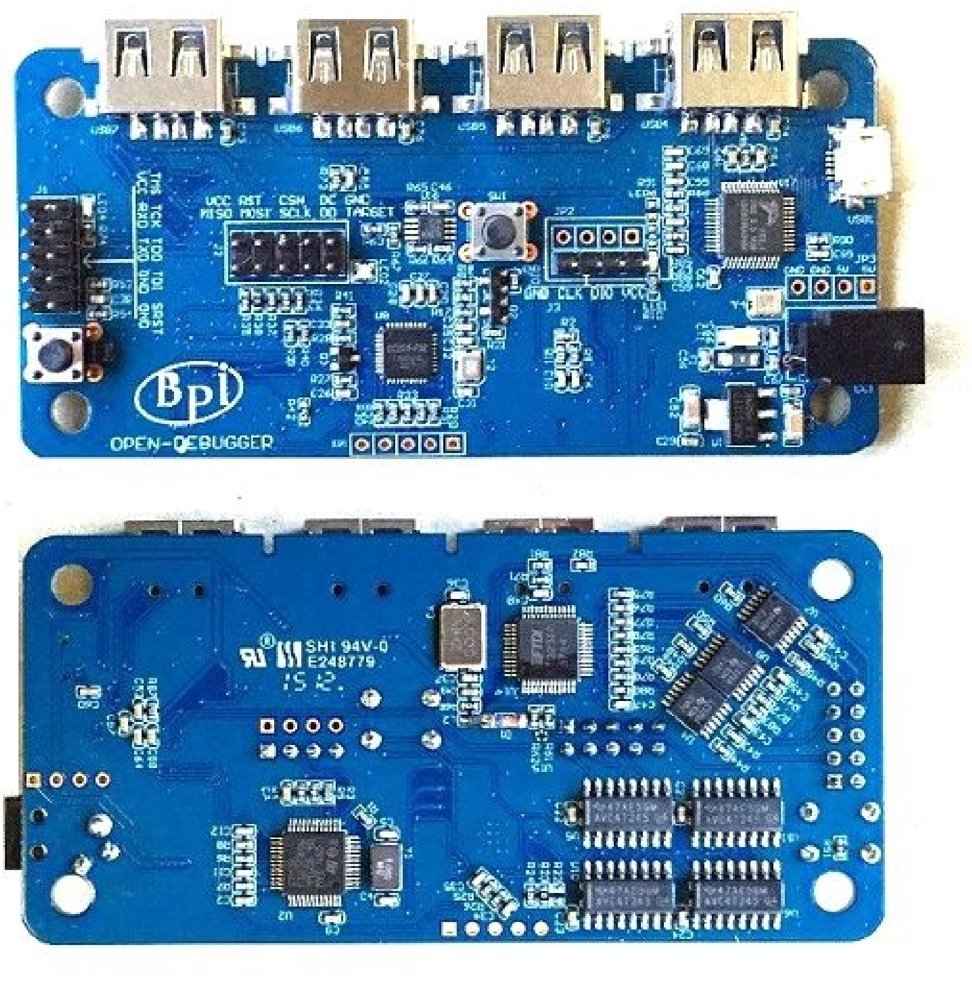
Table of Contents

BPI Open debugger burn board

Open Debugger is a set of programming tools in the development of STM series of microcontrollers, TI CC1x, 2x, TI CC3x series SOC, when STC Series MCU can be programmed quickly and easily debug, debugging programmer does not have to do something different series of chips switching plug tedious work. In addition, it also integrates a USB to serial port, which is a common interface for embedded development. It is also one of USB2.0 HUB, making it easy to extend to other USB tools. With it, no longer need to face a bunch of different writer, complicated mess of wiring, USB port deficiencies.



forum: http://www.banana-pi.org

product: http://www.banana-pi.com

easy to buy a sample :

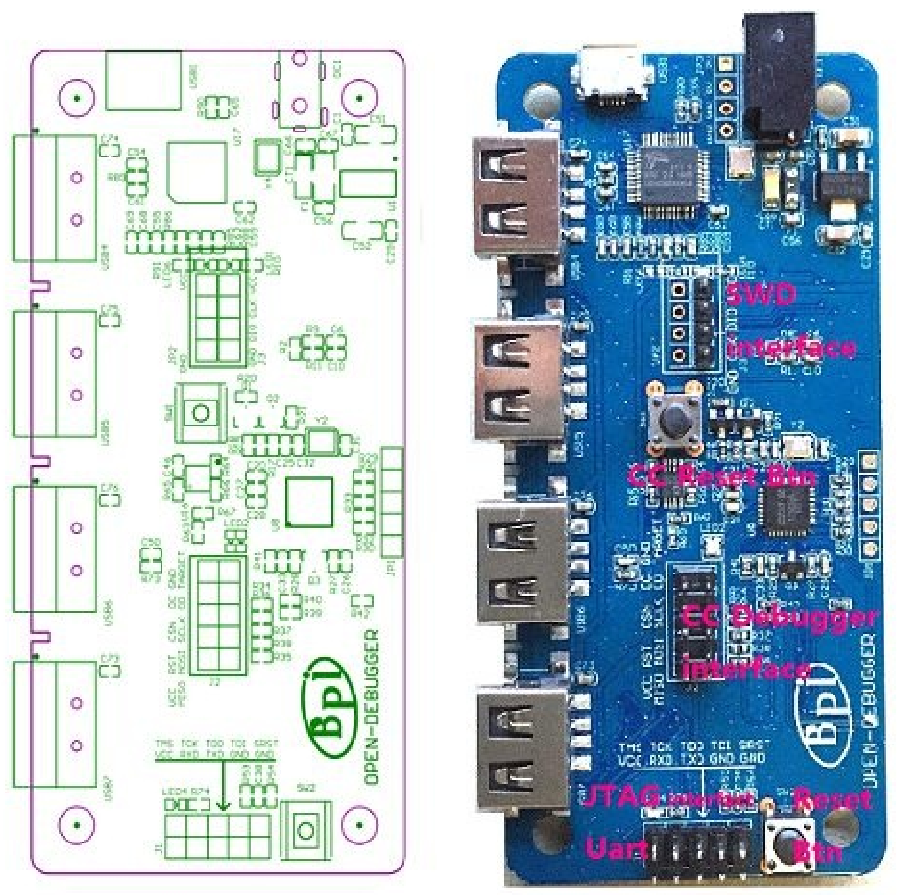
AliExpress

BPI Open debugger burn board Abbreviations

CSn	Chip Select (active low)
DC	Debug Clock
DD	Debug Data
RST	Debug Reset
DUT	Device Under Test
GND	Ground
LED	Light Emitting Diode
MISO	Master In Slave Out
MOSI	Master Out Slave In
RF	Radio Frequency
SCLK	Serial Clock
SoC	System-on-Chip
SPI	Serial Peripheral Interface
USB	Universal Serial Bus
Target	Positive voltage on target
CLK	SWD Clock
DIO	SWD DATA Input/Output
TMS	JTAG Test Mode Select
TCK	JTAG Test Clock Input
TDO	JTAG Test Data Output
TDI	JTAG Test Data Input
SRST	JTAGTest Reset Input, Active Low
RXD	USB Serial Input
TXD	USB Serial Output

Operating condition

1. Operating Voltage：DC 5V
2. ting temperature：0°C to 85°C
3. The onboard voltage output:：3.3 V
4. Maximum output current：2000 mA
5. Supported operating systems：
    - Microsoft® Windows® 2000
    - Windows XP SP2/SP3 operating system (32-bit versions)
    - Windows Vista® operating system (32 and 64 bit)
    - Windows 7 operating system (32 and 64 bit)
    - Windows 8 operating system (32 and 64 bit)



CC Debugger Interface：

Programming and debugging of the following 8051-based TI SoCs：

CC1110，CC1111

CC2430，CC2431

CC2510，CC2511 CC2530，CC2531，CC2533

CC2540，CC2541

CC2543，CC2544，CC2545

To SmartRF Studio control same apparatus. In addition, you can also control the following transceiver SmartRF：

CC1120，CC1121，CC1125，CC1175

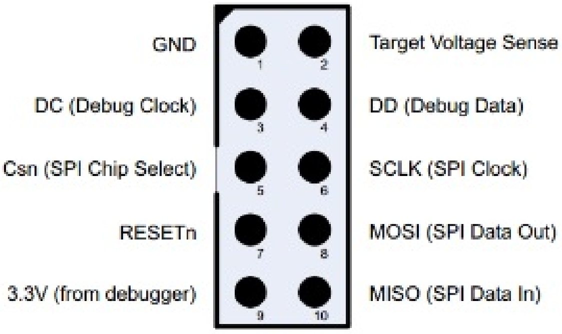
CC1100，CC1101，CC110L，CC113L，CC115L

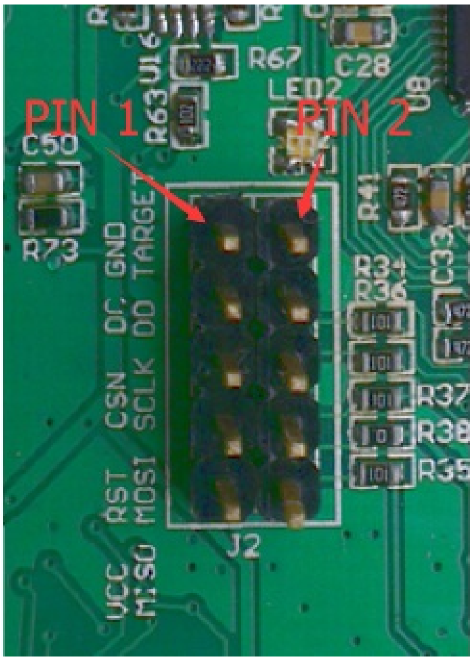
CC2500，CC2520

It can also be used to program the PurePath wireless devices：

CC8520，CC8521

CC8530，CC8531



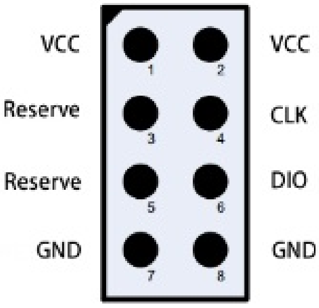


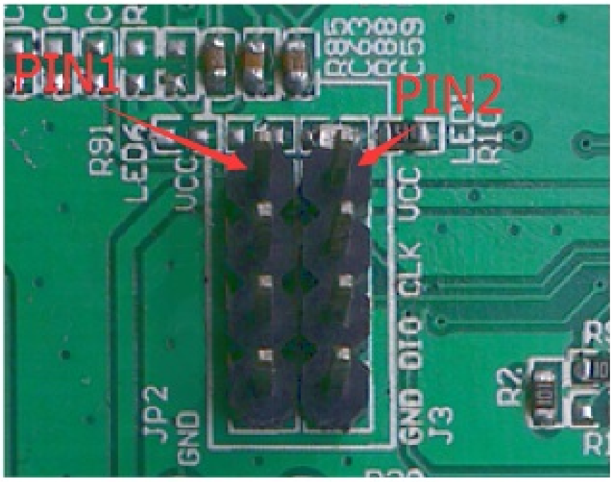
SWD Interface：

Programming and debugging mainstream ARM7 \ 9 \ 11, Cortex-A5 \ A8 \ A9 \ M0 \ M1 \ M3 \ M4

Supported MCU manufacturers： ST，Freecale，Nuvoton，NXP，TI，Cypress，Atmel，Analog，Fujitsu，Toshiba，Energy Micro...。

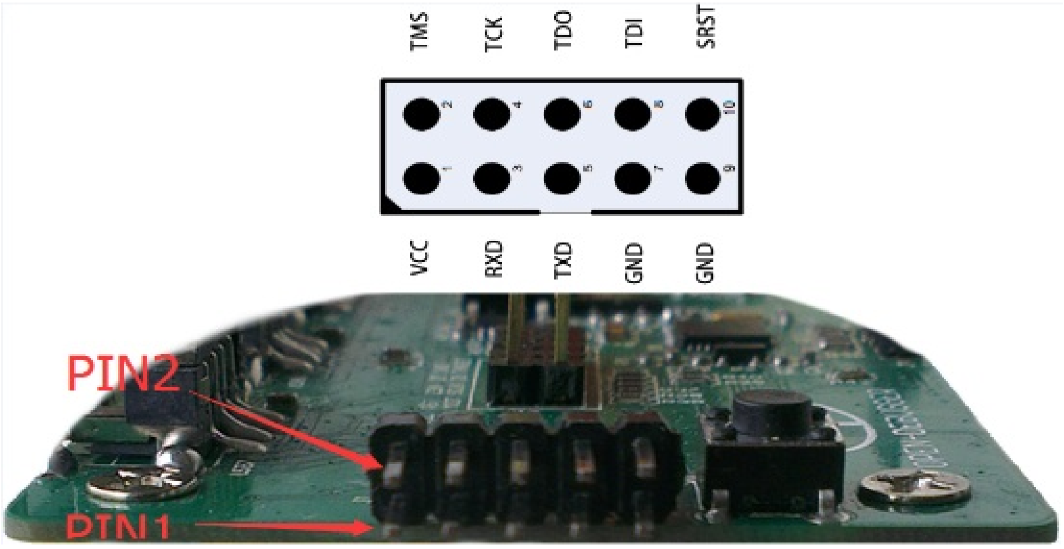
Speeds up the programming and debugging functions 4Mhz: Write Flash Rom, EEPROM, ARF, and support for full speed, stepping, breakpoints





JTAG/Uart Interface：

JTAG programming and debugging interface for all supported kernel。



UART Interface：

STC series microcontroller programming and debugging output port for printing information.

Installing the USB device driver

Install the following software will automatically install Open Debugger driver。

CC Debugger：

SmartRF Studio www.ti.co m/tool/smartrftm-studio

SmartRF Flash Programmer www.ti.com/tool/flash-programmer

SmartRF Packet Sniffer www.ti.com/tool/packet-sniffer

PurePath Wireless Conf igurator www.ti.com/tool/purepath-wl-cfg

PurePath Wireless Commander www.ti.com/tool/purepath-wl-cmd

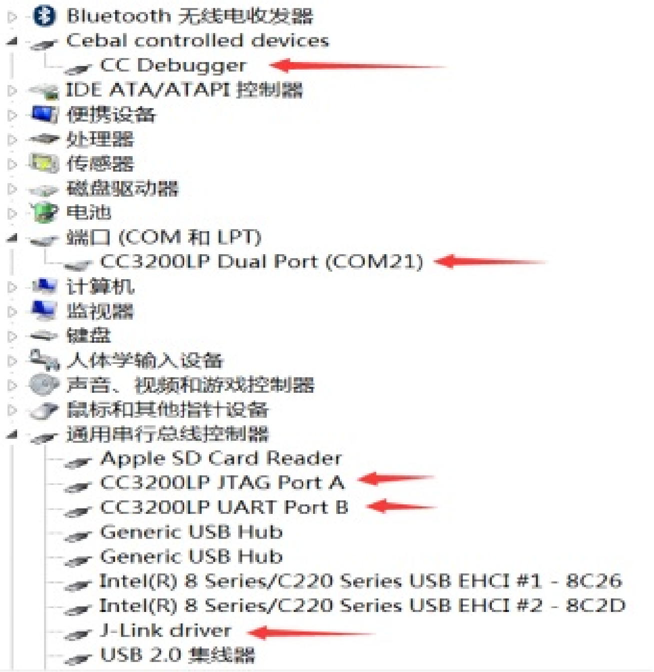
SWD：

JLinkARM www.segger.com/jlink-software.html

JTAG/USB2UART：

Uniflash www.ti.com.cn/tool/cn/uniflash

When all the device drivers installed successfully will be shown below：



```
▷ Bluetooth 无线电收发器
◢ Cebal controlled devices
    CC Debugger
▷ IDE ATA/ATAPI 控制器
▷ 便携设备
▷ 处理器
▷ 传感器
▷ 磁盘驱动器
▷ 电池
◢ 端口 (COM 和 LPT)
    CC3200LP Dual Port (COM21)
▷ 计算机
▷ 监视器
▷ 键盘
▷ 人体学输入设备
▷ 声音、视频和游戏控制器
▷ 鼠标和其他指针设备
◢ 通用串行总线控制器
    Apple SD Card Reader
    CC3200LP JTAG Port A
    CC3200LP UART Port B
    Generic USB Hub
    Generic USB Hub
    Intel(R) 8 Series/C220 Series USB EHCI #1 - 8C26
    Intel(R) 8 Series/C220 Series USB EHCI #2 - 8C2D
    J-Link driver
    USB 2.0 集线器
```

Supported PC Tools

Current CC Debugger supports the following software：

- IAR Embedded Workbench for 8051 In circuit debugging of sys tem-on-chips
- SmartRF Flash Programmer Flash programming of sy stem-on-chips
- SmartRF Studio RF testing of radio devices (transceiv ers and SoCs)
- SmartRF Packet Sniffer Packet sniffing with selected radio devices
- PurePath Wireless Configurator Programming of CC85xx devices
- PurePath Wireless Commander Adv anced control of CC85xx devices

Emulator is between devices and RF device above the software before using the emulator download, please Be sure to connect the RF device and PC。

SWD interface currently supports the following software：

- J-Flash
- Keil uVision4 IDE（MDK-ARM IDE）
- IAR EWARM
- CoIDE
- TASKING
- mikroC Pro for ARM

JTAG interface currently supports the following software：

- IAR EWARM
- IAR EW8051
- Keil uVision4 IDE（MDK-ARM IDE）
- Uniflash
- Code Composer Studio
- OpenOCD

UART interface currently supports the following software：

- STC-ISP
- Putty
- SSCom

CC- Debugger connected to the SoC device

Minimum connection used to debug download

For a TI 8051-based RF SoC, debugging data connection (DD) line, debug clock (DC) line and the reset signal RST line to the device. Note that, DD is a bidirectional signal. In addition, Target and GND must be connected to the target board. Target as input, with the CC Debugger level shifter, thereby allowing the target in addition to the inner portion of the debugger different operating voltages.

For CC111x, CC251x, CC243x, CC253x and CC254x, except CC2544 and CC2545, connection DD and DC signal pins P2.1 pin P2.2. For the CC2544's DD and DC signals are c onnected to P1.3 to P1.2. For the CC2545's DD and DC signals are connected to P1.3 to P1.4. Note that you can connect 3.3 V signal is connected to the target board power supply from the debugger and the target board via pin 9 。

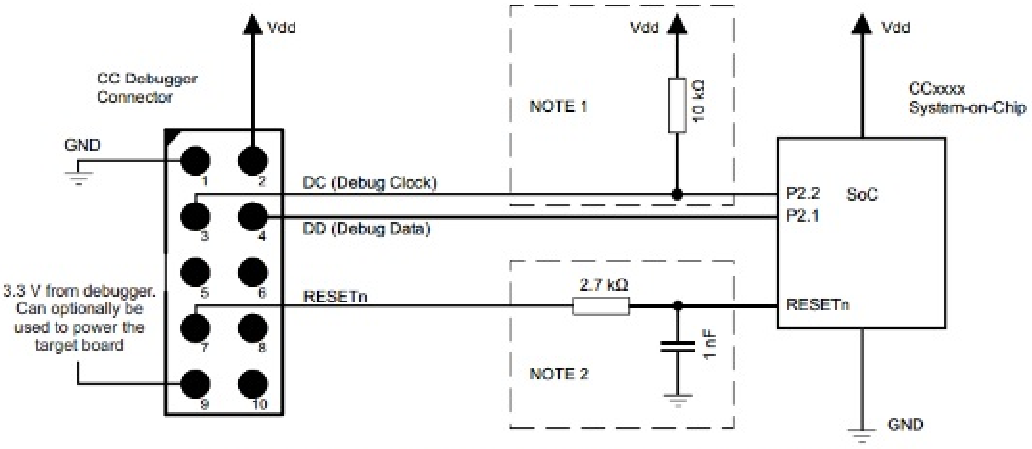


Figure 6. Minimum Connection for Debugging of 8051 SoC

Note

1. Some early SoCs (CC2430, CC2510 and CC1110) requires an external pullup resistor.All new SoC chip has been integrated in the P2.2 pin of the pull, so the external pullup resistor is not needed.
2. TI recommends connecting RC filter circuit on the RESET, thus increasing the reliability of the system, in fact, can also be directly connected to RESET.

SmartRF Studio protocol analysis using the minimum connection

Use with minimum connection used to debug download the same connection.

SmartRF Packet Sniffer protocol analysis using the minimum connection

In order to use CC Debugger protocol analysis, you need to connect the rest of the SPI interface. SPI interface is used to read Take the captured RF packets.

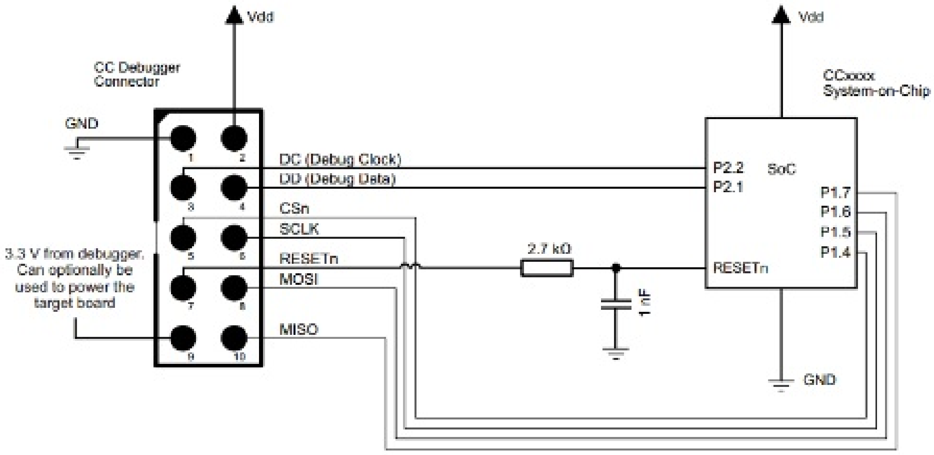


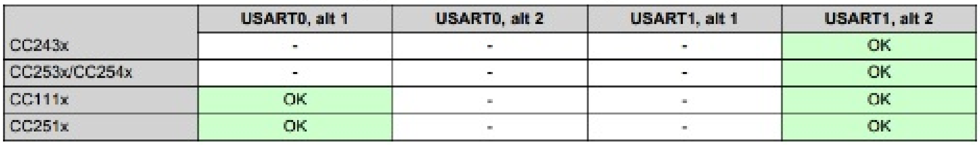
Figure 7. Connection to SoC to Enable Packet Sniffing

Note

1. Use PacketSniffer function will overwrite the original program in the chip.
2. SPI interface Cautions：TI all current SoCs can be configured as SPI slave mode, packetsniffer firmware configures the SoC. Used with CC-Debugger communications protocol analyzer is used in the following diagram port configuration. ◦

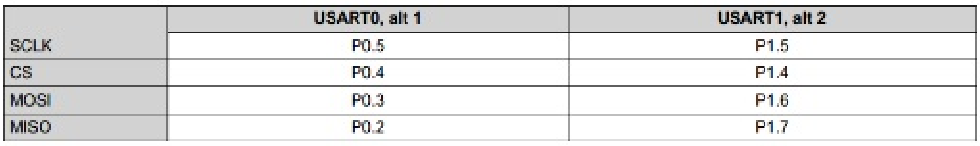
All current TI's RF SOC can be configured to operate as a slave SPI, SPI signals (CS, SCLK, MISO and MOSI) with one USART interface as the data output. SmartRF Packet Sniffer would firmware programmed into the chip, the firmware will choose which one to go according to the SPI serial peripheral PIN pin as an output connection. The firmware can use any of the four pin configuration (USART0 or a pin output substitute 1 or 2). However, at the same time can only use one (see table below).

Table 1. Supported SPI Connections (marked OK)



	USART0, alt 1	USART0, alt 2	USART1, alt 1	USART1, alt 2
CC243x	-	-	-	OK
CC253x/CC254x	-	-	-	OK
CC111x	OK	-	-	OK
CC251x	OK	-	-	OK

Table 2. USART Pin Out Details



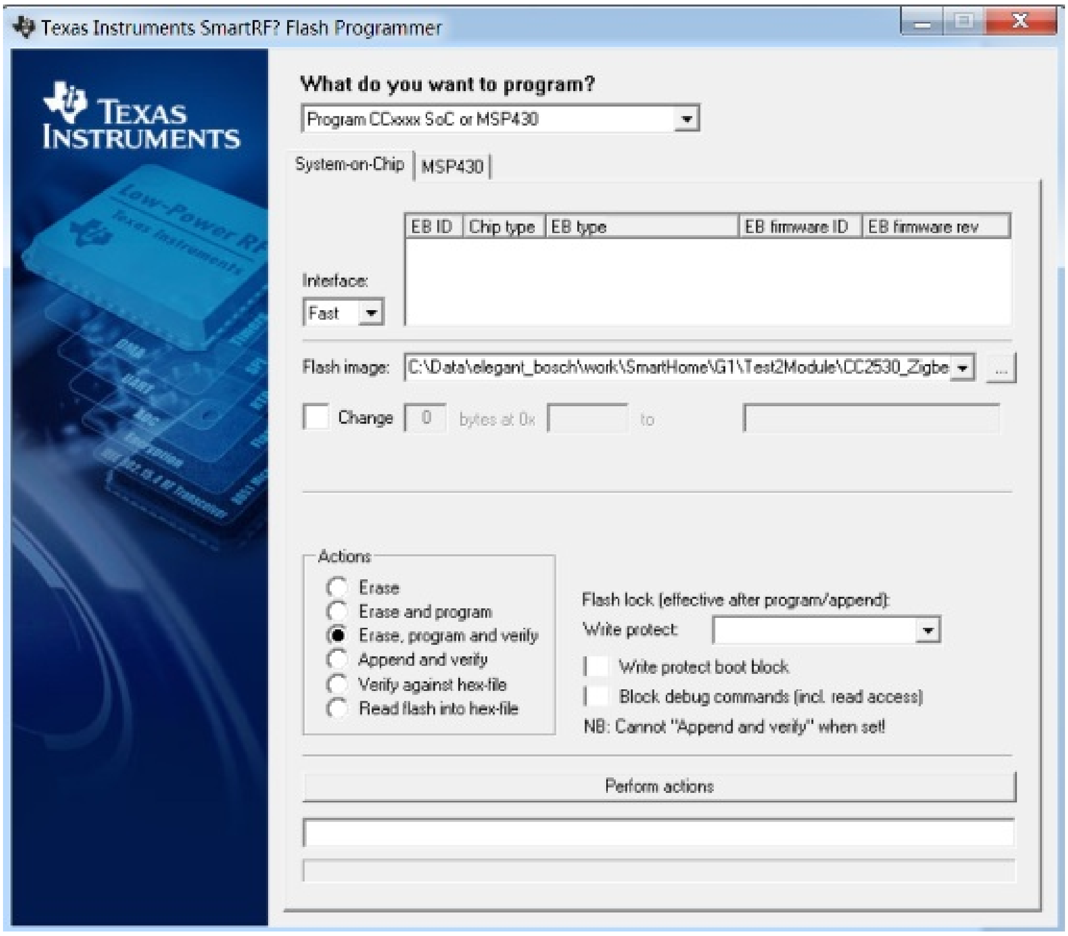
	USART0, alt 1	USART1, alt 2
SCLK	P0.5	P1.5
CS	P0.4	P1.4
MOSI	P0.3	P1.6
MISO	P0.2	P1.7

If you support multiple interfaces, SmartRF Packet Sniffer allows you to choose which interface to use. ◦

Use SmartRFProg tools to program the CC series chips

To illustrate, as an example is used here CC2530

1.First open SmartRFProg software, which can be downloaded separately, can also be found in the bin directory under SmartRF Studio installation directory, open the rear as shown below.



Texas Instruments SmartRF? Flash Programmer

What do you want to program?

Program CCxxxx SoC or MSP430

System-on-Chip | MSP430

EB ID	Chip type	EB type	EB firmware ID	EB firmware rev

Interface: Fast

Flash image: C:\Data\elegant_bosch\work\SmartHome\G1\Test2Module\CC2530_Zigbe

□ Change 0 bytes at 0x to

Actions
- ○ Erase
- ○ Erase and program
- ● Erase, program and verify
- ○ Append and verify
- ○ Verify against hex-file
- ○ Read flash into hex-file

Flash lock (effective after program/append):

Write protect:

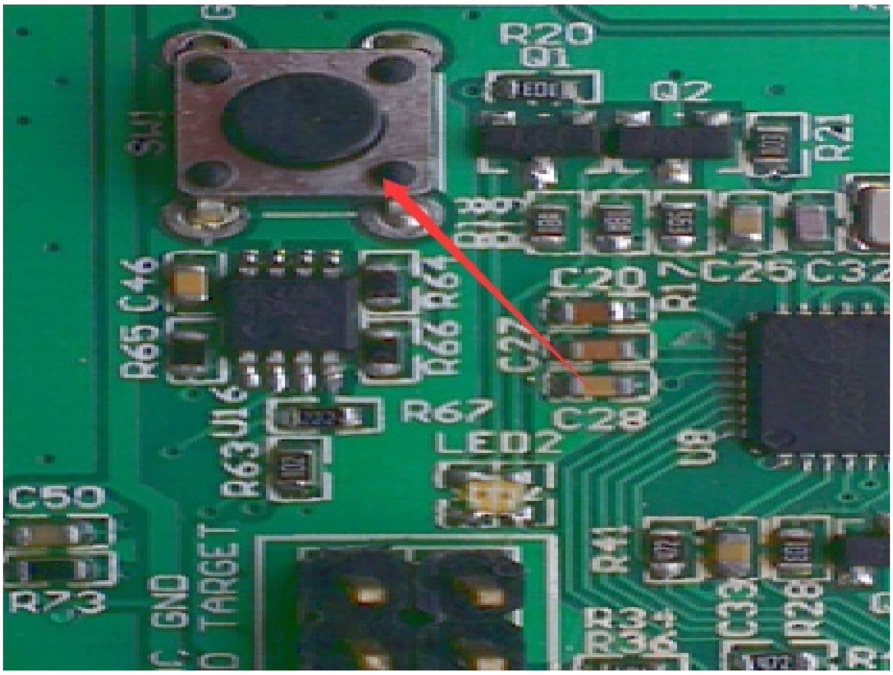
□ Write protect boot block

□ Block debug commands (incl. read access)

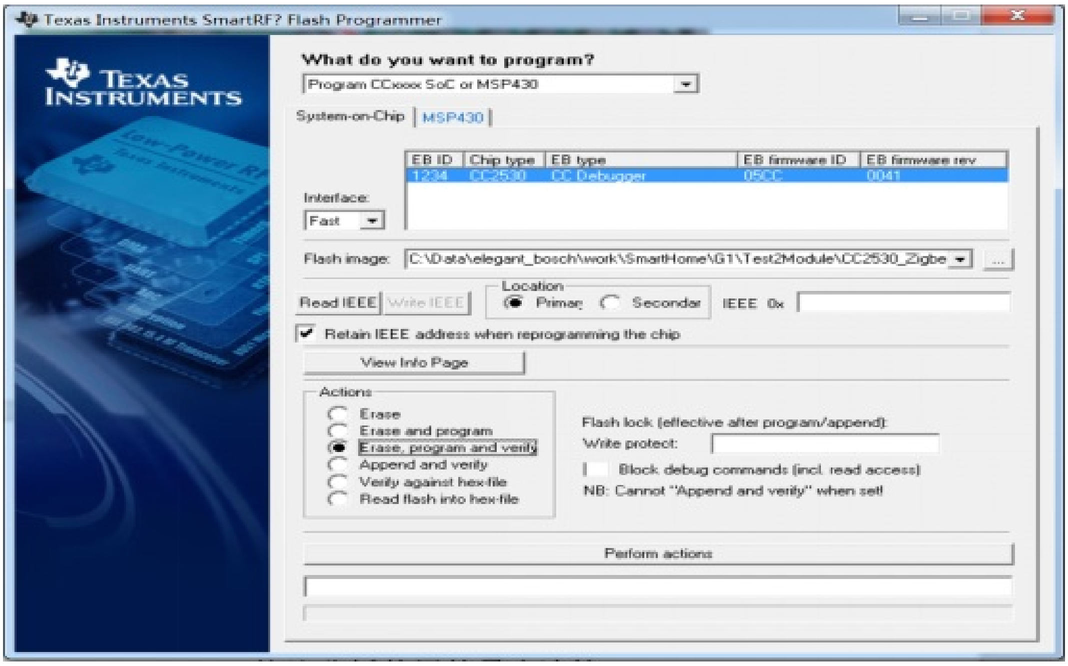
NB: Cannot "Append and verify" when set!

Perform actions

2.Then the device is connected to the interface with the CC2530 in CCdebugger Open Debugger on. There need to connect the line VCC, GND, DC, DD, Target, RST which a few lines. After connecting, press the reset button。



3.Under normal circumstances, the state next to the CC Debugger will turn green, if not green, such as red, orange, or other circumstances, please check the wiring is correct, the power supply is normal. If the status is green, then the device list of scanned SmartRFProg equipment will appear as shown below.



What do you want to program?

Program CCxxxx SoC or MSP430

EB ID	Chip type	EB type	EB firmware ID	EB firmware rev
1234	CC2530	CC Debugger	05CC	0041

Interface: Fast

Flash image: C:\Data\elegant_bosch\work\SmartHome\G1\Test2Module\CC2530_Zigbe

Location: ● Primar ○ Secondar

IEEE 0x

☑ Retain IEEE address when reprogramming the chip

Actions
- ○ Erase
- ○ Erase and program
- ● Erase, program and verify
- ○ Append and verify
- ○ Verify against hex-file
- ○ Read flash into hex-file

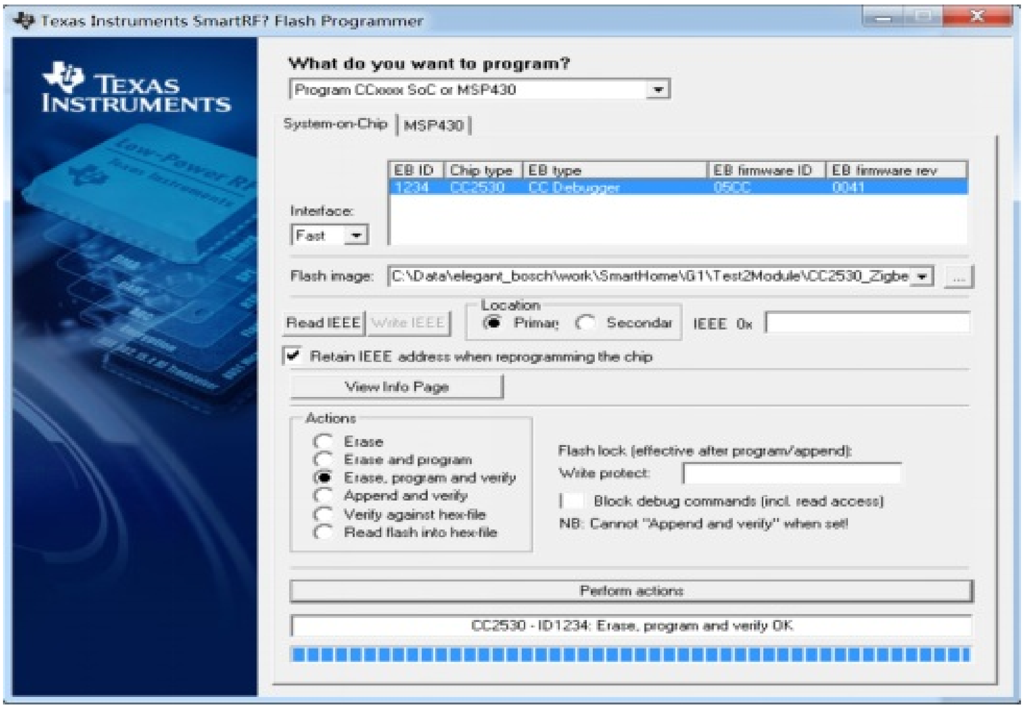
Flash lock (effective after program/append):

Write protect:

□ Block debug commands (incl. read access)

NB: Cannot "Append and verify" when set!

4.Then in the next Flash image input box below the list of devices have a button, click to select the firmware file to be programmed, the following Actions select Erase, program and verify, after everything is ready you can click the button below the long long Perfrom action. If the write error, it will output to the progress bar the wrong reasons, and if successful will be similar to the following programming.



Texas Instruments SmartRF? Flash Programmer

What do you want to program? Program CCxxxx SoC or MSP430

System-on-Chip | MSP430

EB ID	Chip type	EB type	EB firmware ID	EB firmware rev
1234	CC2530	CC Debugger	05CC	0041

Interface: Fast

Flash image: C:\Data\elegant_bosch\work\SmartHome\G1\Test2Module\CC2530_Zigbe

Read IEEE | Write IEEE

Location: ● Primary ○ Secondary

IEEE 0x

☑ Retain IEEE address when reprogramming the chip

View Info Page

Actions

○ Erase
○ Erase and program
● Erase, program and verify
○ Append and verify
○ Verify against hex-file
○ Read flash into hex-file

Flash lock (effective after program/append):

Write protect:

□ Block debug commands (incl. read access)

NB: Cannot "Append and verify" when set!

Perform actions

CC2530 - ID1234: Erase, program and verify OK

CC-Debugger connection and Transceivers Transceiver

SPI CC Debugger interface can be controlled by a variety of CCxxxx SmartRF Studio transceiver, when the transmitter and receiver front support：

CC1100 CC1101 CC1120 CC1121 CC1125 CC1175 CC110L CC113L CC115L CC1200 CC1201 CC2500 CC2520

Note:CC Debugger in SPI Master Mode In a multi-host, please make sure not to other SPI Master conflict. Wiring diagram below：

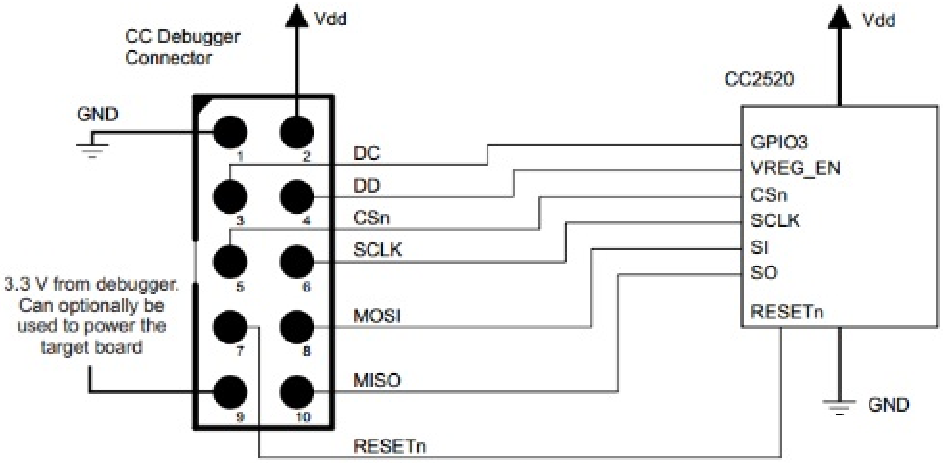


Figure 8. CC Debugger Connected to CC2520

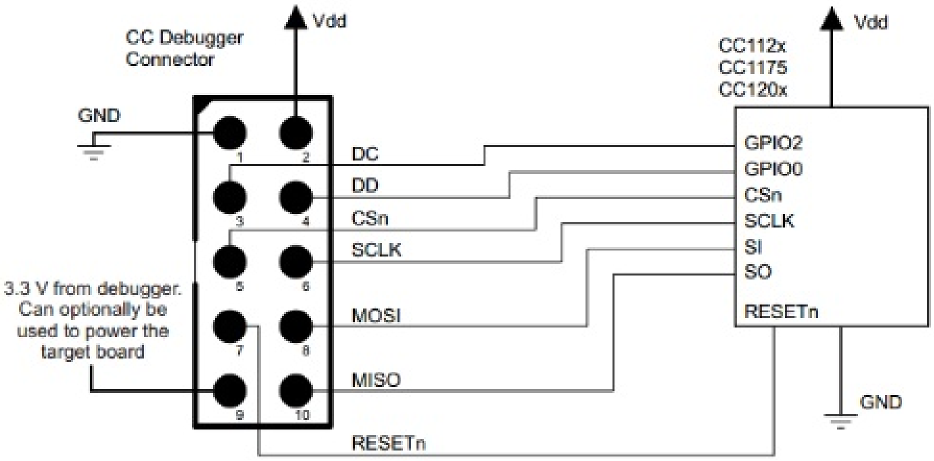


Figure 9. CC Debugger Connected to CC112x/CC1175/CC120x

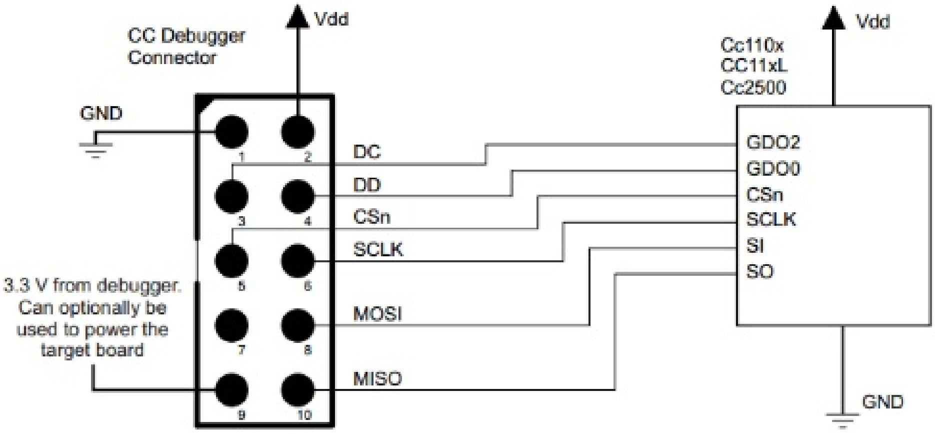


Figure 10. CC Debugger Connected to CC110x/CC11xL/CC2500

Connection CC Debugger and CC85xx

To be able to use PurePath wireless configuration, CC Debugger and CC85xx SPI interface must be connected as shown below：

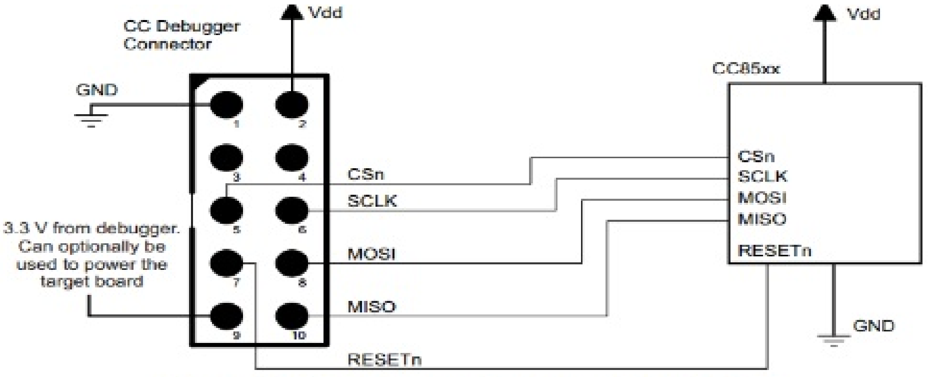
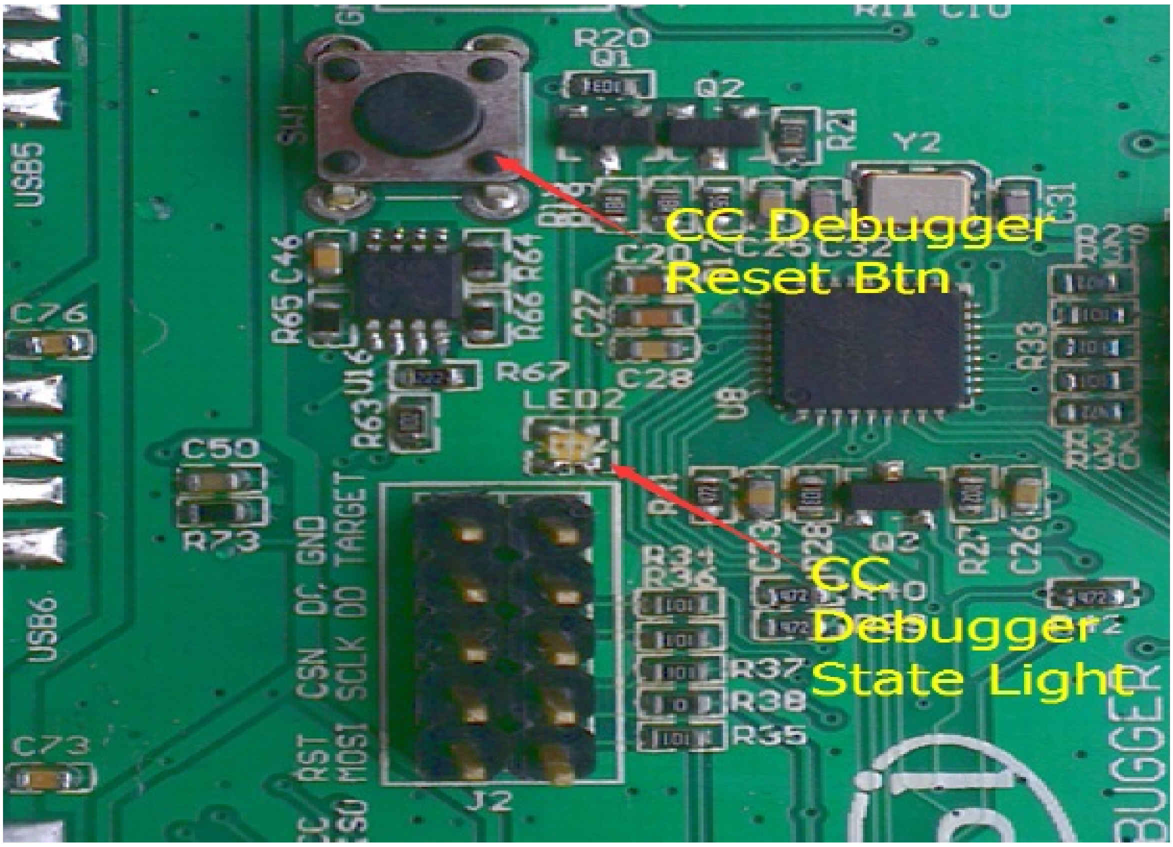


Figure 11. CC Debugger Connected to CC85XX

Open Debugger CC Debugger module functions in use

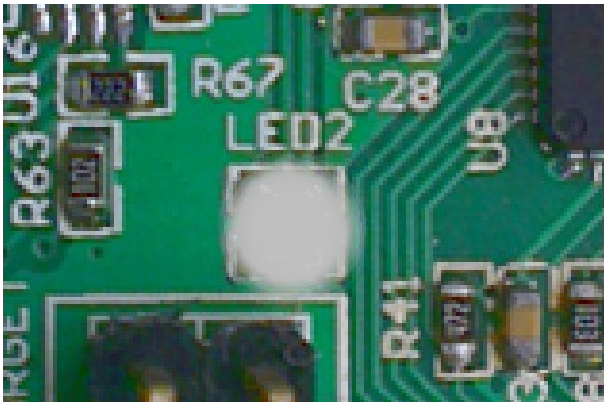
After CC Debugger interface to connect to the target device, you need to press the reset button next to (see below), it will automatically find all known devices. If you do not detect the device, LED will turn red. If the device is detected, LED will be green. If the LED is green, indicating that you can be able to meet the PC side software use。



Note：that only when the status light is green CC Debugger to debug and download work, if it is red, please press the reset button, and check the connection with the target chip, until the status light turns green。

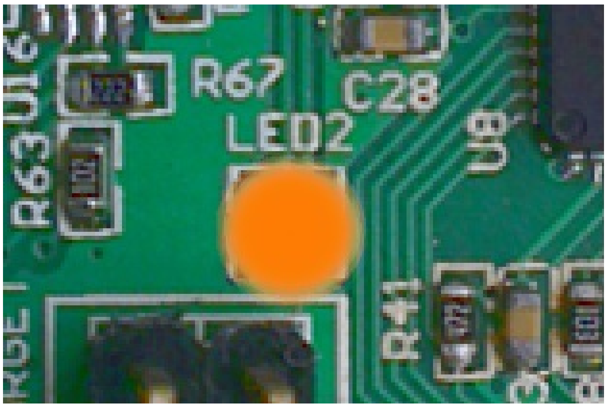
CC Debugger LED light

OFF



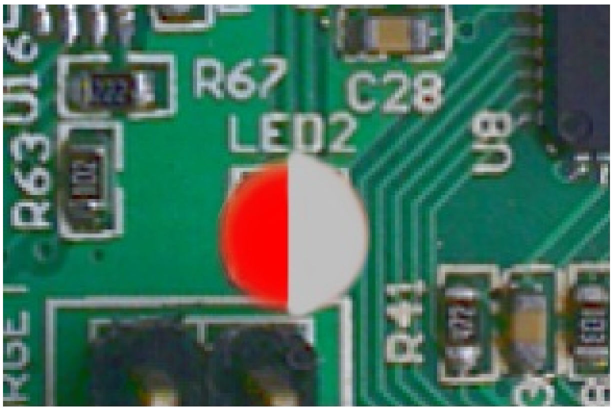
Emulator is not powered, the latter emulator firmware has expired.

A MBER (BOTH LEDS ON)



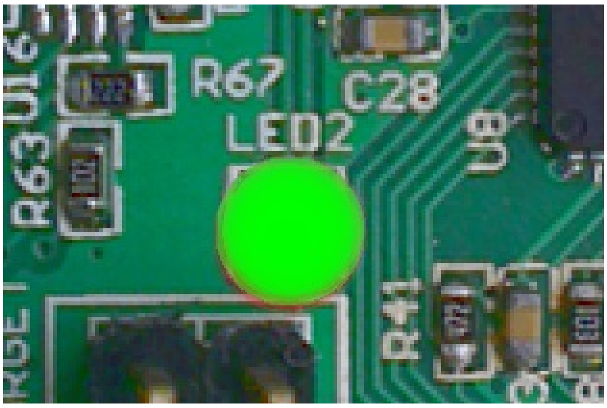
Emulators have the power, but without the right emulator firmware.

RED LED Flashing



Emulator enters firmware recovery mode

ON



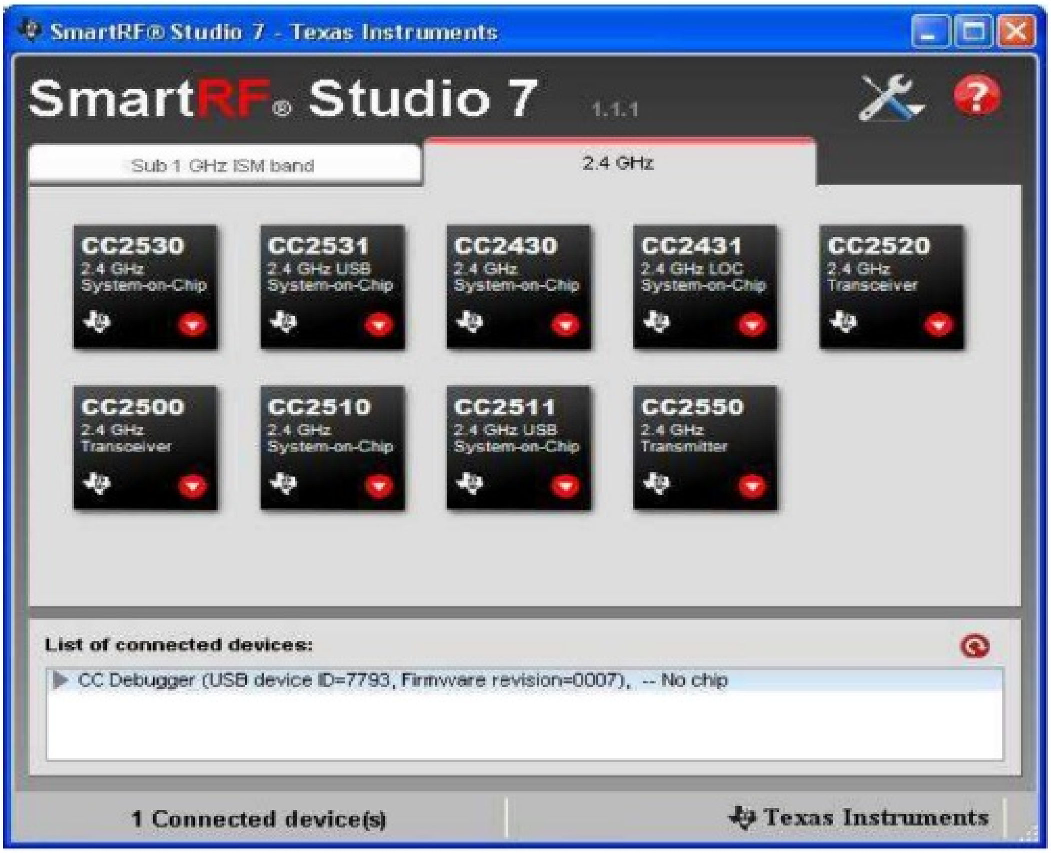
Have been identified to the chip, can be the next step

CC Debugger firmware update

Emulator firmware update by SmartRF Studio

1 Start SmartRF Studio.

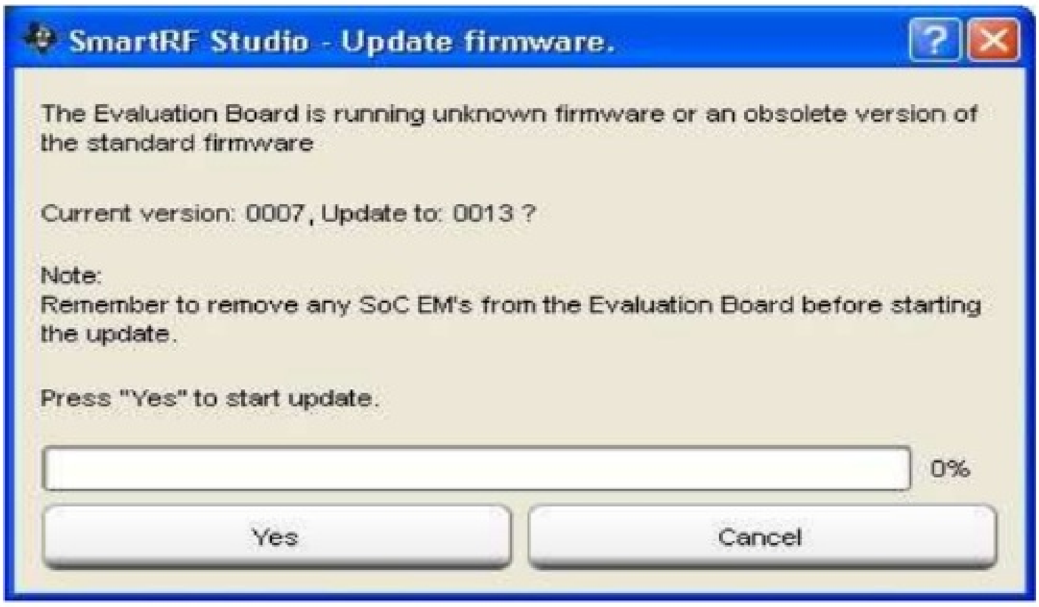
2 Disconnect the emulator with the target board, and then connect to the computer via usb.Emulator will appear in the list of devices SmartRF Studio.


SmartRF® Studio 7 - Texas Instruments
SmartRF® Studio 7
1.1.1
Sub 1 GHz ISM band
2.4 GHz
CC2530 2.4 GHz System-on-Chip
CC2531 2.4 GHz USB System-on-Chip
CC2430 2.4 GHz System-on-Chip
CC2431 2.4 GHz LOC System-on-Chip
CC2520 2.4 GHz Transceiver
CC2500 2.4 GHz Transceiver
CC2510 2.4 GHz System-on-Chip
CC2511 2.4 GHz USB System-on-Chip
CC2550 2.4 GHz Transmitter
List of connected devices:
1 Connected device(s)
Texas Instruments

CC Debugger (USB device ID=7793, Firmware revision=0007), -- No chip

Figure 12. Auto Firmware Upgrade

3.Double-click the device list appears. If there is a new firmware, the following dialog box will appear.



SmartRF Studio - Update firmware.

The Evaluation Board is running unknown firmware or an obsolete version of the standard firmware

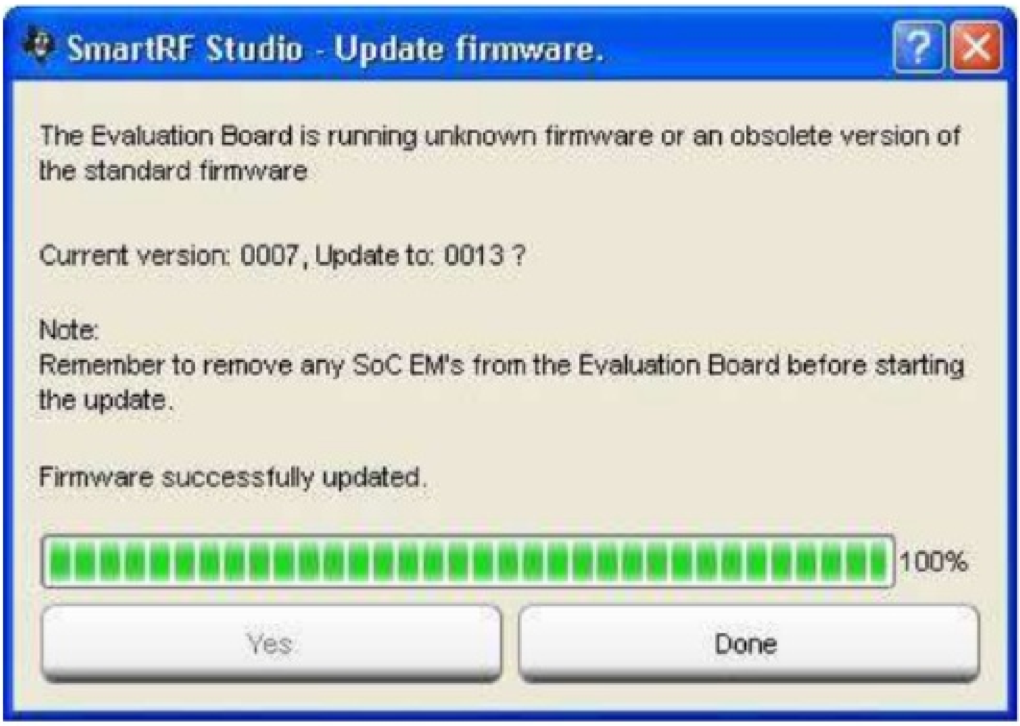
Current version: 0007, Update to: 0013 ?

Note:
Remember to remove any SoC EM's from the Evaluation Board before starting the update.

Press "Yes" to start update.

Figure 13. Auto Firmware Update

4.Select YES, and wait for the rest of the operation is completed SmartRFStudio.



SmartRF Studio - Update firmware.

The Evaluation Board is running unknown firmware or an obsolete version of the standard firmware

Current version: 0007, Update to: 0013 ?

Note:
Remember to remove any SoC EM's from the Evaluation Board before starting the update.

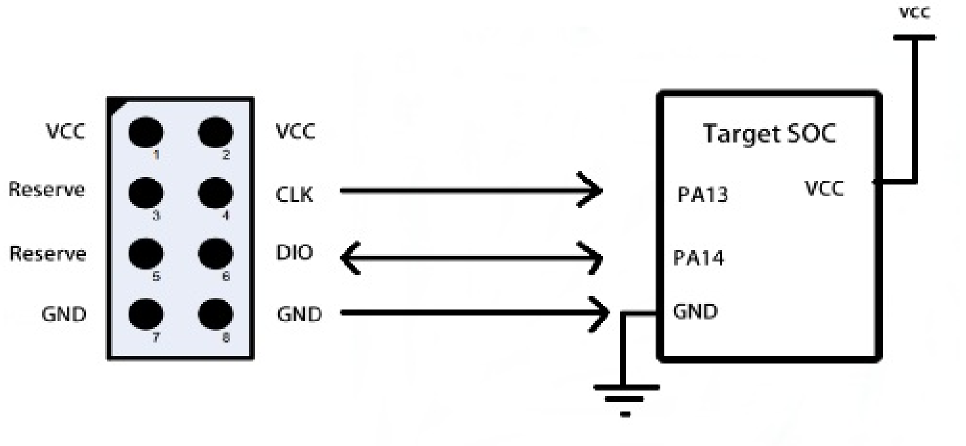
Firmware successfully updated.

5.Select Done after the upgrade.

SWD interface device connected to the SOC

Minimum connection SWD interface

The full name of the SWD interface Serial Wire Debug, which compared to traditional cable JTAG interface of the biggest advantages is the use of fewer, more stable. If you do not consider the VCC supply, then only one CLOCK, a DATA IO line and a ground to achieve high-speed debugging and firmware programming. Minimum connection as shown below：



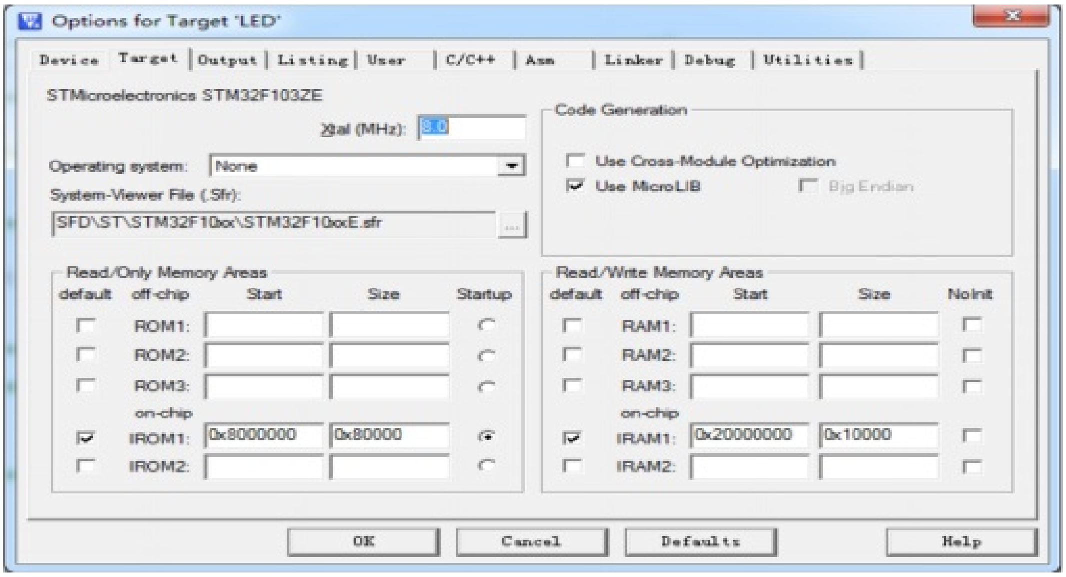
Use SWD debug download the MDK-ARM

To illustrate, here will be used as an example STM32F103C8T6.

SWD interface to download programs in MDK-ARM

1.First need to Open Debugger of SWD interface CLK, DIO, GND STM32F103C8T6 based target board in PA13, PA14 and GND are connected and open MDK-ARM.

2.Click Project, Option for Target'XXX 'or use the corresponding shortcut icon Alt + F7 or click on the toolbar to open the configuration options.



STMicroelectronics STM32F103ZE

Xtal (MHz): 8.0

Operating system: None

System-Viewer File (.Sfr): SFD\ST\STM32F10xx\STM32F10xxE.sfr

Code Generation
- □ Use Cross-Module Optimization
- ☑ Use MicroLIB
- □ Big Endian

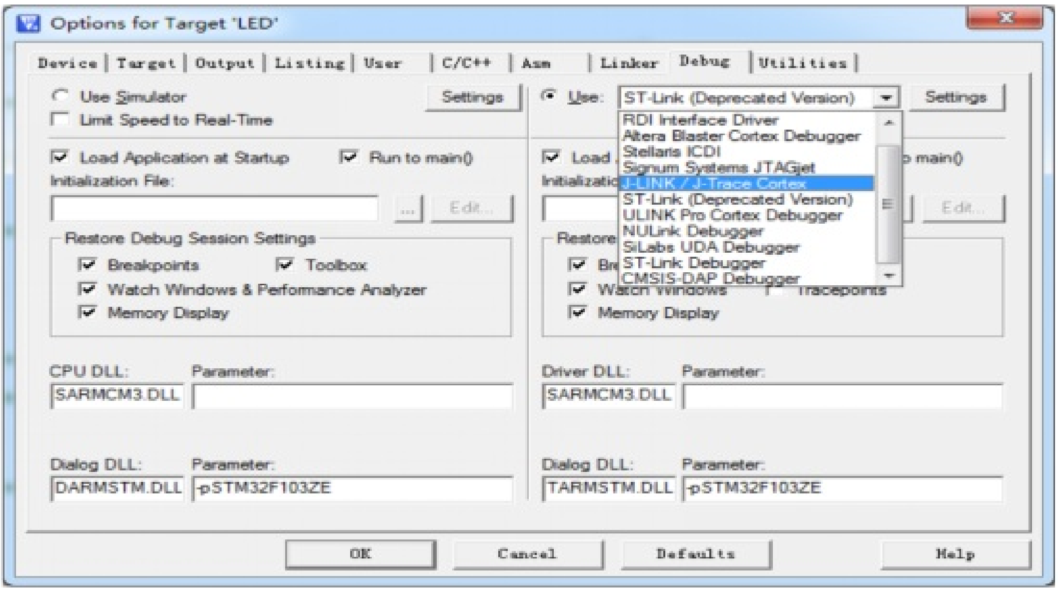
Read/Only Memory Areas

default	off-chip	Start	Size	Startup
□	ROM1:			○
□	ROM2:			○
□	ROM3:			○
	on-chip			
☑	IROM1:	0x8000000	0x80000	●
□	IROM2:			○

Read/Write Memory Areas

default	off-chip	Start	Size	NoInit
□	RAM1:			□
□	RAM2:			□
□	RAM3:			□
	on-chip			
☑	IRAM1:	0x20000000	0x10000	□
□	IRAM2:			□

3.Select the Debug tab, select the J-Link / J-Trace Cortex Options。



○ Use Simulator

□ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Restore Debug Session Settings
- ☑ Breakpoints ☑ Toolbox
- ☑ Watch Windows & Performance Analyzer
- ☑ Memory Display

CPU DLL: SARMCM3.DLL Parameter:

Dialog DLL: DARMSTM.DLL Parameter: -pSTM32F103ZE

● Use: ST-Link (Deprecated Version)

☑ Load ... main()

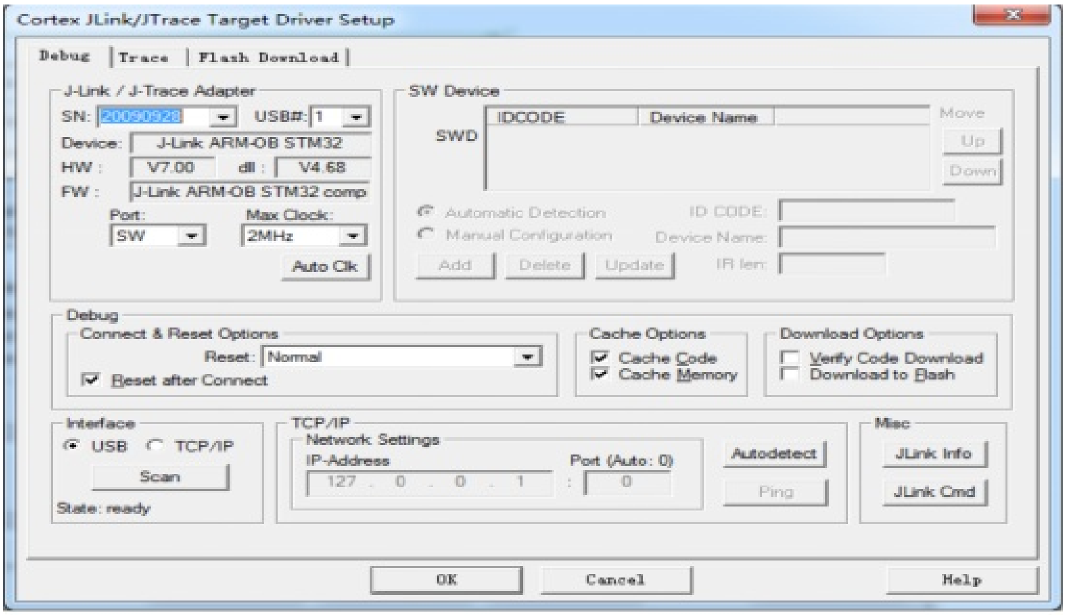
Initializatio...

Restore ...
- ☑ Br...
- ☑ Watch Windows □ Tracepoints
- ☑ Memory Display

Driver DLL: SARMCM3.DLL Parameter:

Dialog DLL: TARMSTM.DLL Parameter: -pSTM32F103ZE

4.Then click the settings button next to enter the Cortex JLink / JTrace Target Driver Setup settings, choose the way in SW Debug tab in Port。



Cortex JLink/JTrace Target Driver Setup

Debug | Trace | Flash Download

J-Link / J-Trace Adapter
SN: 20090928 USB#: 1
Device: J-Link ARM-OB STM32
HW : V7.00 dll : V4.68
FW : J-Link ARM-OB STM32 comp
Port: SW Max Clock: 2MHz
Auto Clk

SW Device
SWD IDCODE Device Name
Move Up Down
● Automatic Detection ID CODE:
○ Manual Configuration Device Name:
Add Delete Update IR len:

Debug
Connect & Reset Options
Reset: Normal
☑ Reset after Connect

Cache Options
☑ Cache Code
☑ Cache Memory

Download Options
☐ Verify Code Download
☐ Download to Flash

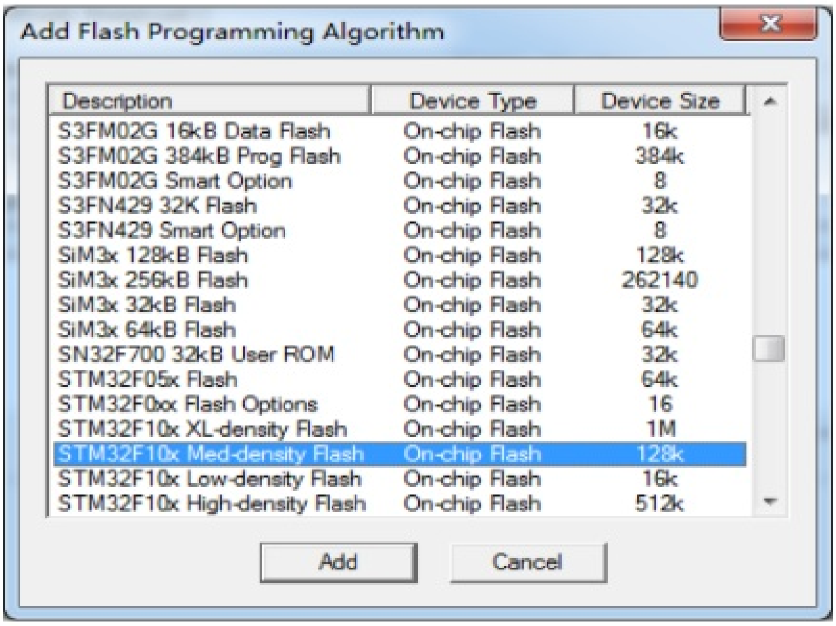
Interface
● USB ○ TCP/IP
Scan
State: ready

TCP/IP
Network Settings
IP-Address 127 . 0 . 0 . 1 Port (Auto: 0) : 0
Autodetect
Ping

Misc
JLink Info
JLink Cmd

OK Cancel Help

5.Adding in Flash Download in Programming Algorithm corresponding algorithms do here is STM32F103C8 SCM instance, select STM32F10x Med-density Flash On-chip Flash 128k algorithm and click Add

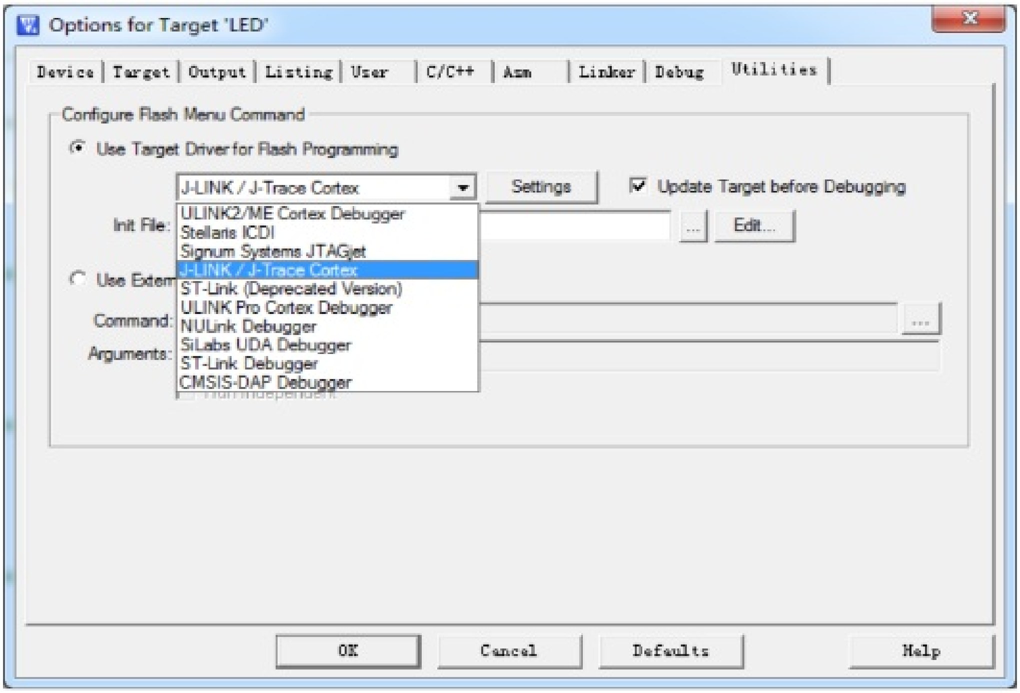


Add Flash Programming Algorithm

Description	Device Type	Device Size
S3FM02G 16kB Data Flash	On-chip Flash	16k
S3FM02G 384kB Prog Flash	On-chip Flash	384k
S3FM02G Smart Option	On-chip Flash	8
S3FN429 32K Flash	On-chip Flash	32k
S3FN429 Smart Option	On-chip Flash	8
SiM3x 128kB Flash	On-chip Flash	128k
SiM3x 256kB Flash	On-chip Flash	262140
SiM3x 32kB Flash	On-chip Flash	32k
SiM3x 64kB Flash	On-chip Flash	64k
SN32F700 32kB User ROM	On-chip Flash	32k
STM32F05x Flash	On-chip Flash	64k
STM32F0xx Flash Options	On-chip Flash	16
STM32F10x XL-density Flash	On-chip Flash	1M
STM32F10x Med-density Flash	On-chip Flash	128k
STM32F10x Low-density Flash	On-chip Flash	16k
STM32F10x High-density Flash	On-chip Flash	512k

Add Cancel

6.Select Use Target for the Utilities tab, select Flash Programming Tool J-Link / J-Trace Cortex, and then click OK to exit。



Options for Target 'LED'

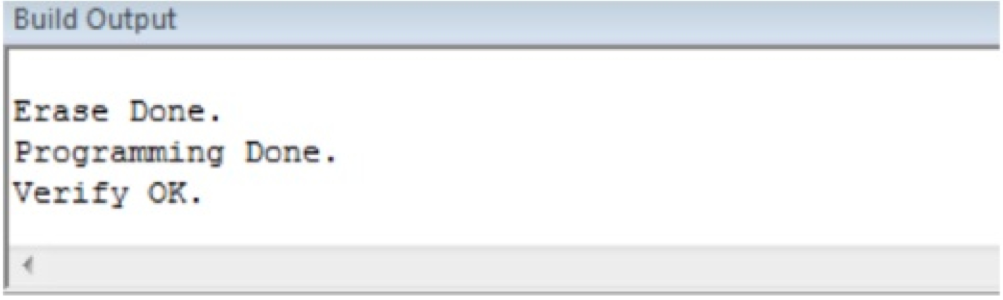
Configure Flash Menu Command

● Use Target Driver for Flash Programming

J-LINK / J-Trace Cortex

☑ Update Target before Debugging

7.At this point, the setup is complete, you can compile the program, such as the error can be able to point LOAD button to start downloading the program to the chip.

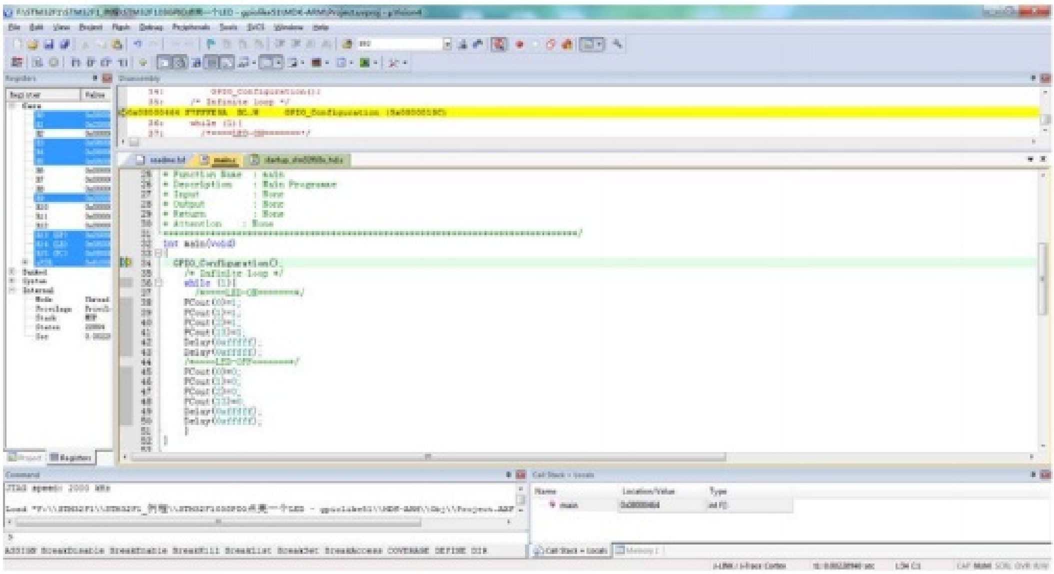


Build Output

```
Erase Done.
Programming Done.
Verify OK.
```

MDK-ARM SWD interface in single-step debugging

1.Click Debugging icon to enter debug mode.



```
34:         GPIO_Configuration();
35:     /* Infinite loop */
[illegible] GPIO_Configuration [illegible]
36:     while (1){
37:       /*=====LED-ON=======*/
```

```
 * Function Name  : main
 * Description    : Main Programme
 * Input          : None
 * Output         : None
 * Return         : None
 * Attention      : None
 *******************************************************************************/
int main(void)
{
  GPIO_Configuration();
    /* Infinite loop */
    while (1){
      /*=====LED-ON=======*/
    PCout(0)=1;
    PCout(1)=1;
    PCout(2)=1;
    PCout(13)=1;
    Delay(0xfffff);
    Delay(0xfffff);
    /*=====LED-OFF=======*/
    PCout(0)=0;
    PCout(1)=0;
    PCout(2)=0;
    PCout(13)=0;
    Delay(0xfffff);
    Delay(0xfffff);
    }
}
```

```
JTAG speed: 2000 kHz

Load "F:\\STM32F1\\STM32F1_例程\\STM32F103RFDO点亮一个LED - [illegible]\\MDK-ARM\\Obj\\Project.AXF"
```

```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR
```

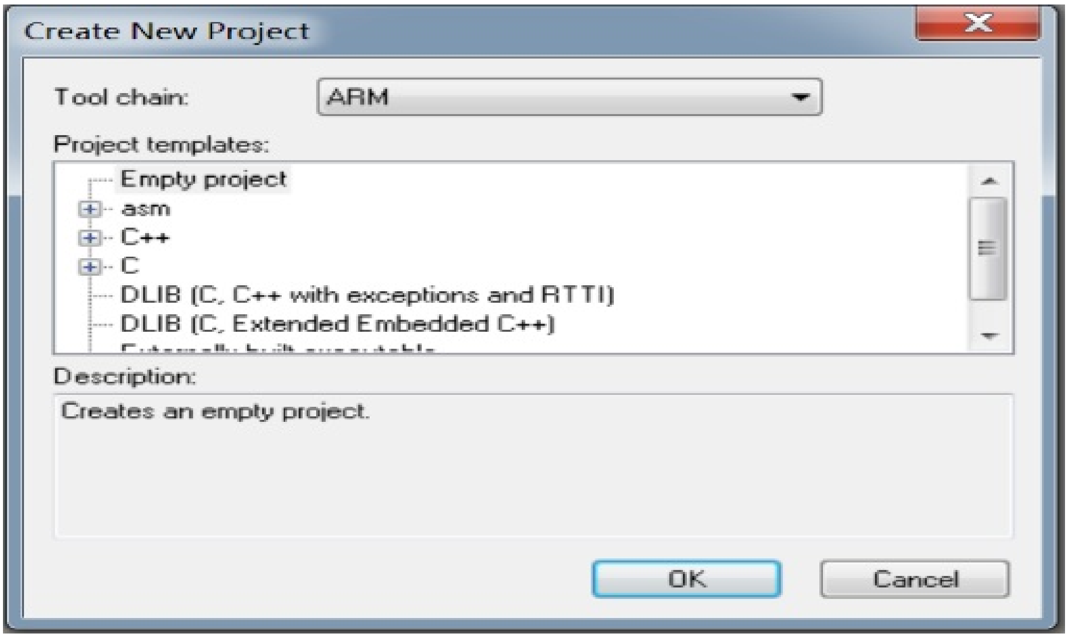
2.After entering the debug mode, the reader step by step debugging can be performed according to their needs.

Download SWD debug using the IAR EWARM

To illustrate, here will be used as an example STM32F103C8T6.

SWD interface to download the program in IAR EWARM

1.Open the IAR EWARM, and create a new project, Tool chain ARM can choose, click OK.



Create New Project

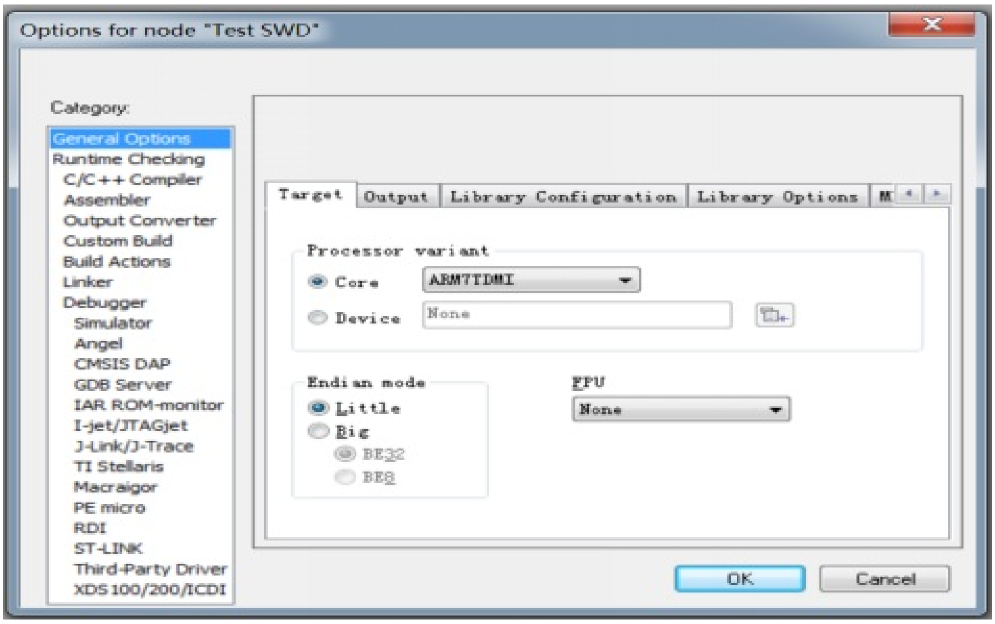
Tool chain: ARM

Project templates: Empty project

Description: Creates an empty project.

Empty project
asm
C++
C
DLIB (C, C++ with exceptions and RTTI)
DLIB (C, Extended Embedded C++)
OK
Cancel

Select the menu bar Project-> Options ..., option box will pop up, as shown below



Options for node "Test SWD"

Category: General Options

Target

Processor variant

● Core ARM7TDMI

○ Device None

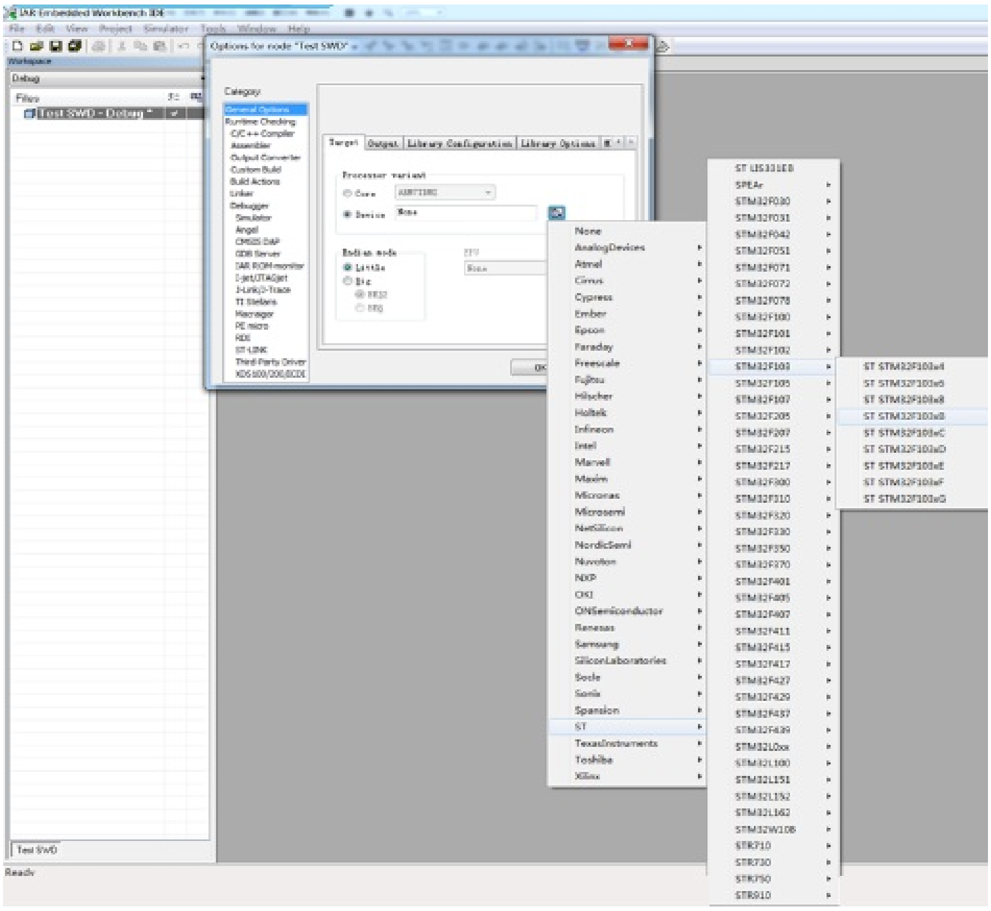
Endian mode

● Little

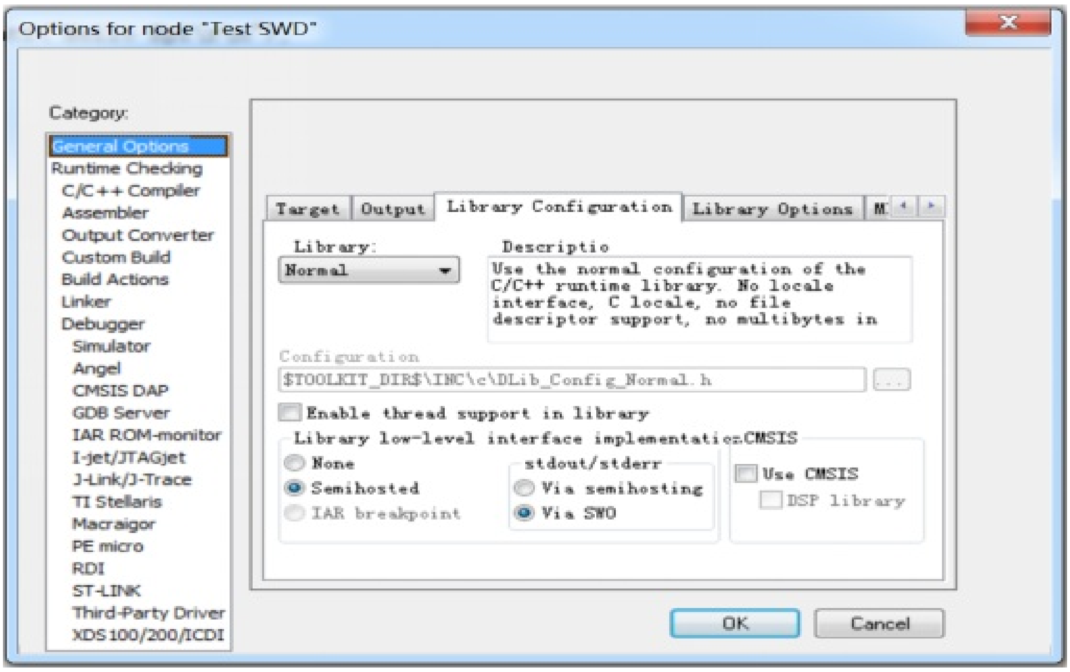
○ Big

FPU None

2.Select General Options -> Target page, then select the Processor variant column in the Device, and then click the right button, and in turn find STM32F103C8 processor.



3.Then choose General Options -> Library Configuration page stdout / stderr section of Via SWO.



Library: Normal

Descriptio Use the normal configuration of the C/C++ runtime library. No locale interface, C locale, no file descriptor support, no multibytes in

Configuration $TOOLKIT_DIR$\INC\c\DLib_Config_Normal.h

□ Enable thread support in library

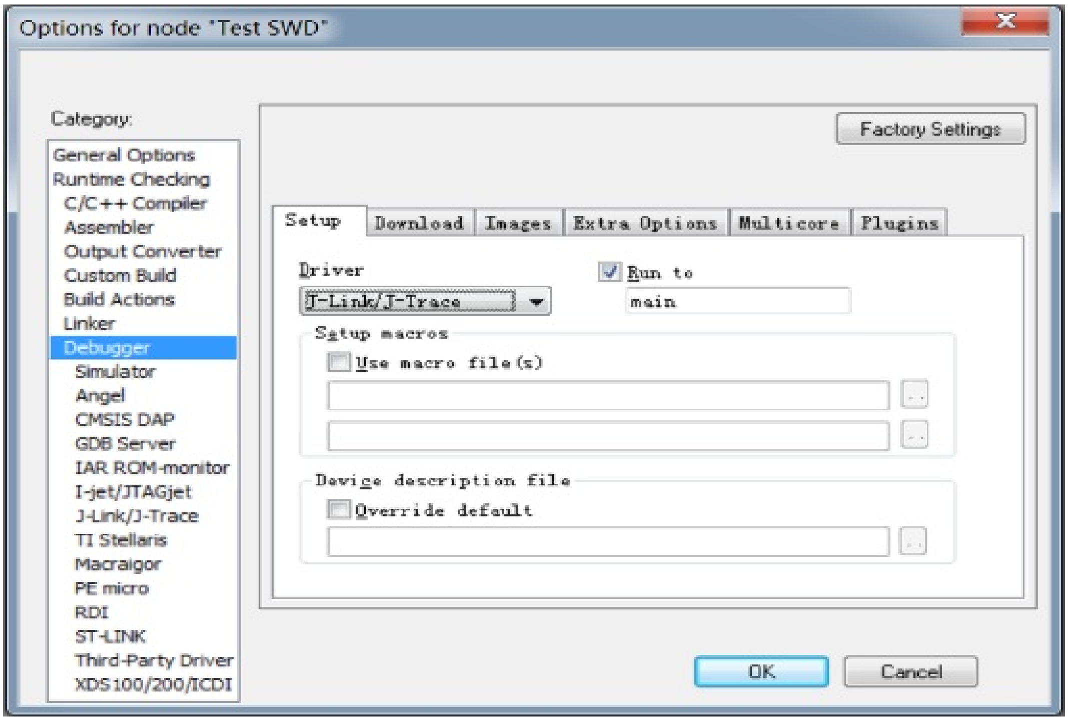
Library low-level interface implementatioCMSIS

○ None
● Semihosted
○ IAR breakpoint

stdout/stderr
○ Via semihosting
● Via SWO

□ Use CMSIS
□ DSP library

4.Then select Debugger -> Setup -> Driver -> J-Link / J-Trace.



Driver: J-Link/J-Trace

☑ Run to: main

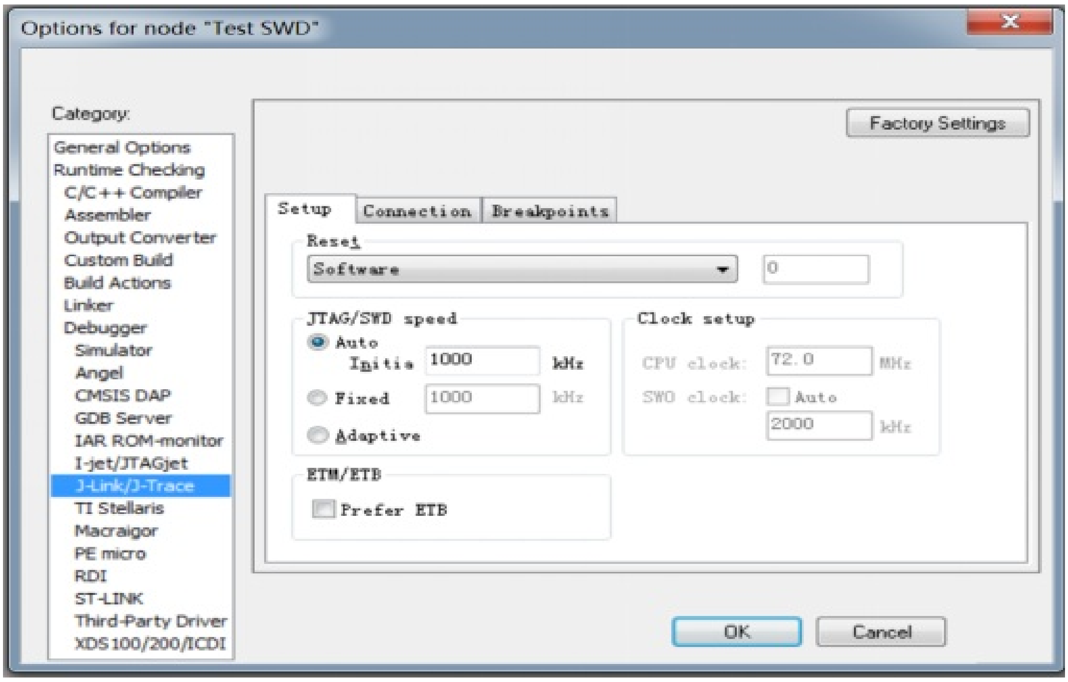
Setup macros

☐ Use macro file(s)

Device description file

☐ Override default

5.Just find a way to drive, set as shown below, and click OK。



Reset: Software 0

JTAG/SWD speed

● Auto

Initia 1000 kHz

○ Fixed 1000 kHz

○ Adaptive

Clock setup

CPU clock: 72.0 MHz

SWO clock: ☐ Auto

2000 kHz

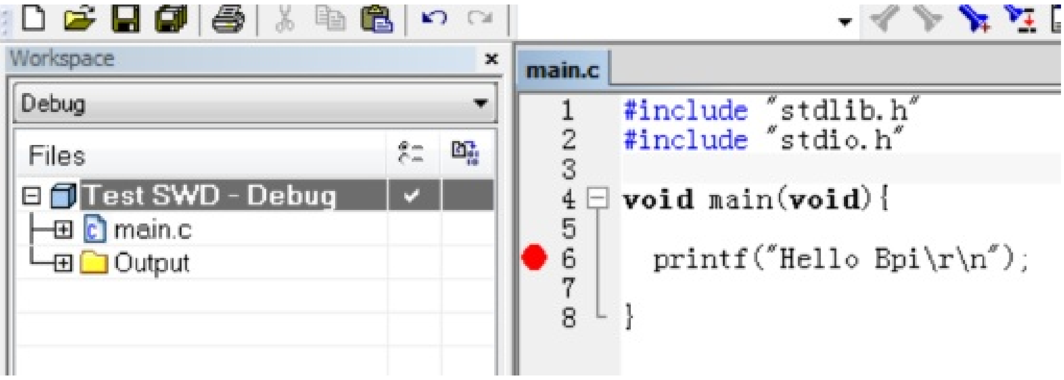
ETM/ETB

☐ Prefer ETB

You can now click on the columnCompile button，Such error can clickRun button to download the firmware.

SWD interface IAR EWARM single-step debugging

1.If you add a breakpoint in the code.

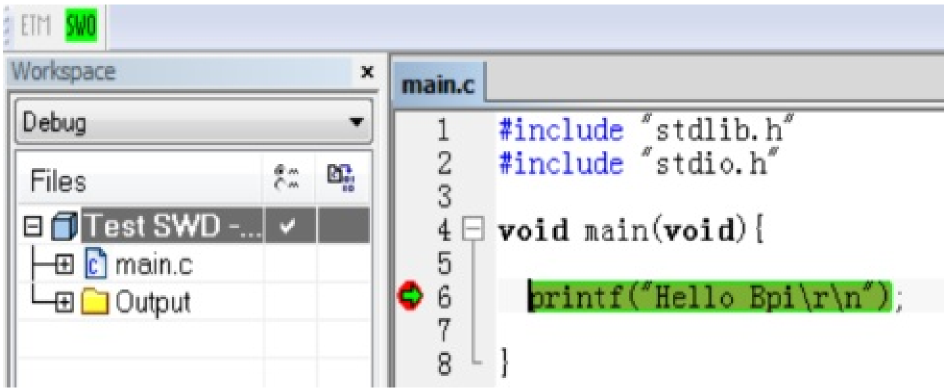


```
#include "stdlib.h"
#include "stdio.h"

void main(void){

  printf("Hello Bpi\r\n");
}
```

2.Then, after the above download will automatically enter debug mode.

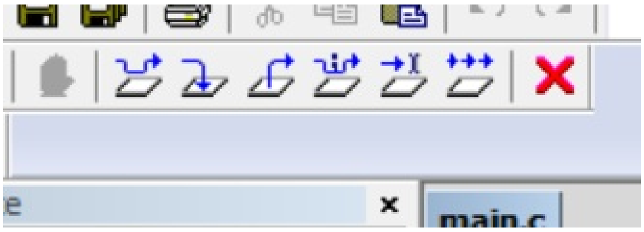


```
#include "stdlib.h"
#include "stdio.h"

void main(void){

  printf("Hello Bpi\r\n");
}
```

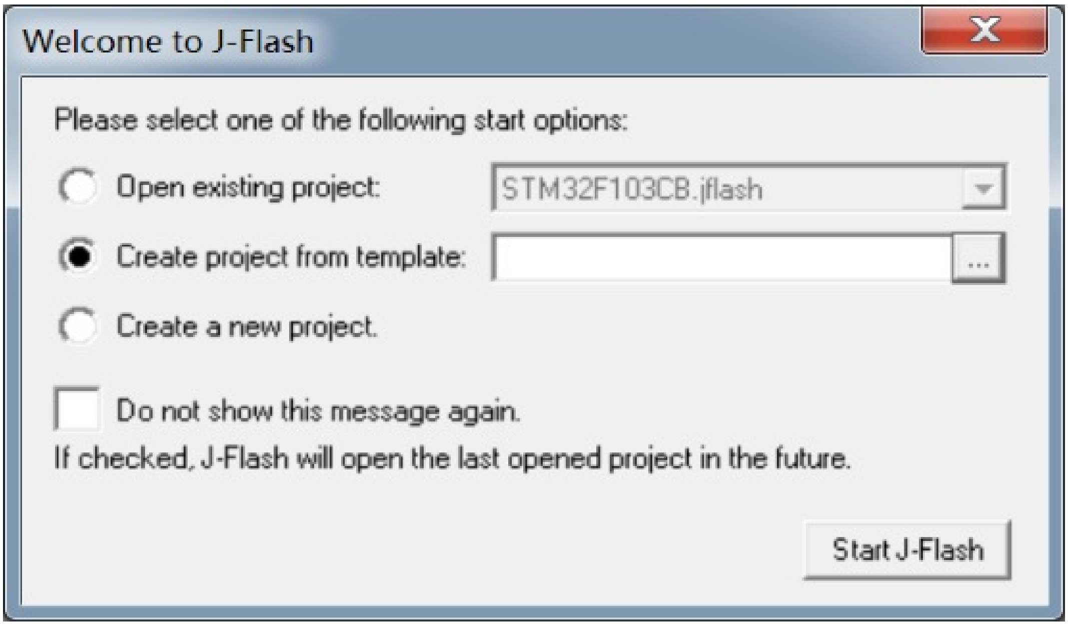
3.According to their own needs to control the program single-step operation into a function out of a function, and other operations running at full speed



Download the program using the SWD in J-Flash

To illustrate, here will be used as an example STM32F103C8T6。

1.Install J-Link ARM toolset open one of J-Flash.



Welcome to J-Flash

Please select one of the following start options:

○ Open existing project: STM32F103CB.jflash

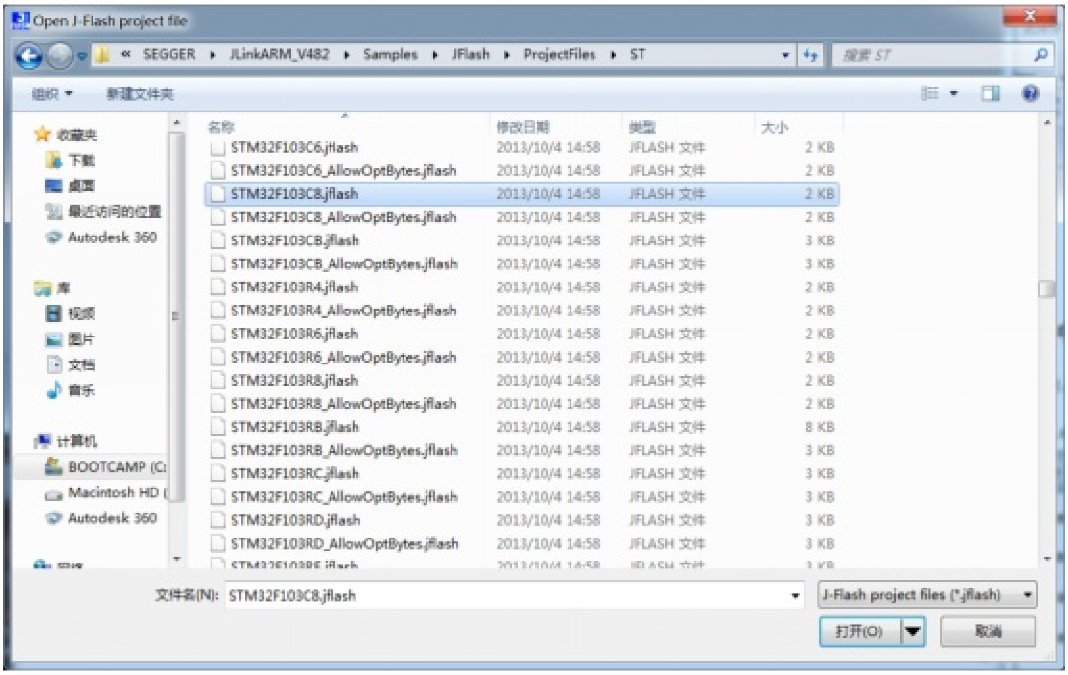
● Create project from template:

○ Create a new project.

□ Do not show this message again.

If checked, J-Flash will open the last opened project in the future.

Select Create project from template to open the file selection box, find STM32F103C8.jflash, as shown.



Open J-Flash project file

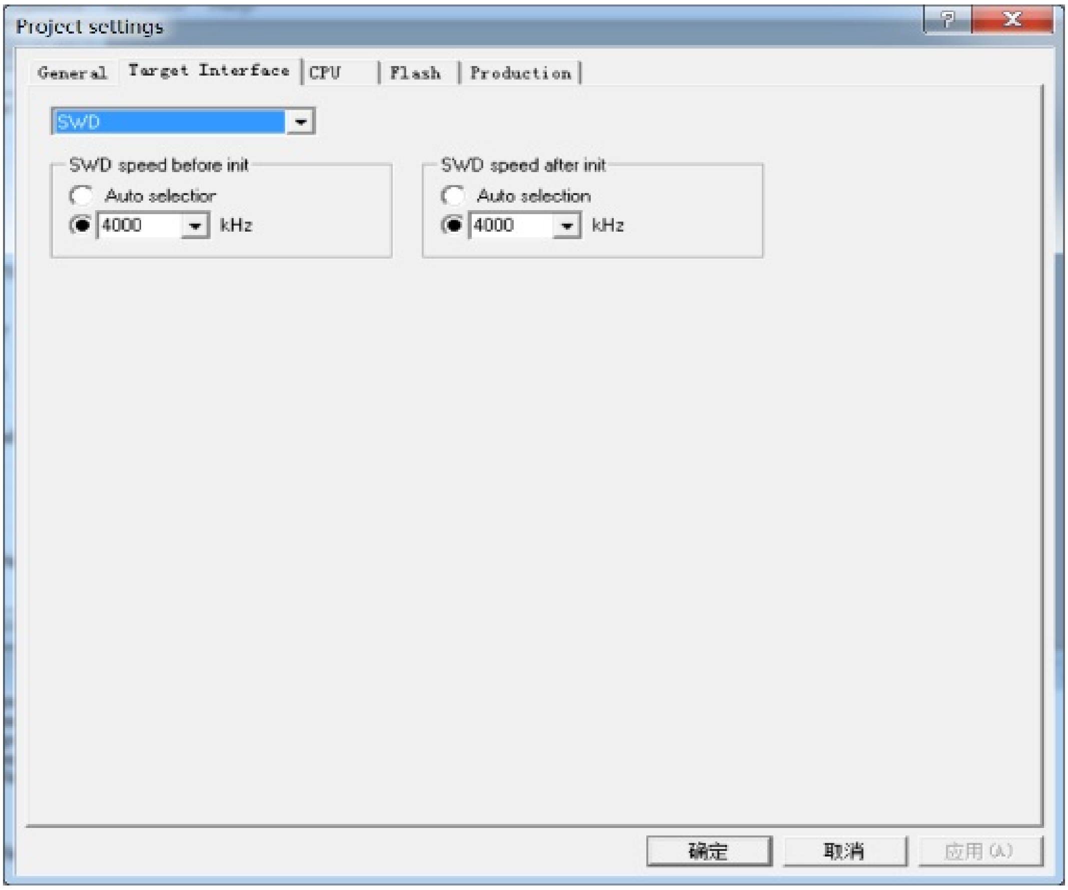
« SEGGER ▸ JLinkARM_V482 ▸ Samples ▸ JFlash ▸ ProjectFiles ▸ ST

名称	修改日期	类型	大小
STM32F103C6.jflash	2013/10/4 14:58	JFLASH 文件	2 KB
STM32F103C6_AllowOptBytes.jflash	2013/10/4 14:58	JFLASH 文件	2 KB
STM32F103C8.jflash	2013/10/4 14:58	JFLASH 文件	2 KB
STM32F103C8_AllowOptBytes.jflash	2013/10/4 14:58	JFLASH 文件	2 KB
STM32F103CB.jflash	2013/10/4 14:58	JFLASH 文件	3 KB
STM32F103CB_AllowOptBytes.jflash	2013/10/4 14:58	JFLASH 文件	3 KB
STM32F103R4.jflash	2013/10/4 14:58	JFLASH 文件	2 KB
STM32F103R4_AllowOptBytes.jflash	2013/10/4 14:58	JFLASH 文件	2 KB
STM32F103R6.jflash	2013/10/4 14:58	JFLASH 文件	2 KB
STM32F103R6_AllowOptBytes.jflash	2013/10/4 14:58	JFLASH 文件	2 KB
STM32F103R8.jflash	2013/10/4 14:58	JFLASH 文件	2 KB
STM32F103R8_AllowOptBytes.jflash	2013/10/4 14:58	JFLASH 文件	2 KB
STM32F103RB.jflash	2013/10/4 14:58	JFLASH 文件	8 KB
STM32F103RB_AllowOptBytes.jflash	2013/10/4 14:58	JFLASH 文件	3 KB
STM32F103RC.jflash	2013/10/4 14:58	JFLASH 文件	3 KB
STM32F103RC_AllowOptBytes.jflash	2013/10/4 14:58	JFLASH 文件	3 KB
STM32F103RD.jflash	2013/10/4 14:58	JFLASH 文件	3 KB
STM32F103RD_AllowOptBytes.jflash	2013/10/4 14:58	JFLASH 文件	3 KB

文件名(N): STM32F103C8.jflash

J-Flash project files (*.jflash)

2.Open the Options menu bar -> Project settings ..., switch to the Target Interface page, select download SWD, then click OK, as shown in Figure.



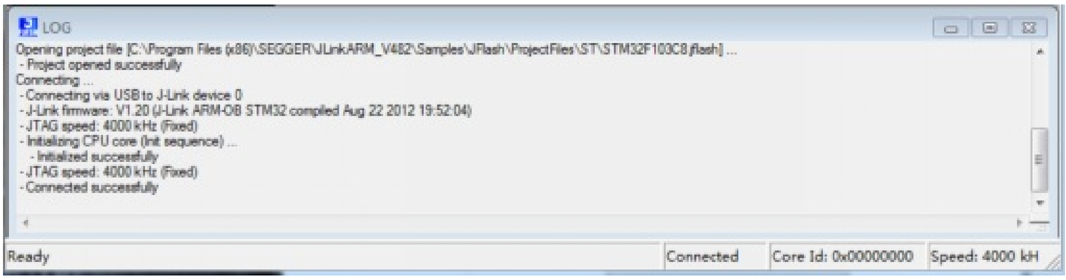
Project settings

SWD

SWD speed before init
○ Auto selection
● 4000 kHz

SWD speed after init
○ Auto selection
● 4000 kHz

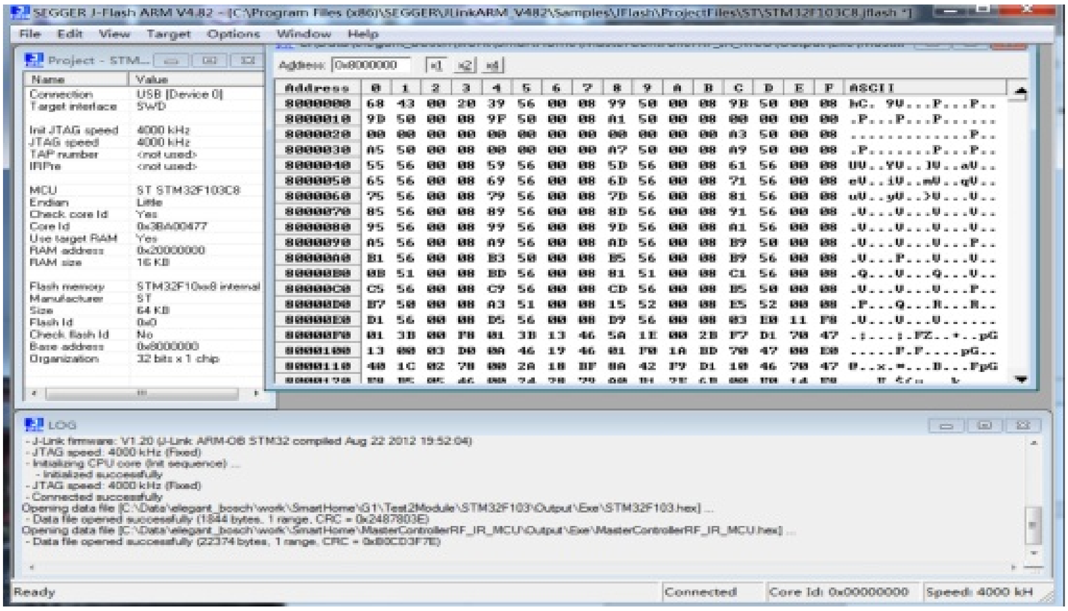
3.Then the Open Debugger for SWD interface CLK, DIO, GND based STM32F103C8T6 target plate PA13, PA14 and GND is connected, and then click on the menu bar Target -> Connect, under normal circumstances, the box will print LOG output similar to the following chart information If not, then please check the connection Open Debbuger with your computer and check the Open Debugger device drivers are properly.



```
Opening project file [C:\Program Files (x86)\SEGGER\JLinkARM_V482\Samples\JFlash\ProjectFiles\ST\STM32F103C8.jflash] ...
 - Project opened successfully
Connecting ...
 - Connecting via USB to J-Link device 0
 - J-Link firmware: V1.20 (J-Link ARM-OB STM32 compiled Aug 22 2012 19:52:04)
 - JTAG speed: 4000 kHz (Fixed)
 - Initializing CPU core (Init sequence) ...
   - Initialized successfully
 - JTAG speed: 4000 kHz (Fixed)
 - Connected successfully
```

Ready | Connected | Core Id: 0x00000000 | Speed: 4000 kH

4.When you see the above information, you can open the firmware file to be programmed menu bar File - after> Open data file ..., find the file you want to program, there will be a firmware binary data, as shown below.



SEGGER J-Flash ARM V4.82 - [C:\Program Files (x86)\SEGGER\JLinkARM_V482\Samples\JFlash\ProjectFiles\ST\STM32F103C8.jflash *]

Name	Value
Connection	USB [Device 0]
Target interface	SWD
Init JTAG speed	4000 kHz
JTAG speed	4000 kHz
TAP number	<not used>
IRPre	<not used>
MCU	ST STM32F103C8
Endian	Little
Check core Id	Yes
Core Id	0x3BA00477
Use target RAM	Yes
RAM address	0x20000000
RAM size	16 KB
Flash memory	STM32F10xx8 internal
Manufacturer	ST
Size	64 KB
Flash Id	0x0
Check flash Id	No
Base address	0x8000000
Organization	32 bits x 1 chip

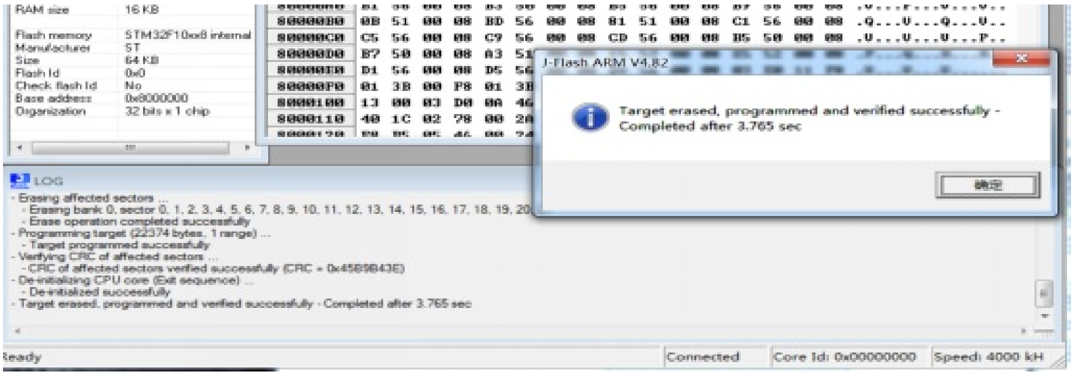
Address: 0x8000000

```
Address   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  ASCII
8000000  68 43 00 20 39 56 00 08 99 50 00 08 9B 50 00 08  hC. 9V...P...P..
8000010  9D 50 00 08 9F 50 00 08 A1 50 00 08 00 00 00 00  .P...P...P......
8000020  00 00 00 00 00 00 00 00 00 00 00 00 A3 50 00 08  .............P..
8000030  A5 50 00 08 00 00 00 00 A7 50 00 08 A9 50 00 08  .P.......P...P..
8000040  55 56 00 08 59 56 00 08 5D 56 00 08 61 56 00 08  UV..YV..]V..aV..
8000050  65 56 00 08 69 56 00 08 6D 56 00 08 71 56 00 08  eV..iV..mV..qV..
8000060  75 56 00 08 79 56 00 08 7D 56 00 08 81 56 00 08  uV..yV..}V...V..
8000070  85 56 00 08 89 56 00 08 8D 56 00 08 91 56 00 08  .V...V...V...V..
8000080  95 56 00 08 99 56 00 08 9D 56 00 08 A1 56 00 08  .V...V...V...V..
8000090  A5 56 00 08 A9 56 00 08 AD 56 00 08 B9 50 00 08  .V...V...V...P..
80000A0  B1 56 00 08 B3 50 00 08 B5 56 00 08 B9 56 00 08  .V...P...V...V..
80000B0  0B 51 00 08 BD 56 00 08 81 51 00 08 C1 56 00 08  .Q...V...Q...V..
80000C0  C5 56 00 08 C9 56 00 08 CD 56 00 08 D5 50 00 08  .V...V...V...P..
80000D0  B7 50 00 08 A3 51 00 08 15 52 00 08 E5 52 00 08  .P...Q...R...R..
80000E0  D1 56 00 08 D5 56 00 08 D9 56 00 08 03 E0 11 F8  .V...V...V......
80000F0  01 3B 00 F8 01 3B 13 46 5A 1E 00 2B F7 D1 70 47  .;...;.FZ..+..pG
8000100  13 00 03 D0 0A 46 19 46 01 F0 1A BD 70 47 00 E0  ......F.F....pG..
8000110  40 1C 02 78 00 2A 18 BF 8A 42 F9 D1 10 46 70 47  @..x.*...B...FpG
```

```
- J-Link firmware: V1.20 (J-Link ARM-OB STM32 compiled Aug 22 2012 19:52:04)
- JTAG speed: 4000 kHz (Fixed)
- Initializing CPU core (Init sequence) ...
  - Initialized successfully
- JTAG speed: 4000 kHz (Fixed)
- Connected successfully
Opening data file [C:\Data\elegant_bosch\work\SmartHome\G1\Test2Module\STM32F103\Output\Exe\STM32F103.hex] ...
 - Data file opened successfully (1844 bytes, 1 range, CRC = 0x2487803E)
Opening data file [C:\Data\elegant_bosch\work\SmartHome\MasterControllerRF_IR_MCU\Output\Exe\MasterControllerRF_IR_MCU.hex] ...
 - Data file opened successfully (22374 bytes, 1 range, CRC = 0xB0CD3F7E)
```

Ready | Connected | Core Id: 0x00000000 | Speed: 4000 kH

5.This time click on the menu bar Target -> Auto will start automatically programmed,without question in the following figure LOG successful programming message will appear,if not, please check the wiring of the SWD Open Debugger interface with the target board is correct, power supply is normal.



RAM size	16 KB
Flash memory	STM32F10xx8 internal
Manufacturer	ST
Size	64 KB
Flash Id	0x0
Check flash Id	No
Base address	0x8000000
Organization	32 bits x 1 chip

```
80000B0  0B 51 00 08 BD 56 00 08 81 51 00 08 C1 56 00 08  .Q...V...Q...V..
80000C0  C5 56 00 08 C9 56 00 08 CD 56 00 08 D5 50 00 08  .V...V...V...P..
80000D0  B7 50 00 08 A3 51
80000E0  D1 56 00 08 D5 56
80000F0  01 3B 00 F8 01 3B
8000100  13 00 03 D0 0A 46
8000110  40 1C 02 78 00 2A
```

J-Flash ARM V4.82

Target erased, programmed and verified successfully - Completed after 3.765 sec

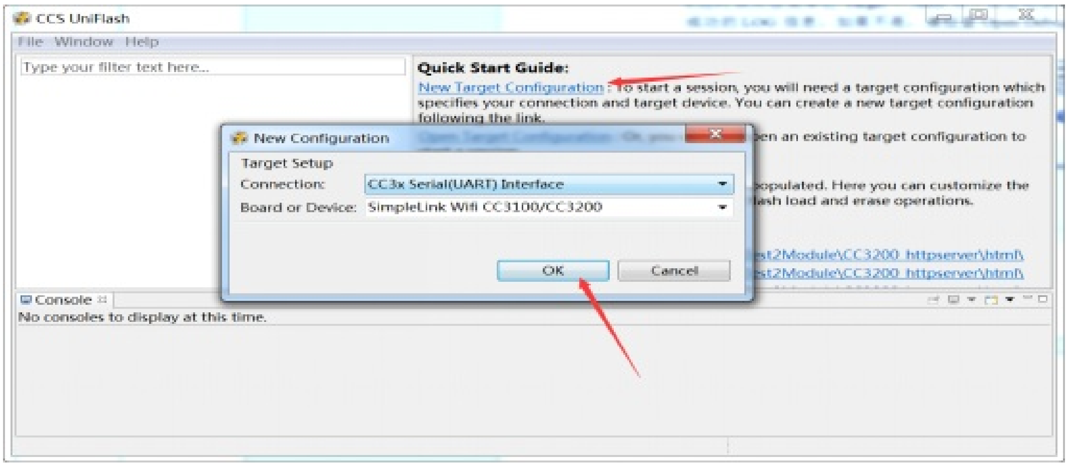
```
- Erasing affected sectors ...
  - Erasing bank 0, sector 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
  - Erase operation completed successfully
- Programming target (22374 bytes, 1 range) ...
  - Target programmed successfully
- Verifying CRC of affected sectors ...
  - CRC of affected sectors verified successfully (CRC = 0x45B9B43E)
- De-initializing CPU core (Exit sequence) ...
  - De-initialized successfully
- Target erased, programmed and verified successfully - Completed after 3.765 sec
```

Ready | Connected | Core Id: 0x00000000 | Speed: 4000 kH

Use Uni flash via JTAG and UART interface to TI CC3200 in the curing process

1.First, the Open Debugger The JTAG interface and UART interface with connection-based TI CC3200 chip development board JTAG interface, UART interface (GPIO1, GPIO2) and GND.

2.Open uniflash, click on the box to the right of the New Target Configuration, a dialog box appears, by default, you can select, point OK, as shown below.



CCS UniFlash

File Window Help

Type your filter text here...

Quick Start Guide:

New Target Configuration: To start a session, you will need a target configuration which specifies your connection and target device. You can create a new target configuration following the link.

New Configuration

Target Setup

Connection: CC3x Serial(UART) Interface

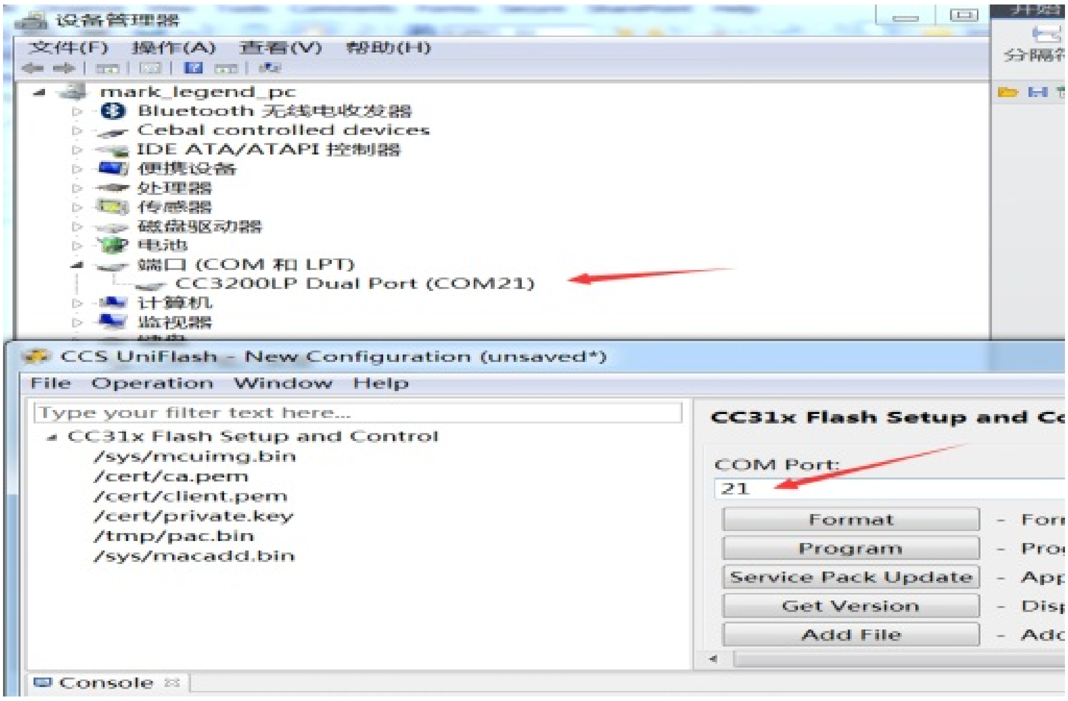
Board or Device: SimpleLink Wifi CC3100/CC3200

OK Cancel

Console

No consoles to display at this time.

3.The Open Debugger plug in the computer, see the serial number in the Windows Device Manager, and fill it to the input box COM Port.



设备管理器

文件(F) 操作(A) 查看(V) 帮助(H)

mark_legend_pc
- Bluetooth 无线电收发器
- Cebal controlled devices
- IDE ATA/ATAPI 控制器
- 便携设备
- 处理器
- 传感器
- 磁盘驱动器
- 电池
- 端口 (COM 和 LPT)
  - CC3200LP Dual Port (COM21)
- 计算机
- 监视器

CCS UniFlash - New Configuration (unsaved*)

File Operation Window Help

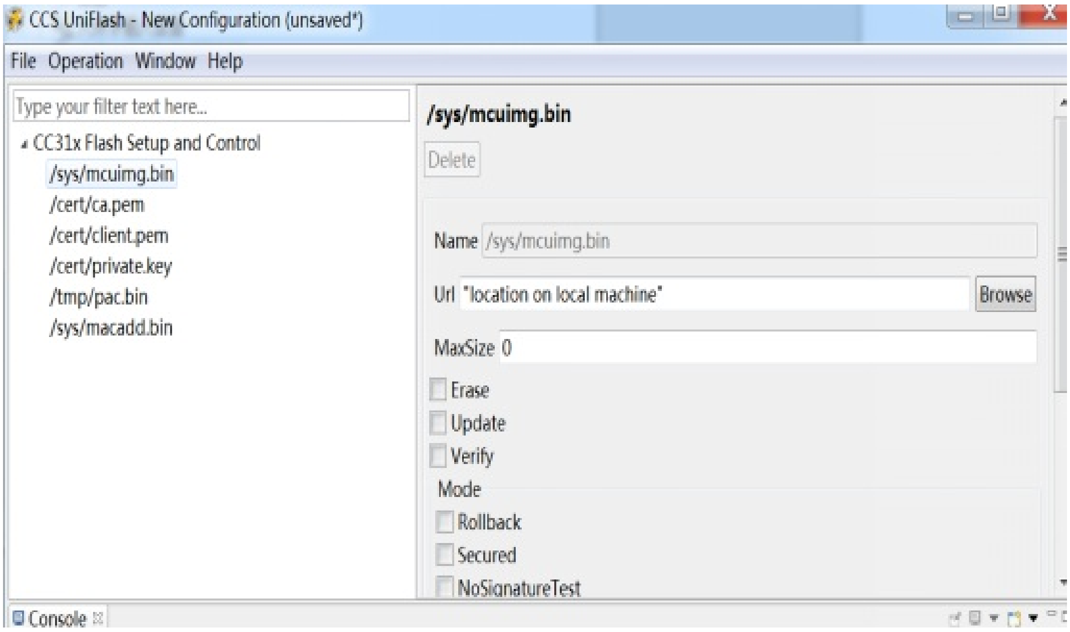
Type your filter text here...

CC31x Flash Setup and Control
- /sys/mcuimg.bin
- /cert/ca.pem
- /cert/client.pem
- /cert/private.key
- /tmp/pac.bin
- /sys/macadd.bin

CC31x Flash Setup and Co

COM Port: 21

4.Select the list on the left /sys/mcuimg.bin one, will appear as shown.


CCS UniFlash - New Configuration (unsaved*)
File Operation Window Help
Type your filter text here...
CC31x Flash Setup and Control
/sys/mcuimg.bin
/cert/ca.pem
/cert/client.pem
/cert/private.key
/tmp/pac.bin
/sys/macadd.bin
/sys/mcuimg.bin
Delete
Browse
Console

Name /sys/mcuimg.bin

Url "location on local machine"

MaxSize 0

□ Erase

□ Update

□ Verify

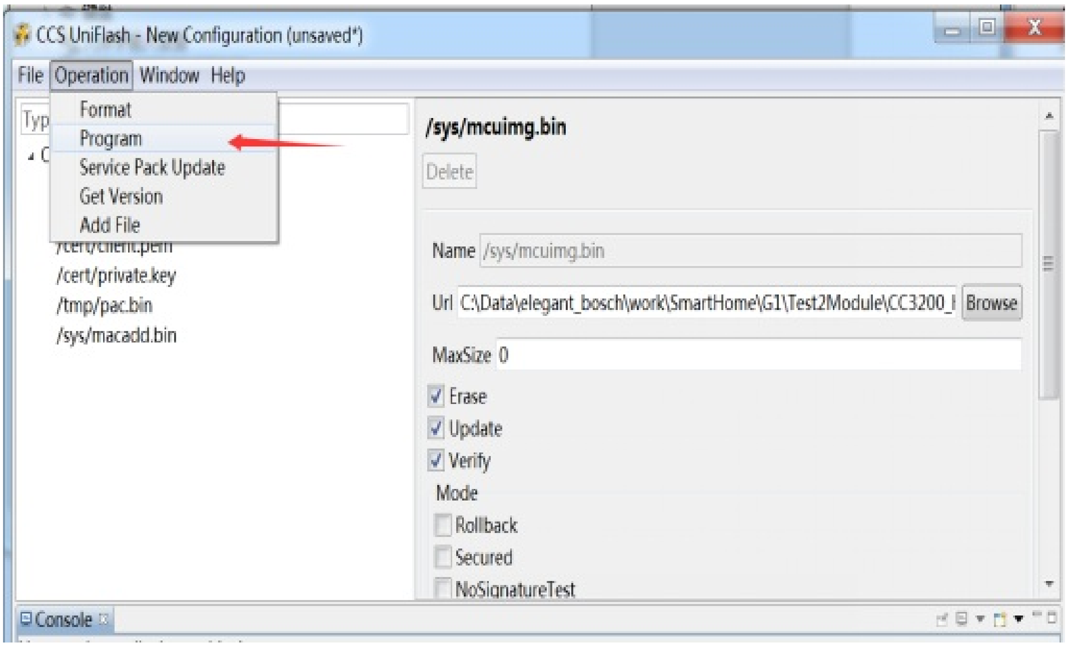
Mode

□ Rollback

□ Secured

□ NoSignatureTest

5.Click the Browse button to select the firmware file you want to program, and the following checkbox Erase, Update, Verify the hook. Then click on the menu bar Operation-> Program.



Name /sys/mcuimg.bin

Url C:\Data\elegant_bosch\work\SmartHome\G1\Test2Module\CC3200_I

MaxSize 0

☑ Erase

☑ Update

☑ Verify

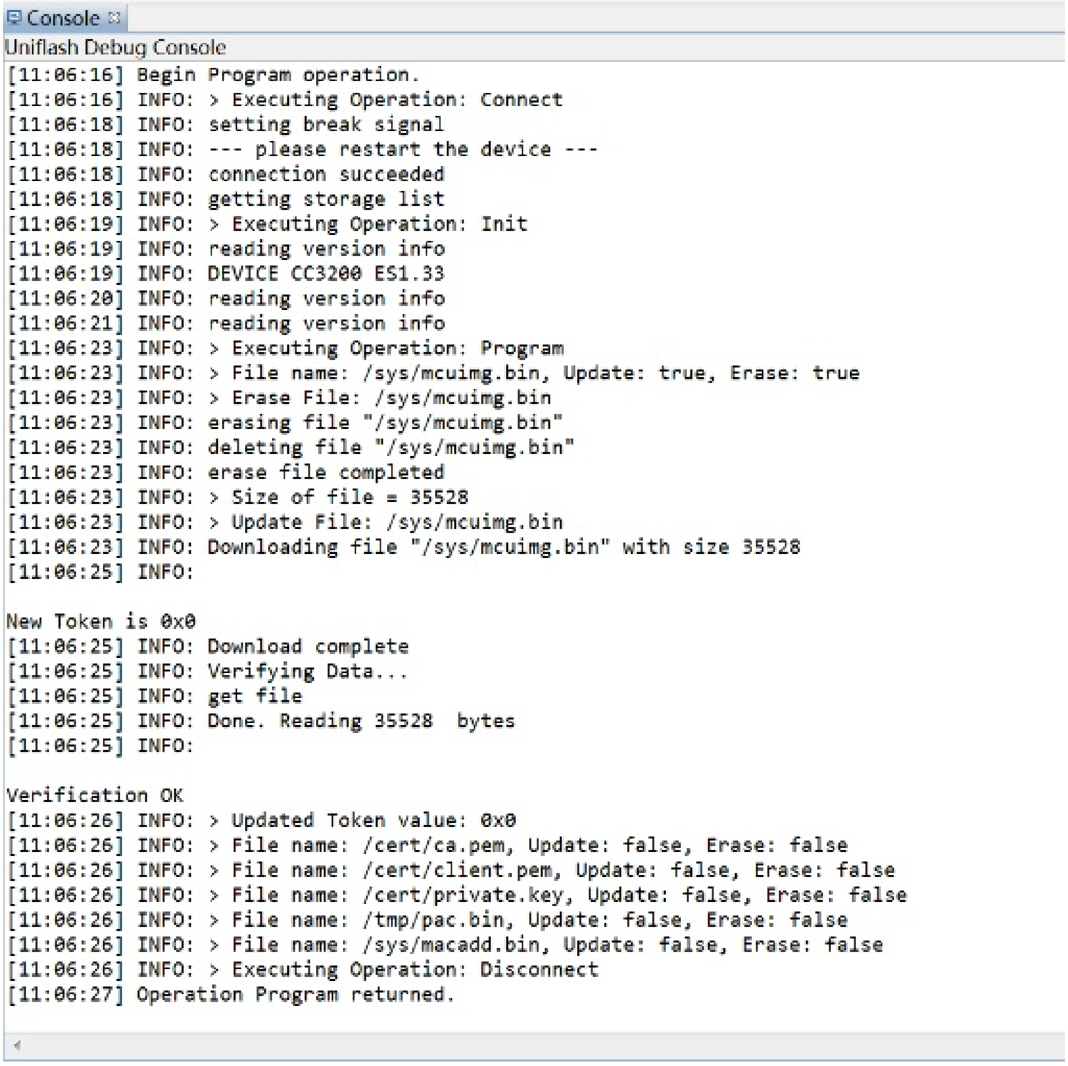
Mode

□ Rollback

□ Secured

□ NoSignatureTest

6.LOG following information will click on Program programming software automatically after waiting for a series of operations carried out under normal circumstances arise.

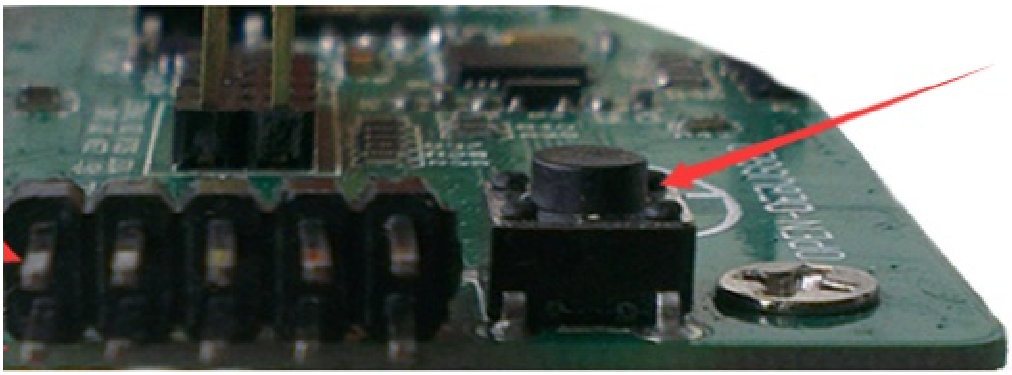


```
[11:06:16] Begin Program operation.
[11:06:16] INFO: > Executing Operation: Connect
[11:06:18] INFO: setting break signal
[11:06:18] INFO: --- please restart the device ---
[11:06:18] INFO: connection succeeded
[11:06:18] INFO: getting storage list
[11:06:19] INFO: > Executing Operation: Init
[11:06:19] INFO: reading version info
[11:06:19] INFO: DEVICE CC3200 ES1.33
[11:06:20] INFO: reading version info
[11:06:21] INFO: reading version info
[11:06:23] INFO: > Executing Operation: Program
[11:06:23] INFO: > File name: /sys/mcuimg.bin, Update: true, Erase: true
[11:06:23] INFO: > Erase File: /sys/mcuimg.bin
[11:06:23] INFO: erasing file "/sys/mcuimg.bin"
[11:06:23] INFO: deleting file "/sys/mcuimg.bin"
[11:06:23] INFO: erase file completed
[11:06:23] INFO: > Size of file = 35528
[11:06:23] INFO: > Update File: /sys/mcuimg.bin
[11:06:23] INFO: Downloading file "/sys/mcuimg.bin" with size 35528
[11:06:25] INFO:

New Token is 0x0
[11:06:25] INFO: Download complete
[11:06:25] INFO: Verifying Data...
[11:06:25] INFO: get file
[11:06:25] INFO: Done. Reading 35528  bytes
[11:06:25] INFO:

Verification OK
[11:06:26] INFO: > Updated Token value: 0x0
[11:06:26] INFO: > File name: /cert/ca.pem, Update: false, Erase: false
[11:06:26] INFO: > File name: /cert/client.pem, Update: false, Erase: false
[11:06:26] INFO: > File name: /cert/private.key, Update: false, Erase: false
[11:06:26] INFO: > File name: /tmp/pac.bin, Update: false, Erase: false
[11:06:26] INFO: > File name: /sys/macadd.bin, Update: false, Erase: false
[11:06:26] INFO: > Executing Operation: Disconnect
[11:06:27] Operation Program returned.
```

7.In the automatic programming process, if LOG stay in --- please restart the device --- in, then press the reset button next to the JTAG interface, as shown below.

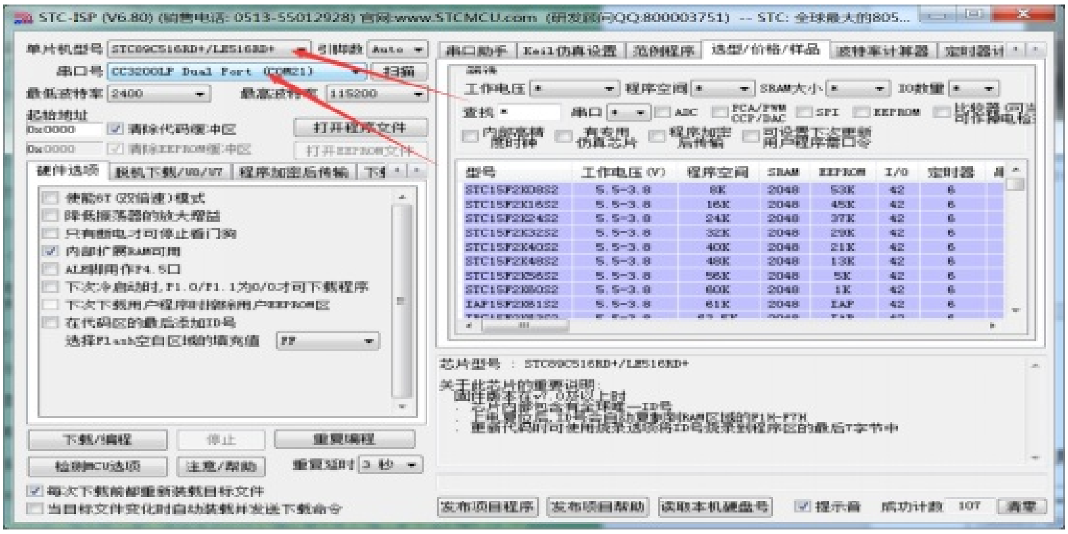


8.If not, the above scenario or press the reset button, but the programming software for a long time to stay in an unresponsive state, JTAG interface target board and Uart interface and Open Debbuger check the wiring is correct, the power supply is normal.

Open Debugger Uart use of STC microcontroller programming interface

To illustrate, here using the model for STC 89C516RD + microcontroller

1.Uart in the Open Debugger interface and connection with the STC microcontroller as STC microcontroller requires a cold start, do not use Open Debugger VCC supply to STC, and then open the STC-ISP software, select the model and serial microcontroller, as.



单片机型号 STC89C516RD+/LE516RD+　引脚数 Auto
串口号 CC3200LF Dual Port (COM21)
最低波特率 2400　最高波特率 115200
起始地址
0x0000 ☑ 清除代码缓冲区
0x0000 ☑ 清除EEPROM缓冲区

□ 使能6T (双倍速)模式
□ 降低振荡器的放大增益
□ 只有断电才可停止看门狗
☑ 内部扩展RAM可用
□ ALE脚用作P4.5口
□ 下次冷启动时, P1.0/P1.1为0/0才可下载程序
□ 下次下载用户程序时擦除用户EEPROM区
□ 在代码区的最后添加ID号
选择Flash空白区域的填充值 FF

重复延时 3 秒
☑ 每次下载前都重新装载目标文件
□ 当目标文件变化时自动装载并发送下载命令

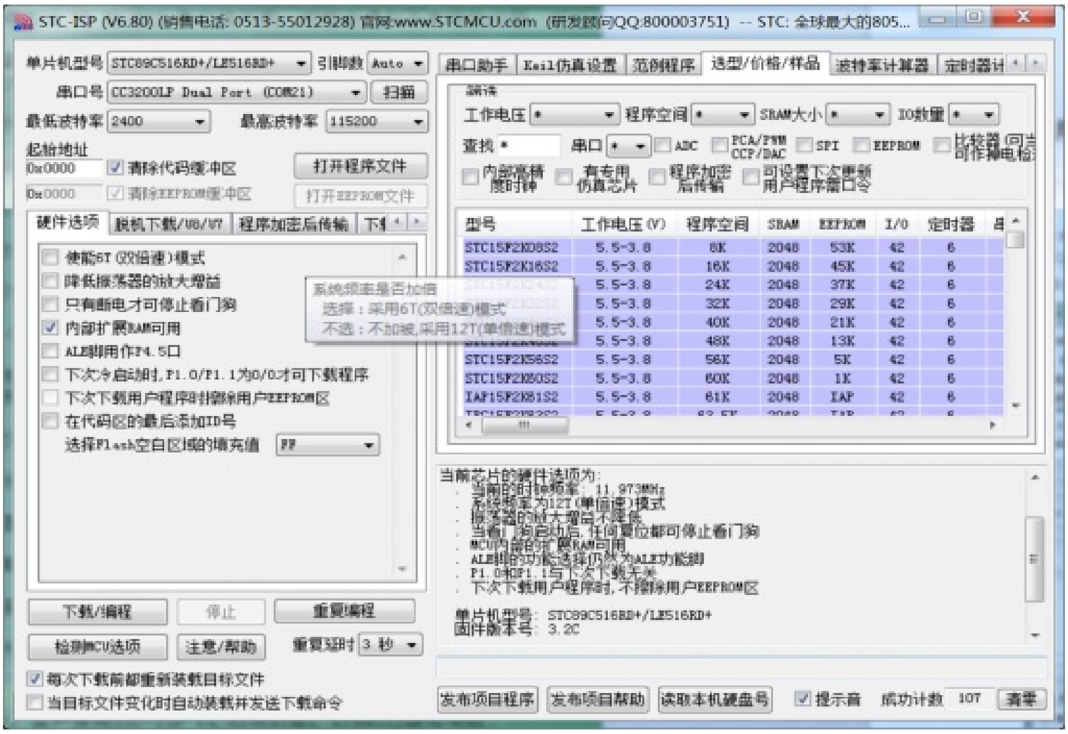
筛选
工作电压 *　程序空间 *　SRAM大小 *　IO数量 *
查找 *　串口 *　□ ADC　□ PCA/PWM CCP/DAC　□ SPI　□ EEPROM　□ 比较器(可当可作掉电检测)
□ 内部高精度时钟　□ 有专用仿真芯片　□ 程序加密后传输　□ 可设置下次更新用户程序启口令

型号	工作电压(V)	程序空间	SRAM	EEPROM	I/O	定时器
STC15F2K08S2	5.5-3.8	8K	2048	53K	42	6
STC15F2K16S2	5.5-3.8	16K	2048	45K	42	6
STC15F2K24S2	5.5-3.8	24K	2048	37K	42	6
STC15F2K32S2	5.5-3.8	32K	2048	29K	42	6
STC15F2K40S2	5.5-3.8	40K	2048	21K	42	6
STC15F2K48S2	5.5-3.8	48K	2048	13K	42	6
STC15F2K56S2	5.5-3.8	56K	2048	5K	42	6
STC15F2K60S2	5.5-3.8	60K	2048	1K	42	6
IAP15F2K61S2	5.5-3.8	61K	2048	IAP	42	6

芯片型号 : STC89C516RD+/LE516RD+
关于此芯片的重要说明:
固件版本在v7.0及以上时
· 芯片内部包含有全球唯一ID号
· 上电复位后, ID号会自动复制到RAM区域的F1H-F7H
· 重新代码时可使用该选项将ID号添加到程序区的最后7字节中

☑ 提示音　成功计数 107

2.Disconnect the power microcontroller, and then point detection MCU option, and then give the power microcontroller, the microcontroller will read out information.



单片机型号 STC89C516RD+/LE516RD+　引脚数 Auto
串口号 CC3200LF Dual Port (COM21)
最低波特率 2400　最高波特率 115200
起始地址
0x0000 ☑ 清除代码缓冲区
0x0000 ☑ 清除EEPROM缓冲区

□ 使能6T (双倍速)模式
□ 降低振荡器的放大增益
□ 只有断电才可停止看门狗
☑ 内部扩展RAM可用
□ ALE脚用作P4.5口
□ 下次冷启动时, P1.0/P1.1为0/0才可下载程序
□ 下次下载用户程序时擦除用户EEPROM区
□ 在代码区的最后添加ID号
选择Flash空白区域的填充值 FF

系统频率是否加倍
选择：采用6T(双倍速)模式
不选：不加倍,采用12T(单倍速)模式

重复延时 3 秒
☑ 每次下载前都重新装载目标文件
□ 当目标文件变化时自动装载并发送下载命令

筛选
工作电压 *　程序空间 *　SRAM大小 *　IO数量 *
查找 *　串口 *　□ ADC　□ PCA/PWM CCP/DAC　□ SPI　□ EEPROM　□ 比较器(可当可作掉电检测)
□ 内部高精度时钟　□ 有专用仿真芯片　□ 程序加密后传输　□ 可设置下次更新用户程序启口令

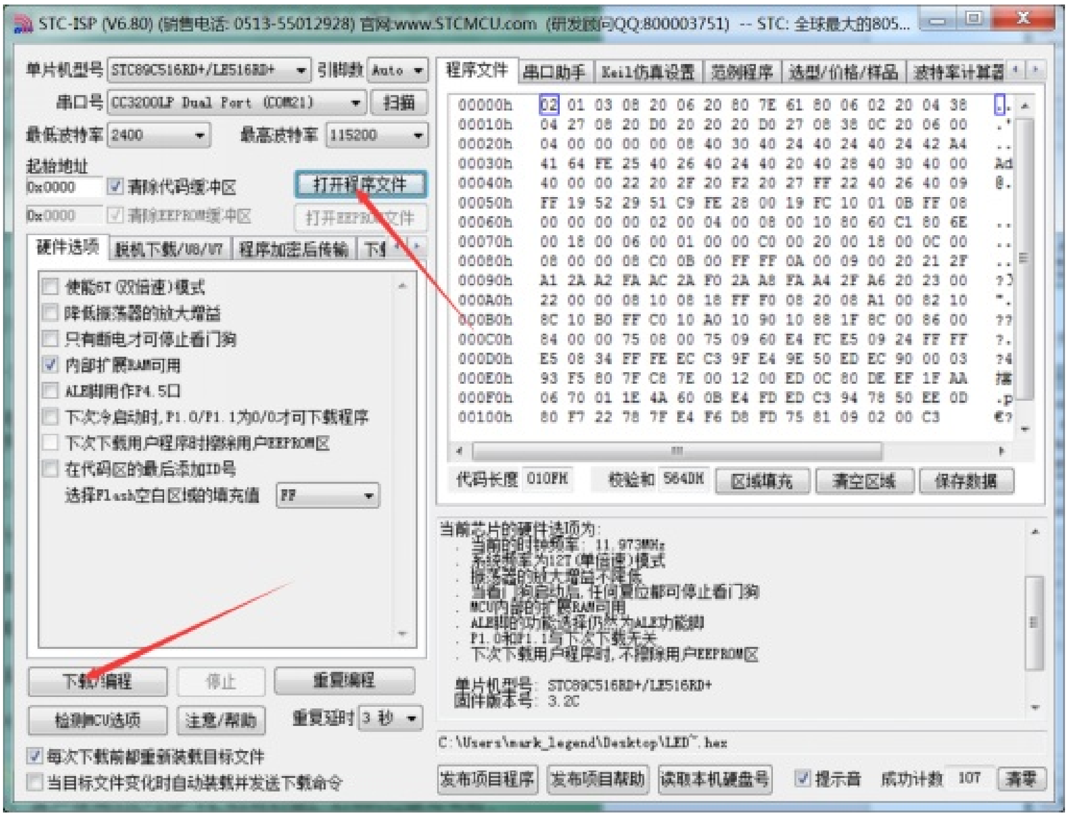
型号	工作电压(V)	程序空间	SRAM	EEPROM	I/O	定时器
STC15F2K08S2	5.5-3.8	8K	2048	53K	42	6
STC15F2K16S2	5.5-3.8	16K	2048	45K	42	6
STC15F2K24S2	5.5-3.8	24K	2048	37K	42	6
STC15F2K32S2	5.5-3.8	32K	2048	29K	42	6
STC15F2K40S2	5.5-3.8	40K	2048	21K	42	6
STC15F2K48S2	5.5-3.8	48K	2048	13K	42	6
STC15F2K56S2	5.5-3.8	56K	2048	5K	42	6
STC15F2K60S2	5.5-3.8	60K	2048	1K	42	6
IAP15F2K61S2	5.5-3.8	61K	2048	IAP	42	6

当前芯片的硬件选项为:
· 当前的时钟频率: 11.973MHz
· 系统频率为12T(单倍速)模式
· 振荡器的放大增益不降低
· 当看门狗启动后, 任何复位都可停止看门狗
· MCU内部的扩展RAM可用
· ALE脚的功能选择仍然为ALE功能脚
· P1.0和P1.1与下次下载无关
· 下次下载用户程序时, 不擦除用户EEPROM区

单片机型号: STC89C516RD+/LE516RD+
固件版本号: 3.2C

☑ 提示音　成功计数 107

3.Then point to open the program file, select STC microcontroller firmware to be programmed, then the microcontroller power, and then point to download / Programming.



STC-ISP (V6.80) (销售电话: 0513-55012928) 官网:www.STCMCU.com (研发顾问QQ:800003751) -- STC: 全球最大的805...

单片机型号 STC89C516RD+/LE516RD+ 引脚数 Auto

串口号 CC3200LP Dual Port (COM21) 扫描

最低波特率 2400 最高波特率 115200

起始地址

0x0000 ☑ 清除代码缓冲区 打开程序文件

0x0000 ☑ 清除EEPROM缓冲区 打开EEPROM文件

硬件选项 | 脱机下载/U8/U7 | 程序加密后传输 | 下

☐ 使能6T(双倍速)模式

☐ 降低振荡器的放大增益

☐ 只有断电才可停止看门狗

☑ 内部扩展RAM可用

☐ ALE脚用作P4.5口

☐ 下次冷启动时,P1.0/P1.1为0/0才可下载程序

☐ 下次下载用户程序时擦除用户EEPROM区

☐ 在代码区的最后添加ID号

选择Flash空白区域的填充值 FF

```
00000h   02 01 03 08 20 06 20 80 7E 61 80 06 02 20 04 38
00010h   04 27 08 20 D0 20 20 20 D0 27 08 38 0C 20 06 00
00020h   04 00 00 00 00 08 40 30 40 24 40 24 40 24 42 A4
00030h   41 64 FE 25 40 26 40 24 40 20 40 28 40 30 40 00
00040h   40 00 00 22 20 2F 20 F2 20 27 FF 22 40 26 40 09
00050h   FF 19 52 29 51 C9 FE 28 00 19 FC 10 01 0B FF 08
00060h   00 00 00 00 02 00 04 00 08 00 10 80 60 C1 80 6E
00070h   00 18 00 06 00 01 00 00 C0 00 20 00 18 00 0C 00
00080h   08 00 00 08 C0 0B 00 FF FF 0A 00 09 00 20 21 2F
00090h   A1 2A A2 FA AC 2A F0 2A A8 FA A4 2F A6 20 23 00
000A0h   22 00 00 08 10 08 18 FF F0 08 20 08 A1 00 82 10
000B0h   8C 10 B0 FF C0 10 A0 10 90 10 88 1F 8C 00 86 00
000C0h   84 00 00 75 08 00 75 09 60 E4 FC E5 09 24 FF FF
000D0h   E5 08 34 FF FE EC C3 9F E4 9E 50 ED EC 90 00 03
000E0h   93 F5 80 7F C8 7E 00 12 00 ED 0C 80 DE EF 1F AA
000F0h   06 70 01 1E 4A 60 0B E4 FD ED C3 94 78 50 EE 0D
00100h   80 F7 22 78 7F E4 F6 D8 FD 75 81 09 02 00 C3
```

代码长度 010FH 校验和 564DH 区域填充 清空区域 保存数据

```
当前芯片的硬件选项为:
  . 当前的时钟频率: 11.973MHz
  . 系统频率为12T(单倍速)模式
  . 振荡器的放大增益不降低
  . 当看门狗启动后,任何复位都可停止看门狗
  . MCU内部的扩展RAM可用
  . ALE脚的功能选择仍然为ALE功能脚
  . P1.0和P1.1与下次下载无关
  . 下次下载用户程序时,不擦除用户EEPROM区

  单片机型号: STC89C516RD+/LE516RD+
  固件版本号: 3.2C
```

下载/编程 停止 重复编程

检测MCU选项 注意/帮助 重复延时 3 秒

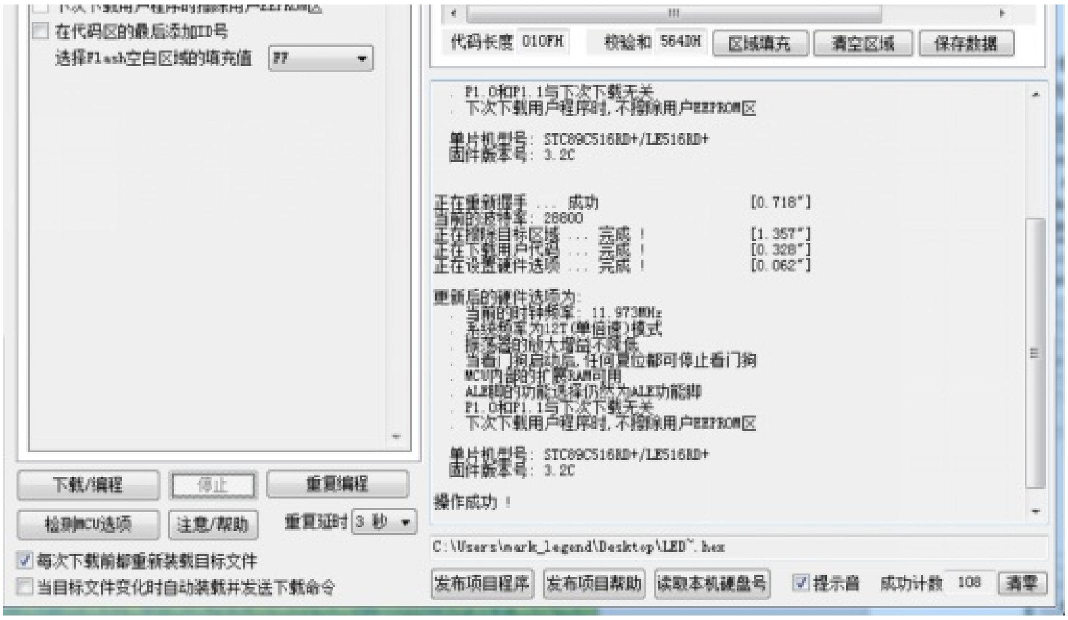
☑ 每次下载前都重新装载目标文件

☐ 当目标文件变化时自动装载并发送下载命令

C:\Users\mark_legend\Desktop\LED~.hex

发布项目程序 发布项目帮助 读取本机硬盘号 ☑ 提示音 成功计数 107 清零

4.Download/programmed, to give the power microcontroller, programmed to be completed.



☐ 在代码区的最后添加ID号

选择Flash空白区域的填充值 FF

代码长度 010FH 校验和 564DH 区域填充 清空区域 保存数据

```
  . P1.0和P1.1与下次下载无关
  . 下次下载用户程序时,不擦除用户EEPROM区

  单片机型号: STC89C516RD+/LE516RD+
  固件版本号: 3.2C


正在重新握手 ... 成功                    [0.718"]
当前的波特率: 28800
正在擦除目标区域 ... 完成 !              [1.357"]
正在下载用户代码 ... 完成 !              [0.328"]
正在设置硬件选项 ... 完成 !              [0.062"]

更新后的硬件选项为:
  . 当前的时钟频率: 11.973MHz
  . 系统频率为12T(单倍速)模式
  . 振荡器的放大增益不降低
  . 当看门狗启动后,任何复位都可停止看门狗
  . MCU内部的扩展RAM可用
  . ALE脚的功能选择仍然为ALE功能脚
  . P1.0和P1.1与下次下载无关
  . 下次下载用户程序时,不擦除用户EEPROM区

  单片机型号: STC89C516RD+/LE516RD+
  固件版本号: 3.2C

操作成功 !
```

下载/编程 停止 重复编程

检测MCU选项 注意/帮助 重复延时 3 秒

☑ 每次下载前都重新装载目标文件

☐ 当目标文件变化时自动装载并发送下载命令

C:\Users\mark_legend\Desktop\LED~.hex

发布项目程序 发布项目帮助 读取本机硬盘号 ☑ 提示音 成功计数 108 清零

5.If the power for a long time did not respond, check the cable is connected right, or attempt to re-power. Note: Open Debugger UART interface through RXD / TXD pin on the microcontroller power supply, causing power outages can not completely downloaded. TxD pins in microcontroller series with a diode (then positive), so that the microcontroller can be powered down completely cold start.

Reference

[1] CC-Debugger product web site

http://www.ti.com/tool/cc-debugger

[2] CC-Debugger Quick Start Guide

http://www.ti.com/lit/swru196

[3] CC-Debugger Lay out and Schematics

http://www.ti.com/lit/zip/swrr105

[4] Cebal – CCxxxx Development Tools USB Driv er for Windows x86 and x64

http://www.ti.com/lit/zip/swrc212

[5] DN304 – CCxxxx Development Tools USB Driv er I nstallation Guide

http://www.ti.com/lit/swra366

[6] Texas I nstruments Support

http://support.ti.com

[7] Texas I nstruments Low Power RF Online Community

http://www.ti.com/lprf-forum

[8] SmartRF Studio

http://www.ti.com/tool/smartrft m-studio

[9] SmartRF Flash Programmer

http://www.ti.com/tool/flash-programmer

[10] SmartRF Packet Sniffer

http://www.ti.com/tool/packet-sniffer

[11] SmartRF Flash Programmer User Manual

http://www.ti.com/lit/swru069

[12] PurePath Wireless Configurator

http://www.ti.com/tool/purepath-wl-cfg

[13] PurePath Wireless Commander

http://www.ti.com/tool/purepath-wl-cmd

[14] SoC Battery Board product web site

http://www.ti.com/tool/soc-bb

[15] I AR Embedded Workbench for 8051

http://www.iar.com/ew8051

All banana pi product

- **banana pi BPI-M1 allwinner A20 dual core single board computer**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-m1/content/en/

- **banana pi BPI-M1+(BPI-M1+ plus) allwinner A20 dual core single board computer**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-m1-bpi-m1-plus-/content/en/

- **banana pi BPI-M2+（BPI-M2 Plus,BPI-M2+ edu）allwinner H3/H2+/H5 quad cord single board computer**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-m2-/content/en/

- **banana pi BPI-M2 Ultra allwinner R40/V40 quad core single board computer**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-m2-ultra/content/

- **banana pi BPI-M2 Berry allwinner R40/V40 quad core single board computer**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-m2-ultra/content/

- **banana pi BPI-M2 Magic allwinner R16/A33 quad core single board computer**

  gitbook online datasheet: https://bananapi.gitbooks.io/banana-pi-bpi-m2-magic-iot-development-board/content/

- **Banana pi BPI-M3 allwinner A83T (R58 H8) octa-core single board computer**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-m3/content/en/

- **banana pi BPI-M64 allwinner A64/R18 64 bit single board computer**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-m64/content/en/

- **banana pi BPI-R1 allwinner A20 dual core smart router board**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-r1/content/en/

- **banana pi BPI-R2 MTK 7623N smart router board**

  gitbook online datasheet:https://bananapi.gitbooks.io/banana-pi-bpi-r2-open-source-smart-router/content/

- **banana pi BPI-D1 open source IP camera board**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-d1/content/en/

- **banana pi BPI-G1 open source IoT development board**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-g1/content/en/

- **banana pi BPI Accessories**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-accessories/content/en/

- **BPI Open debugger burn development tool board**

  gitbook online datasheet:https://bananapi.gitbooks.io/bpi-open-debugger-burn-board/content/en/

Stop production

- **banana pi BPI-M2 allwinner A31s quad core single board computer**
- gitbook online datasheet:datasheet: https://bananapi.gitbooks.io/bpi-m2/content/en/

BPI 4.0 customized Server

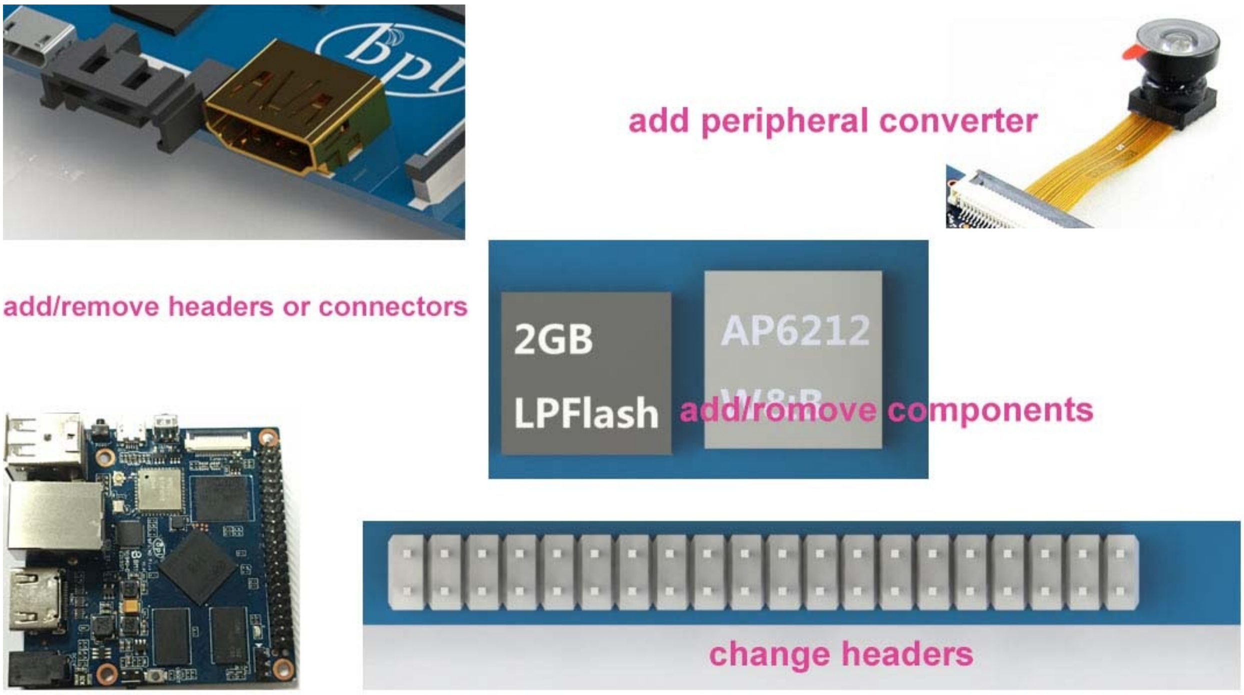
Having been doing R&D in embeded systems for more than 12 years. Our experienced teams are able to help you to carry out your dream. Whether you want to customize banana pi or want to design a computer from scratch, we got you covered. Also, our modern ISO factory spans over 10,000 square meters can help you mass manufacture products to hit the market

Our factory:Sinovoip In order to meet the companys development needs, and further production capacity and product quality. Sources Communication shareholders decided to invest in new plant to build their own, the new factory site is located in manholes and covers an area of more than 10000 square meters,equipped with full range of production equipment and high quality technical management personnel . We have complete SMT production lines, plug-ins production line, assembly line, production line testing.

- your Idea, we will help you optimize and design.
- your Design, we will help you bring it to live.
- your Product, we will help you mass produce it.

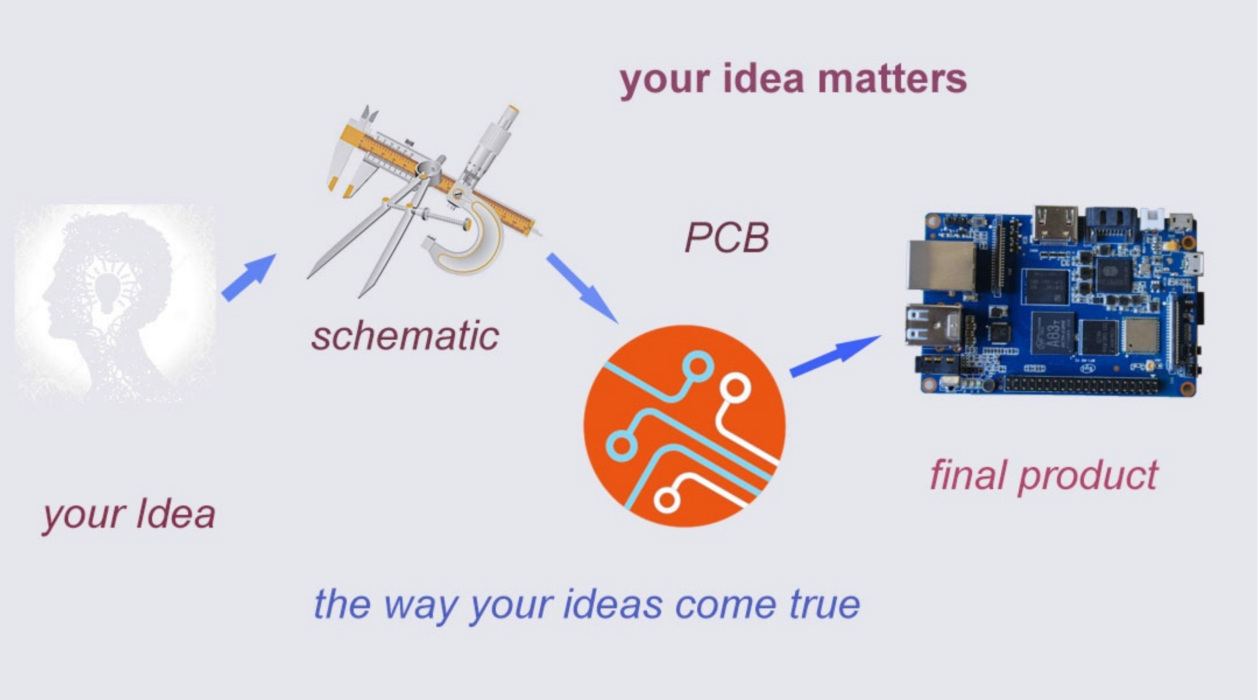
Customize Pi

If you want to tailor your Banana pi to your specific use or to minimize the cost for mass production purpose, you are coming the right place. We provide the customization service of banana pi such as remove/add headers or connectors,change component layout,add/remove components,change interfaces etc.



Start from scratch (ODM)

An idea flashes through your mind in your dreams or a solution bearing in your mind for a long time, which you think would turn out to be great gadget to hit the market, but you are worrying about how to start and realize it without R&D and manufacturing, now that's no longer a trouble to you. Taking advantage of our expertise,we provide full ODM service for you. We let you have your sample products from scratch within 45 days. Don't wait,come to realize your dreams.



Have a prototype (OEM)

You are an expert, you designed a wonderful device that most people would want to have it, you knew it quite well that your success is just around the corner. The only last step is to produce it,but without manufactruing capability..., no problem, let us carry you through. Our 13 years of SCM experience and mass manufacturing facilities enable you free from quality issue,delivery...