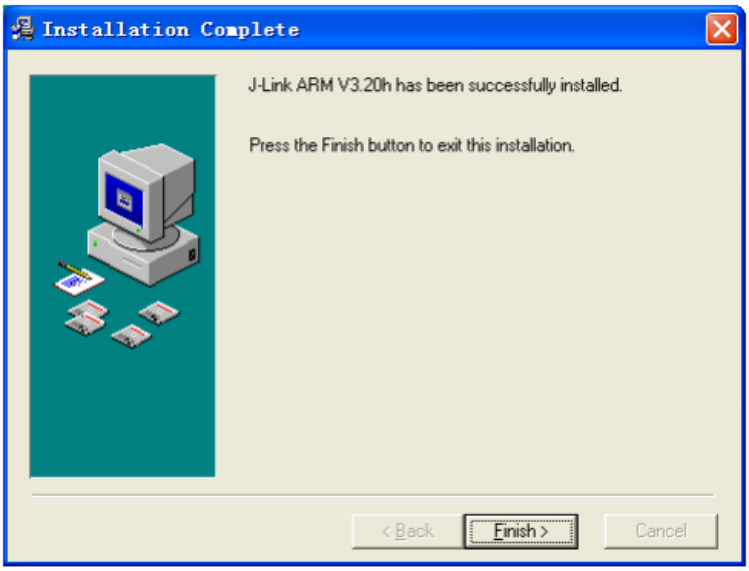




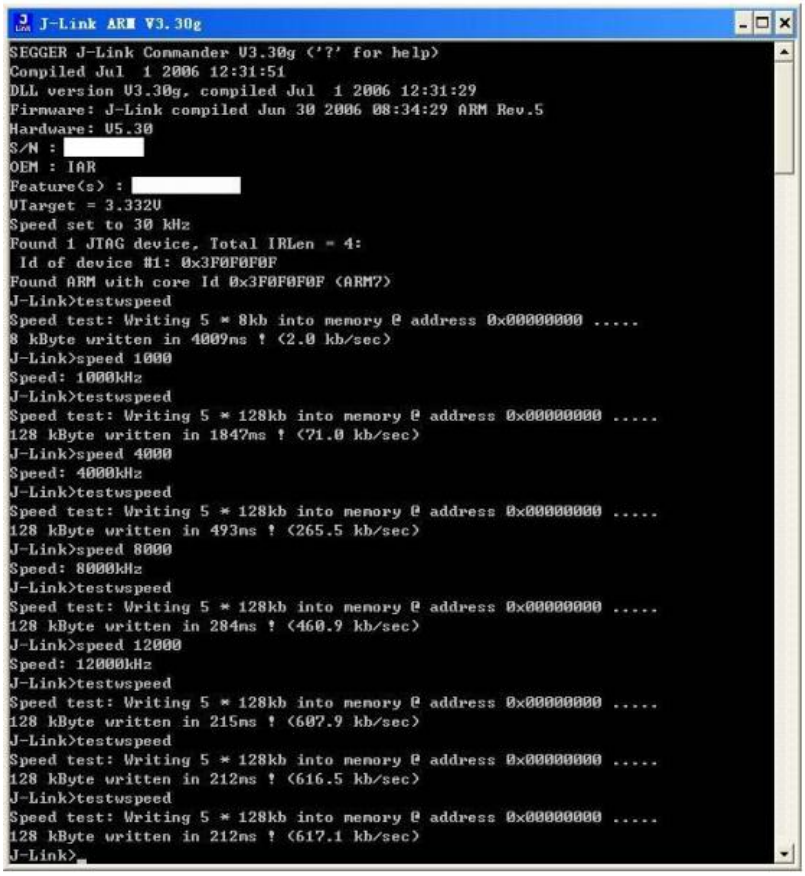

Choose options
Choose options for creating shortcuts
Create entry in start menu
Add shortcuts to desktop
< Back
Next >
Cancel


Start Installation
You are now ready to install J-Link ARM V3.20h.
Press the Next button to begin the installation or the Back button to reenter the installation information.
< Back
Next >
Cancel

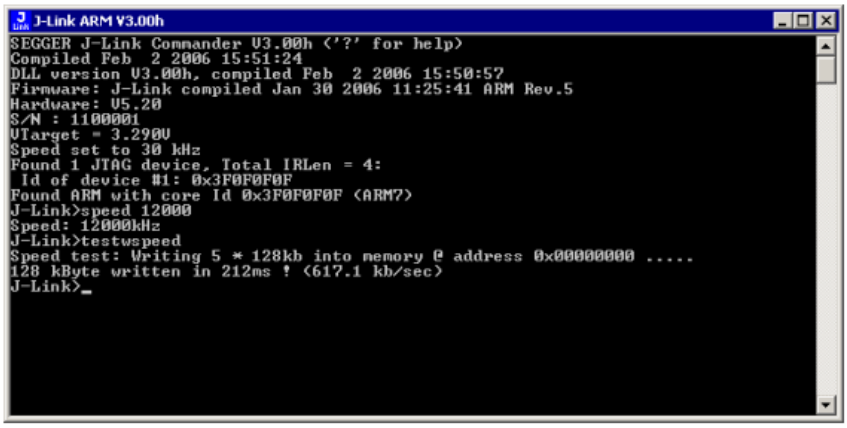

Installing
Current File
Copying file:
C:\...\SEGGER\JLinkARM_V320h\JLinkTCPIPServer.exe
All Files
Time Remaining 7 minutes 50 seconds
< Back
Next >
Cancel



After that, please insert the hardware of J-Link, it will install the driver automatic. When it finish, there will be two desktop shortcut icons, one of them called J-Link ARM can be used to set up and have a test. Next let us read the test data of J-Link.(testing on the 7X256 EK)

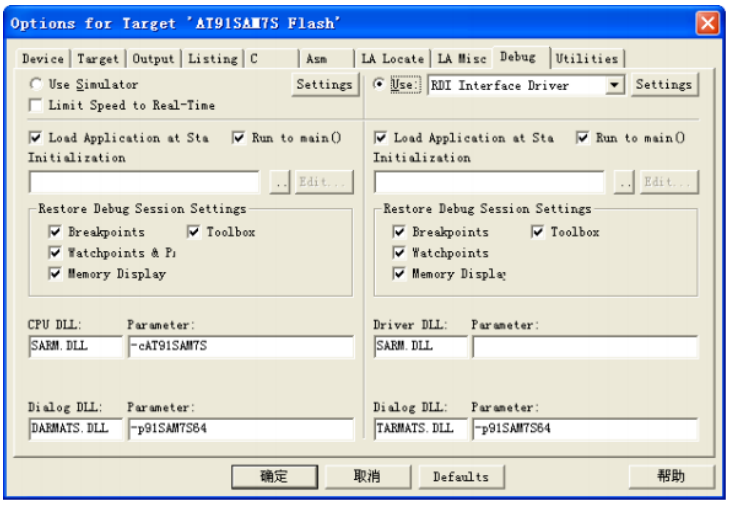


And the raw test data:

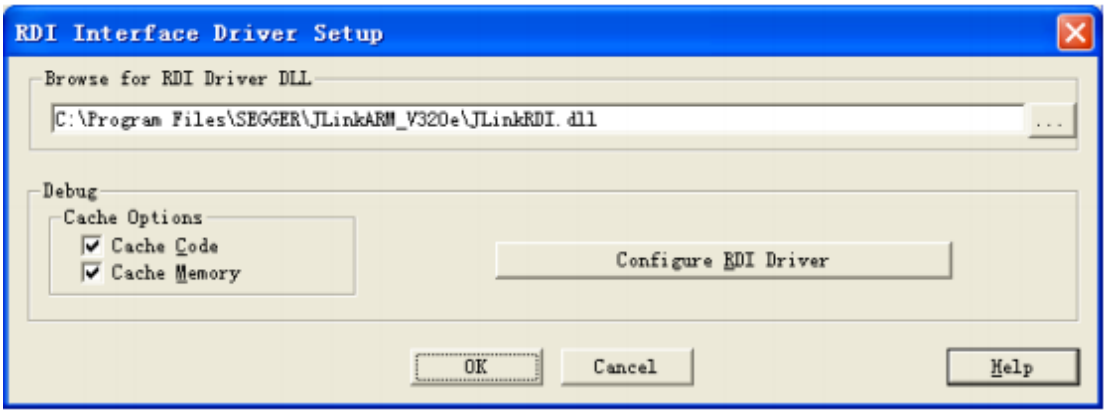


How we can set it up with mainstream development environment

1: Use with keil

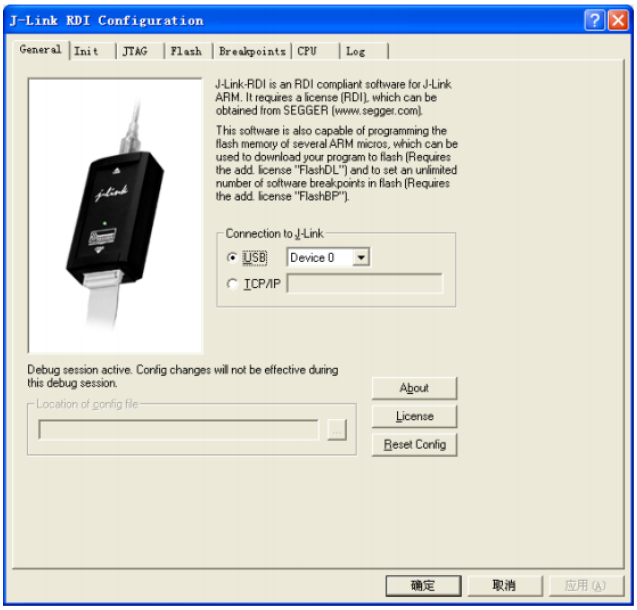


Then choose the “RDI Interface Driver”, and click the “Settings”

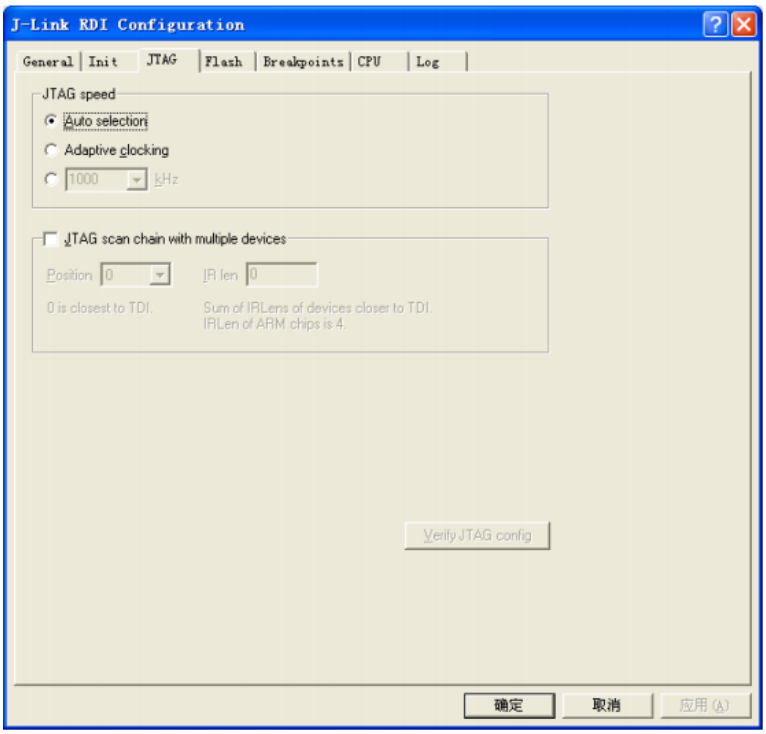


Click the “…” to set up the installation path. Next click the “ 

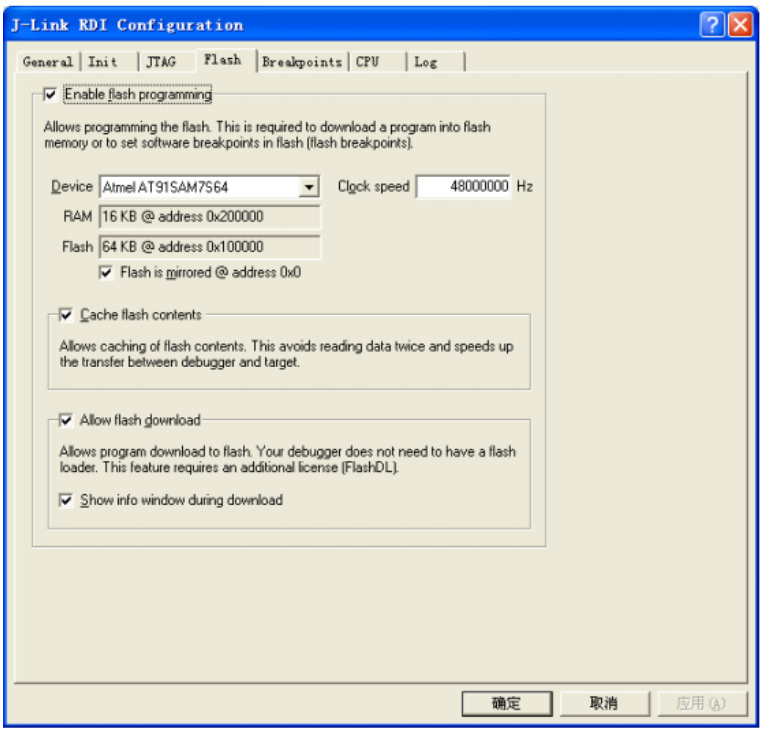
”, it will pop up a window as shown:



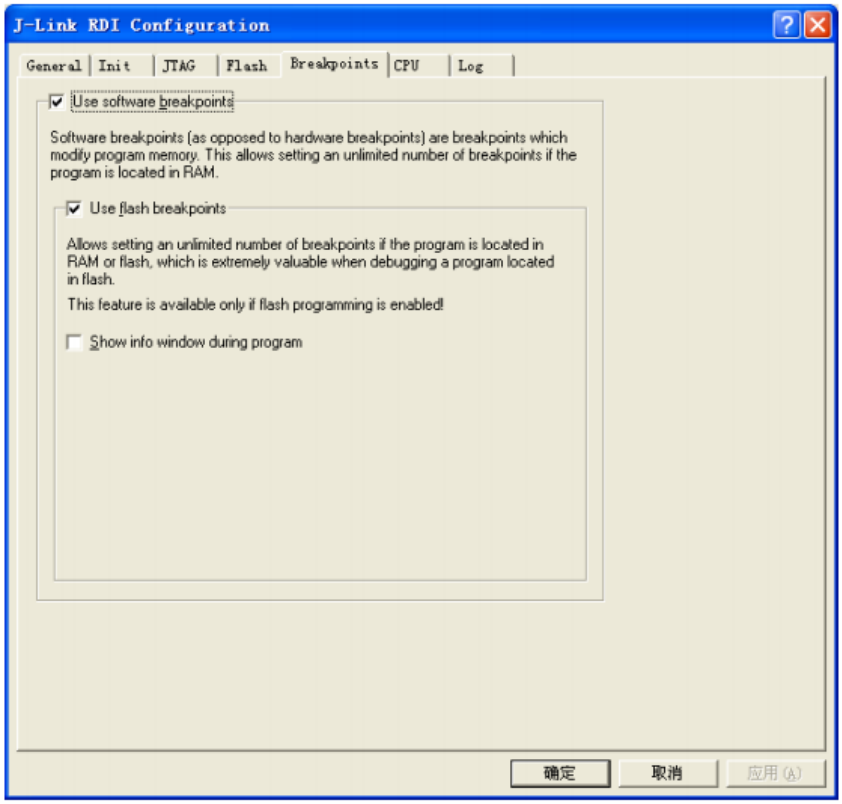
If Native debugging, use the USB port. If LAN debugging, choose the TCP/IP, and Designation a IP of the PC that mounted J-Link.



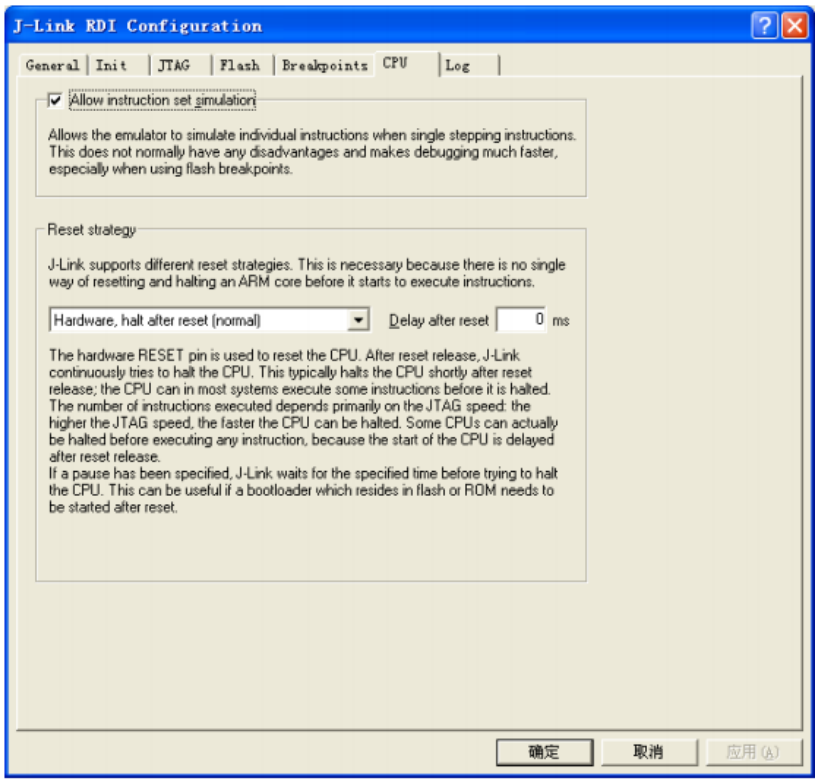
Set up the JTAG speed. Suggest “Auto selecting” if there is “–s” kernel. You could reduce the speed if there something unstable, otherwise, always use the highest speed-12M.



Using the FLASH programming function if your goal chip is the ARM with on-chip FLASH. Like this J-Link will be programmed FLASH before commissioning.

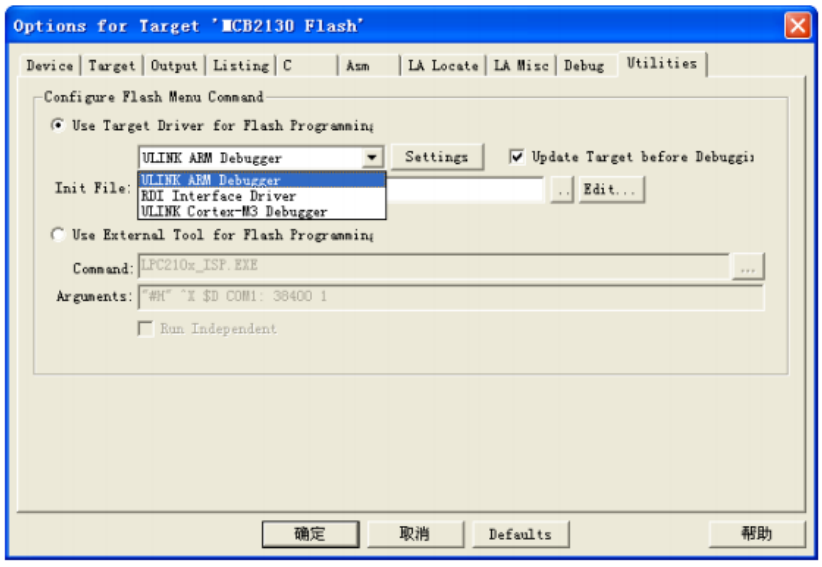


Use this “software breakpoints” function to easy commissioning.



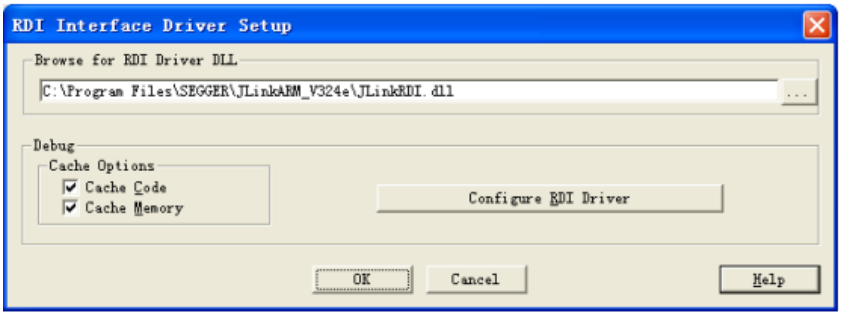
Setting reset mode and the delay time after reset.

If you need to use the “DOWNLOAD” function in the “KEIL”. You should have a set as same as “DEBUG” in the “Utilities” menu:



Choose” RDI Interface Driver”, and click 

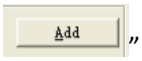
Choose “J-Link Flash Programmer”

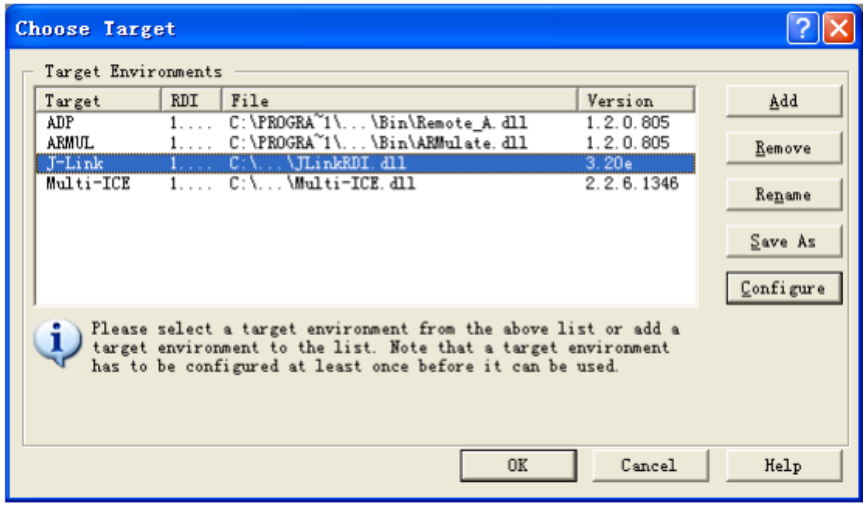


The next steps are same as “Debug” set up.

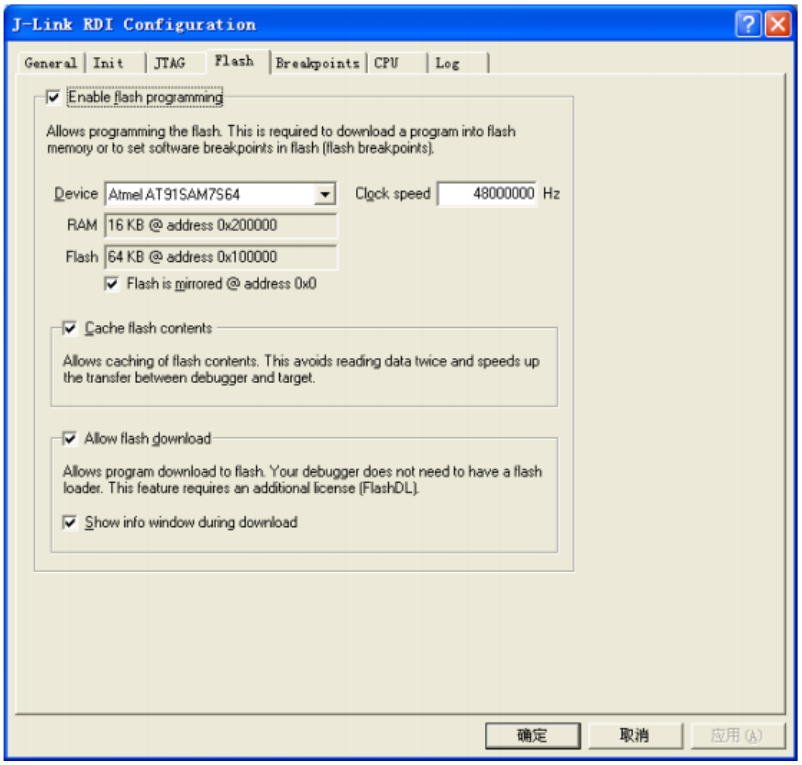
You can use the  to download directly.(only support the ARM7/9 chip have FLASH)

2.Use with ADS

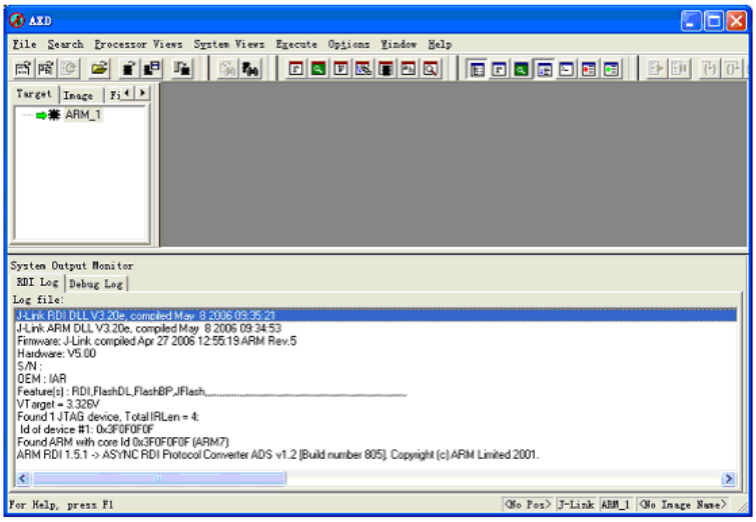
Click the “  ” then choose “JLINKRDI.DLL”:



Click  , pop up a window as below:

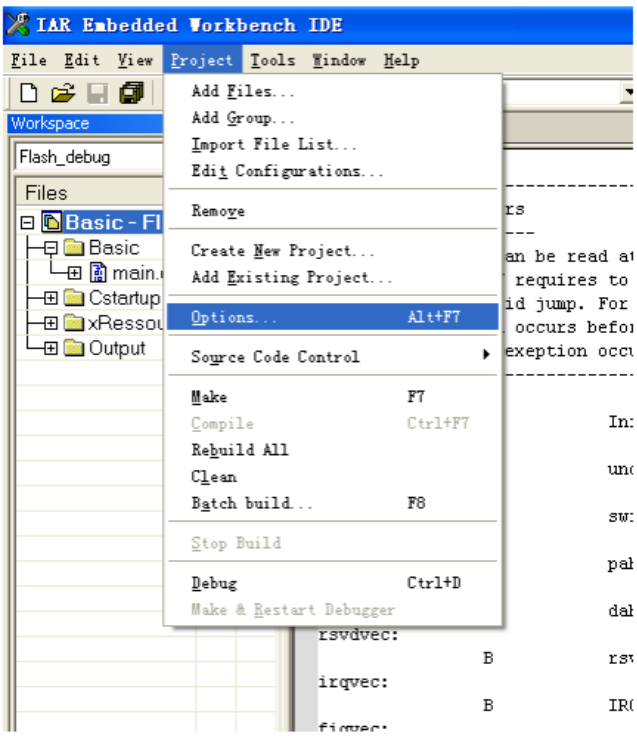


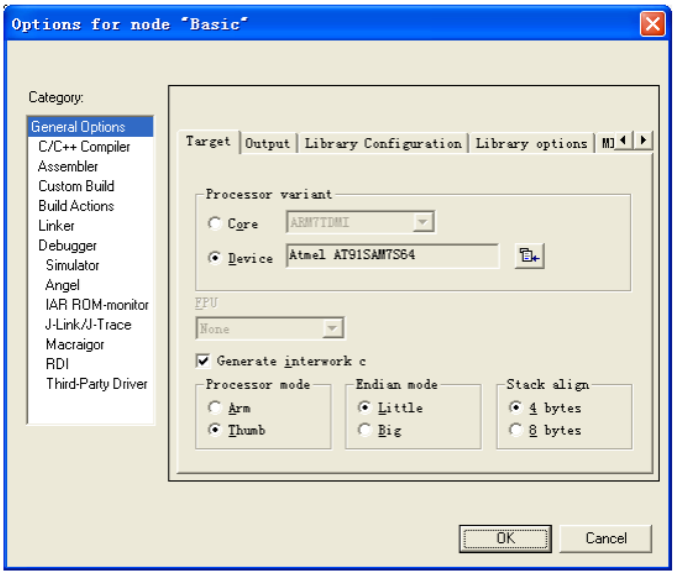
Information after entering the AXD:

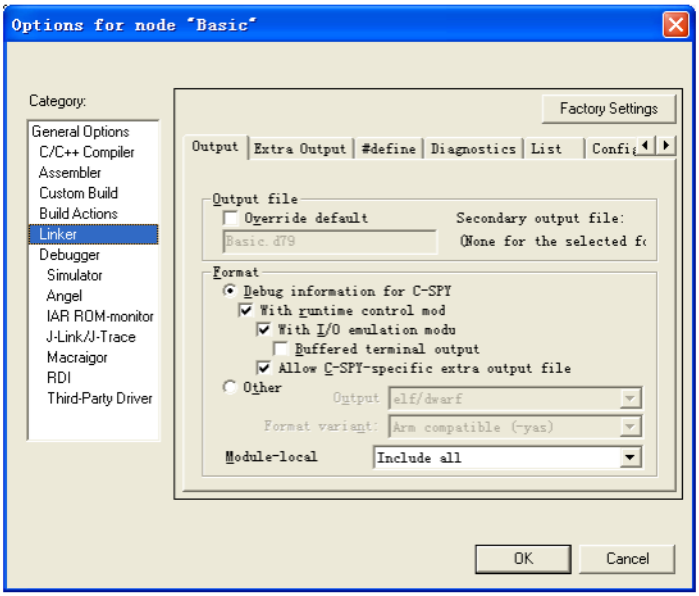


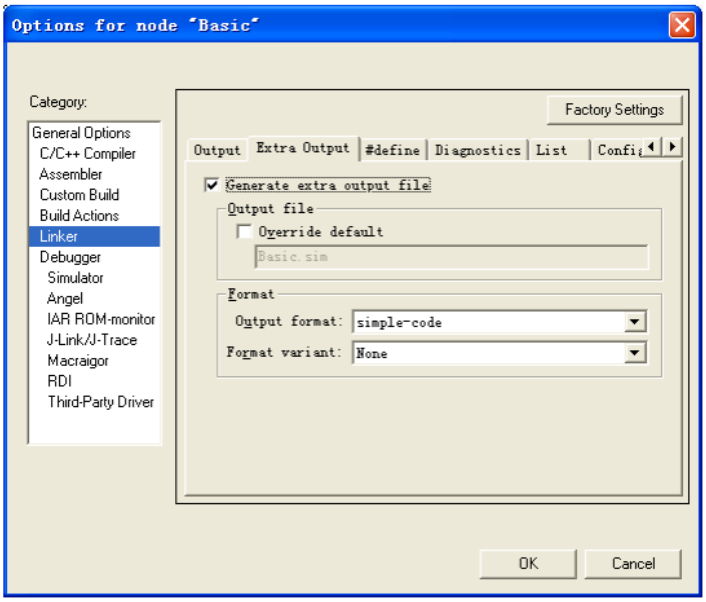
3.Use with IAR:

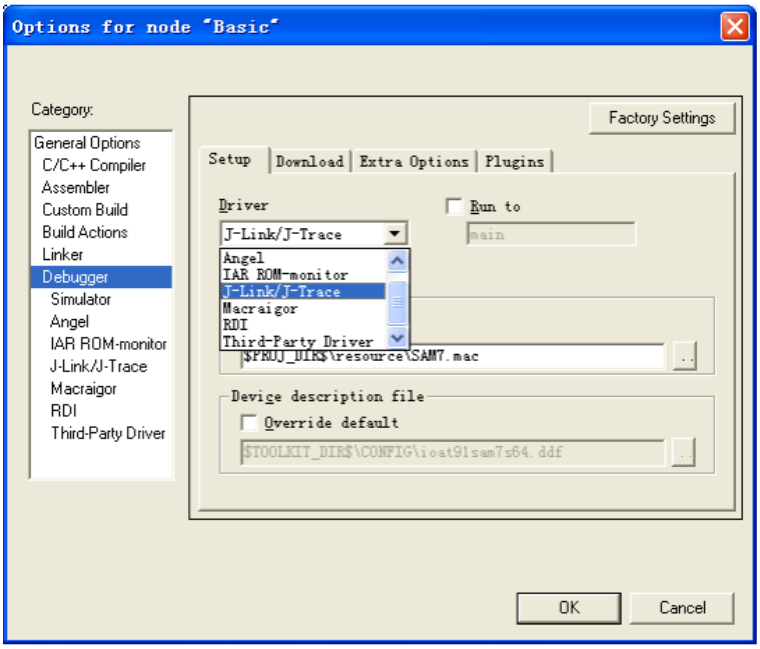
Open a project and enter the settings page as shown below:


IAR Embedded Workbench IDE
File Edit View Project Tools Window Help
Workspace
Flash_debug
Files
Basic - Fl
Basic
main.
Cstartup
xRessou
Output
Add Files...
Add Group...
Import File List...
Edit Configurations...
Remove
Create New Project...
Add Existing Project...
Options... Alt+F7
Source Code Control
Make F7
Compile Ctrl+F7
Rebuild All
Clean
Batch build... F8
Stop Build
Debug Ctrl+D
Make & Restart Debugger
rsvdvec:
B
irqvec:
B

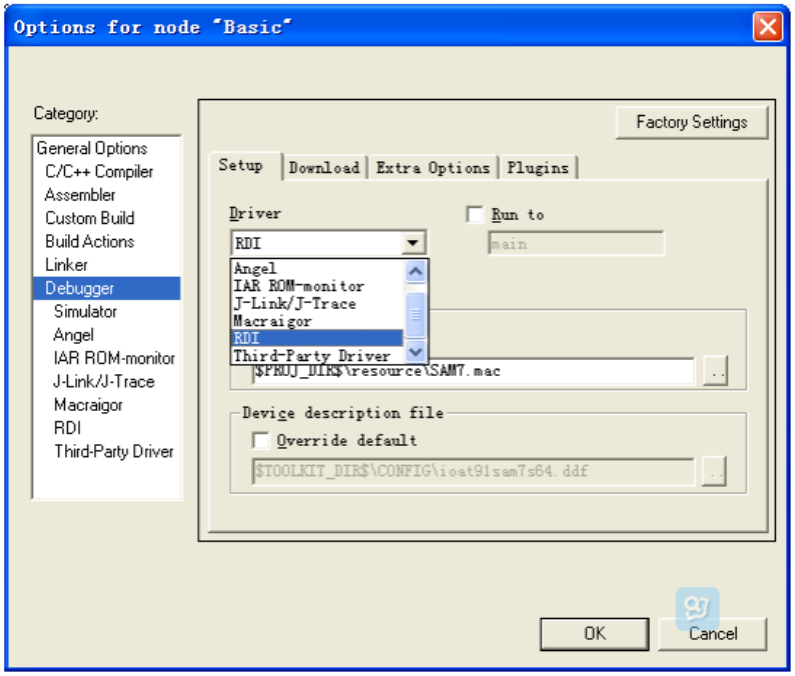

Options for node "Basic"
Category:
General Options
C/C++ Compiler
Assembler
Custom Build
Build Actions
Linker
Debugger
Simulator
Angel
IAR ROM-monitor
J-Link/J-Trace
Macraigor
RDI
Third-Party Driver
Target
Output
Library Configuration
Library options
Processor variant
Core
ARM7TDMI
Device
Atmel AT91SAM7S64
FPU
None
Generate interwork c
Processor mode
Arm
Thumb
Endian mode
Little
Big
Stack align
4 bytes
8 bytes
OK
Cancel


Options for node "Basic"
Category:
General Options
C/C++ Compiler
Assembler
Custom Build
Build Actions
Linker
Debugger
Simulator
Angel
IAR ROM-monitor
J-Link/J-Trace
Macraigor
RDI
Third-Party Driver
Factory Settings
Output
Extra Output
#define
Diagnostics
List
Output file
Override default
Basic.d79
Secondary output file:
(None for the selected fo
Format
Debug information for C-SPY
With runtime control mod
With I/O emulation modu
Buffered terminal output
Allow C-SPY-specific extra output file
Other
Output
elf/dwarf
Format variant:
Arm compatible (-yas)
Module-local
Include all
OK
Cancel

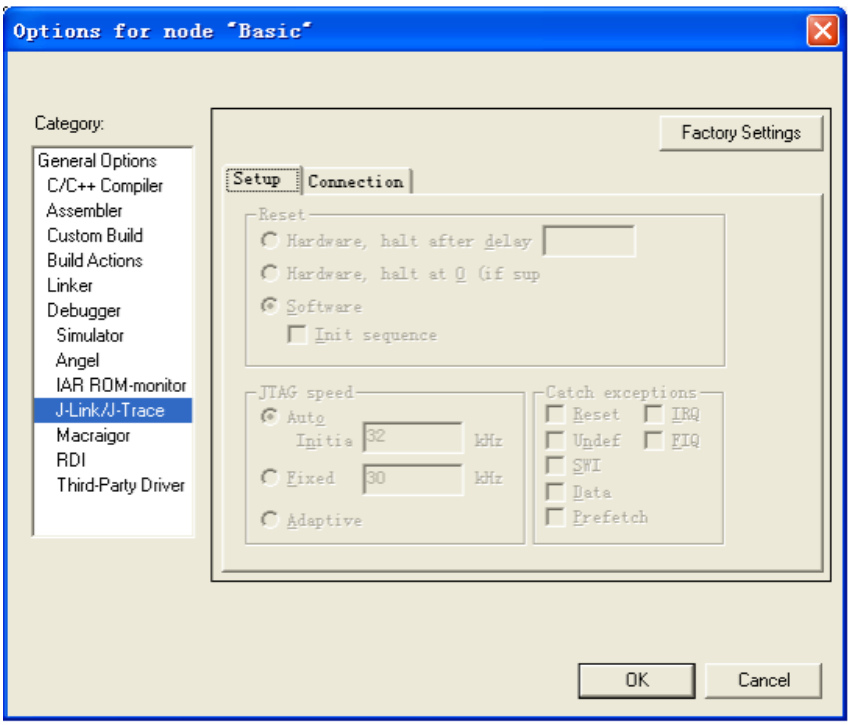

Options for node "Basic"
Category:
General Options
C/C++ Compiler
Assembler
Custom Build
Build Actions
Linker
Debugger
Simulator
Angel
IAR ROM-monitor
J-Link/J-Trace
Macraigor
RDI
Third-Party Driver
Factory Settings
Output
Extra Output
#define
Diagnostics
List
Generate extra output file
Output file
Override default
Basic.sim
Format
Output format:
simple-code
Format variant:
None
OK
Cancel



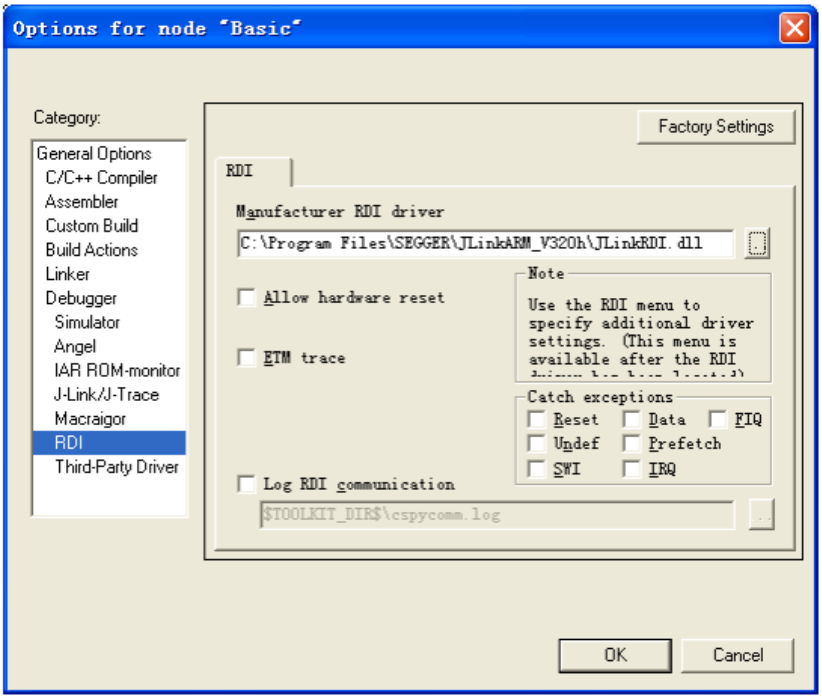
If your J-LINK is full-featured version. You can choose the RDI to promote the performance.



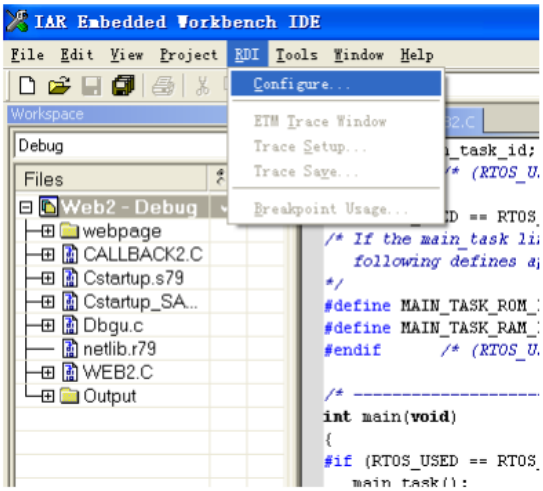
And J-LINK/J Trace do not need to have setup:



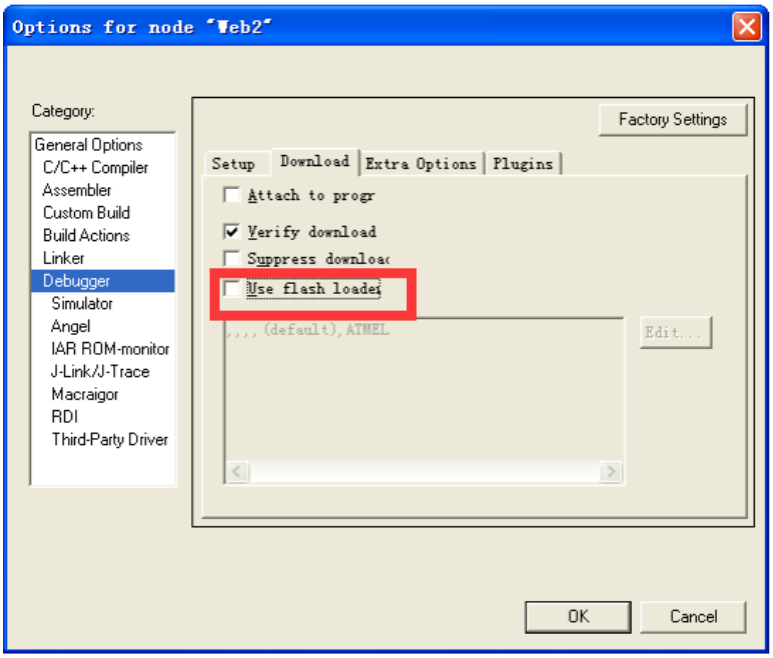
Choose “RDI”, you need to designation the path of “JLINKRDI.DLL”



After finishing, it will add a “RDI” menu:



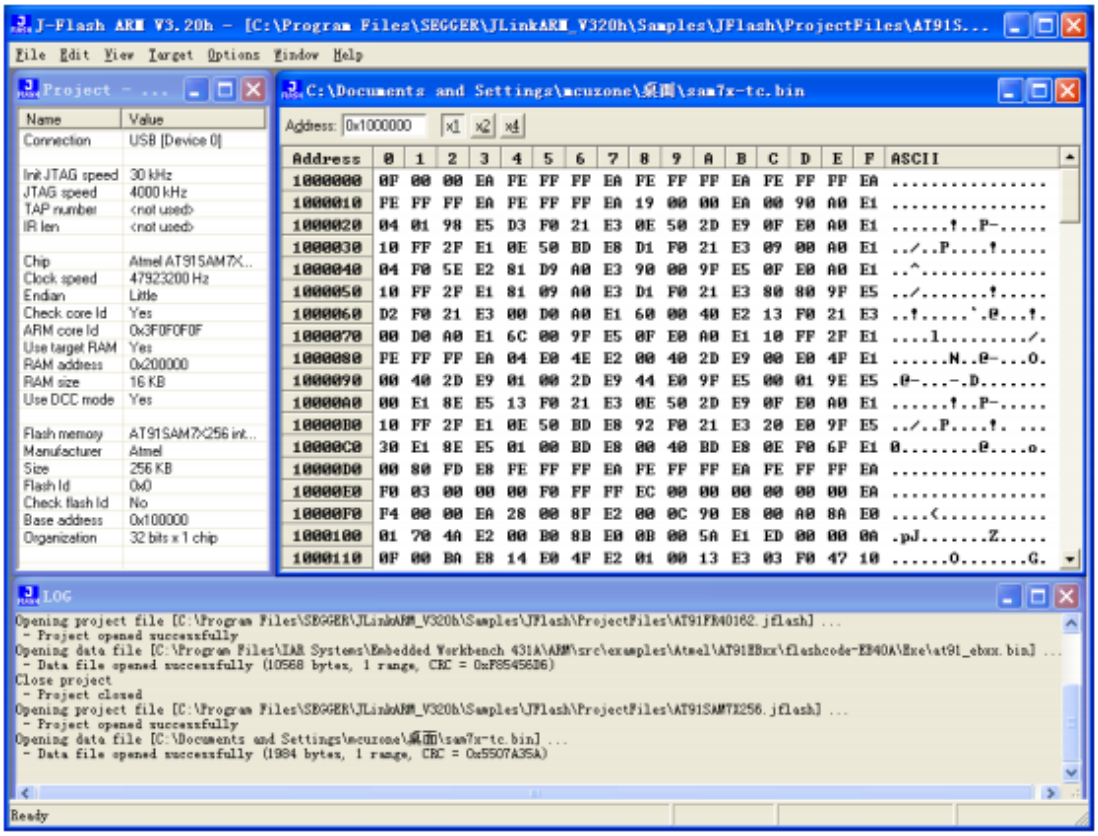
And you can to set up the “JTAG CLOCK, FLASH, breakpoints, CPU and so on. Please not use the “FLASHLOADER” to have flash downloading.



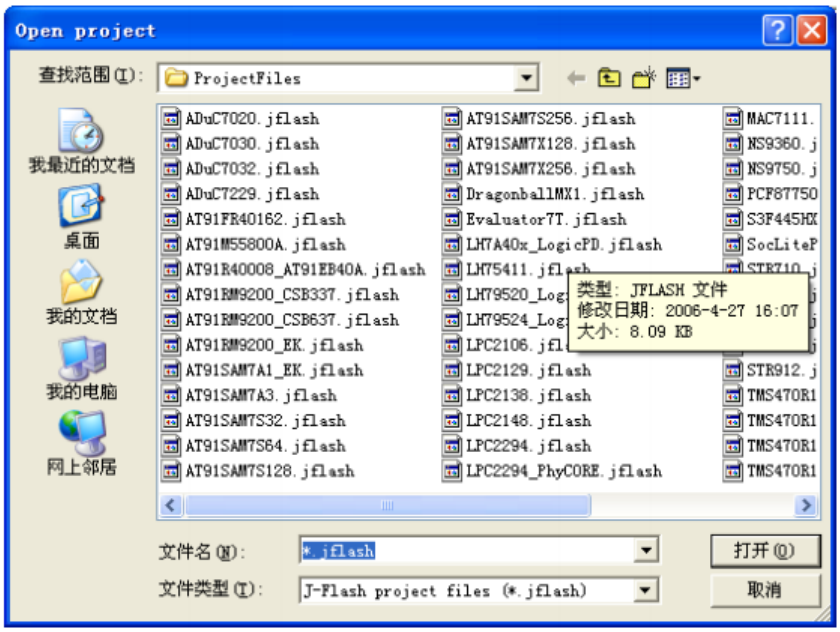
Remove the front tick.

J-FLASH ARM use guide:

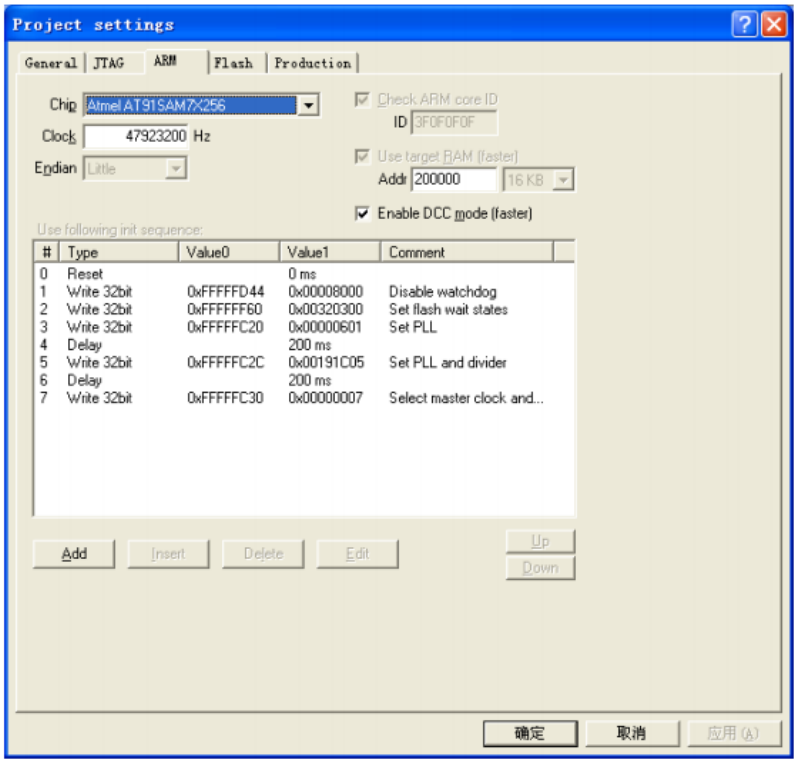
It use for Individually programming “FLASH”



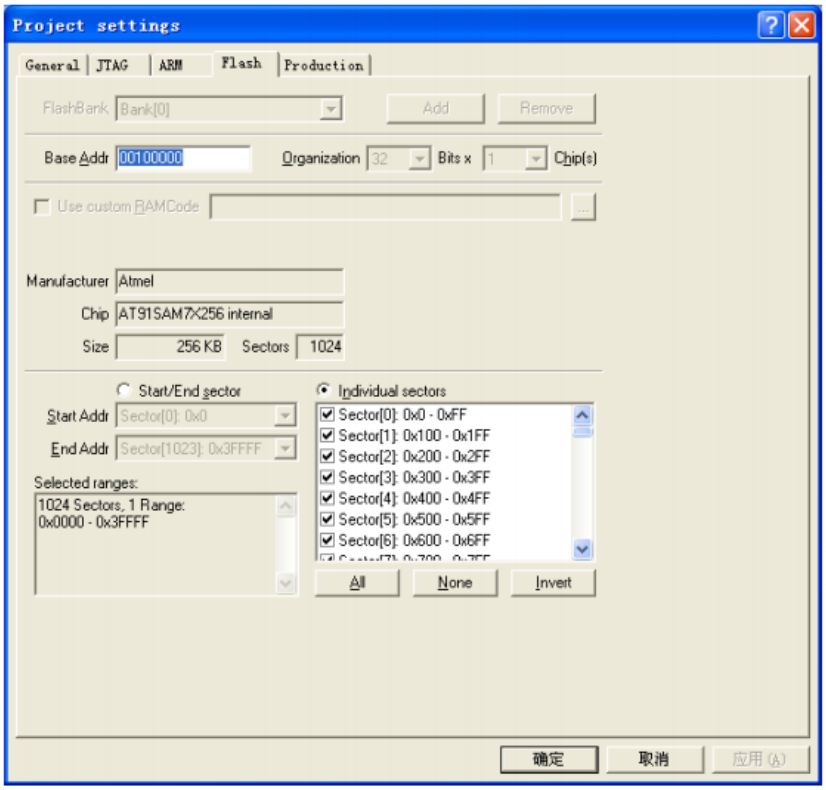
1:Click the “open project” in the “file” menu to choose the goal chip.



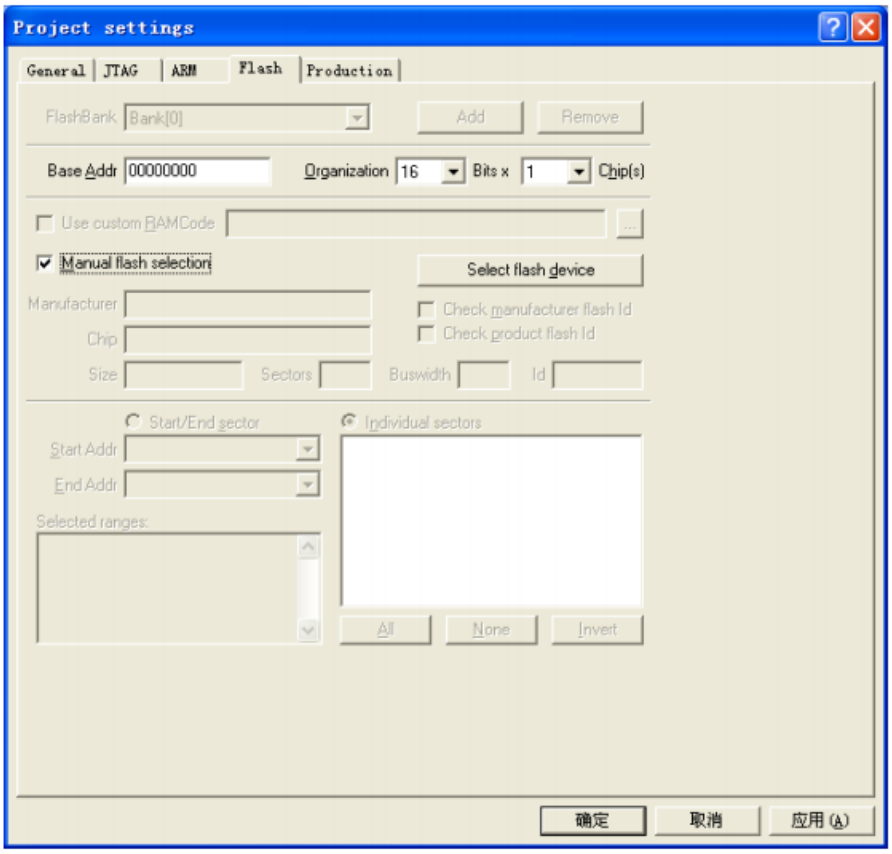
Then click the “Open…” in the “file” menu to open the documents which are need to program. Next “Project settings” in the “Options” menu.

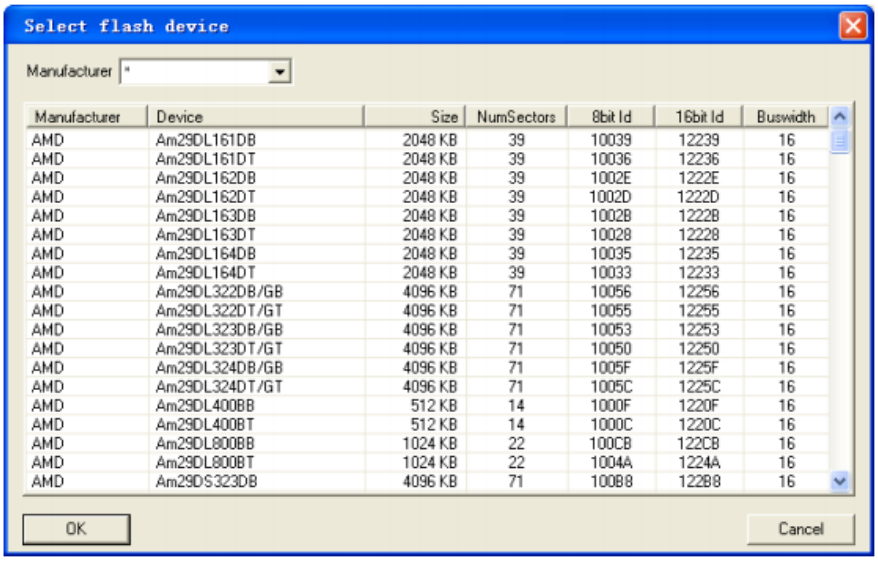


Choose the goal chip in the “ARM” tab. And you need to choose “Generic ARM7/ARM9” if the chip don’t have built-in flash.



In the “FLASH” tab, you need to set carefully if your project is newly created by yourself.
If you choose the “Generic ARM7/ARM9” in the previous step. You could set the flash model in this “FLASH” tab.



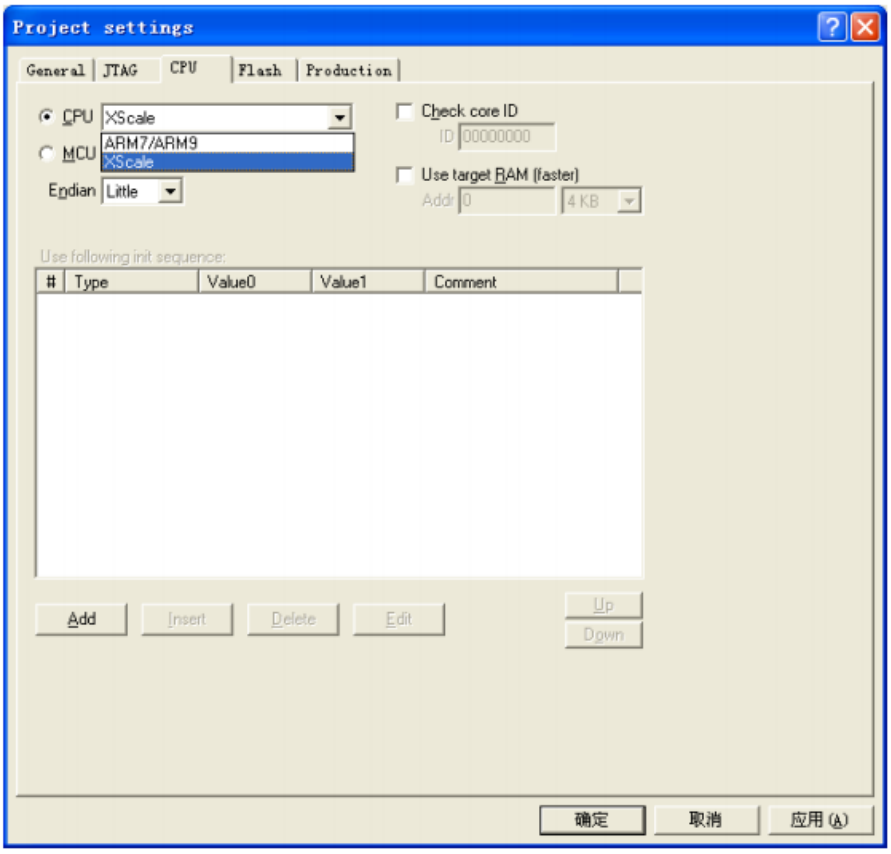


After setting, you can operate on the Target menu, usually follow the below steps:

"Connect"→"Erase Chip"→"Program".

The main operations are done in this menu.

Else J-FLASH ARM start to support XSCALE from 3.30 versions.


Project settings
General
JTAG
CPU
Flash
Production
CPU
XScale
MCU
ARM7/ARM9
XScale
Endian
Little
Check core ID
ID
00000000
Use target RAM (faster)
Addr
0
4 KB
Use following init sequence:
#
Type
Value0
Value1
Comment
Add
Insert
Delete
Edit
Up
Down
确定
取消
应用 (A)