DZ-LINK 使用说明书 V1.0



一. DZ-LINK 是什么? 具有什么功能?

DZ-LINK 是支持芯唐 MO 的开发工具,具有 NU-LINK 所具有的一切功能,可对 MO 进行程序、数据、配置位烧写,还可以利用它配合 KEIL MDK 等开发环境进行仿真调试。





(注:所用到的管脚定义—9,10为GND,7为RST,5为CLK,3为DAT,2为KEY,1为VCC; 排线插在1-9的排针上;2和10可接入按键作脱机烧写用)

二. 如何设置 M0 在 MDK 环境的下载和仿真?

首先,你的电脑必须安装好 MDK4.1 或以上版本软件,且装好芯唐的 M0 插件

 NuMicro Keil uVision® IDE driver (Build 4228) V1.03.zip (芯唐官方网下载)
 使用 DZ-LINK 按上述连接目标板后,目标板不用另外供电,由 DZ-LINK 直接供电。

1) 按下图所示方法打开一个例子工程:



则出现界面

| 🔣 Smpl_DrvGPIO – µVision4 | |
|---|---|
| <u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject Fl <u>a</u> sh | <u>D</u> ebug Pe <u>r</u> ipherals <u>T</u> ools <u>S</u> VCS <u>W</u> indow <u>H</u> elp |
| 🗋 🗃 🖬 🖉 🕺 🛍 🖄 🧐 | ▶ + → * & & & 律 津 /// /// ❷ |
| 😵 🏥 🎬 🥔 🔜 📴 Smp1_GPIG | |
| Project 🥑 🗙 | Smp1_DrvGPIO. c |
| E- 🔄 Smpl_GPIO | 001 [] /* |
| 🖻 🤤 CMSIS files | 002 /* |
| 🗄 🔛 core_cmO.c | 003 /* Copyright(c) 2009 Nuvoton Technology Corp. |
| 🗄 🔛 system_NUC1xx.c | 004 /* |
| startup_NUC1xx.s | 005 /* |
| 🖻 🔄 Source files | 006 #include <stdic.h></stdic.h> |
| 😟 🔝 Smpl_DrvGPIO. c | 007 #include "NUC1xx.h" |
| 🖻 🔄 Library files | 008 #include "Driver\DrvGPIO.h" |
| 🖃 🔛 DrvGPIO. c | 009 #include "Driver\DrvUART.h" |
| stdio.h | 010 #include "Driver\DrvSYS.h" |
| nuclxx. h | 011 |
| core_cmO.h | 012 - |
| - 🔜 stdint. h | 013 void GPABCallback(uint32_t u32GpaStatus, uint3 |
| system_nuclxx. h | 014 4 |
| sysinfra.h | 015 printf("GPAB Interrupt! GPA:0x%04x GPB:0x |
| moduleid.h | 016 |
| drvgpio. h | 017 /* Toggle LED */ |

点击上图红色圈圈,出现界面

| 👿 Sept_Dreepio — p Ties | ion4 | | | | | | | |
|---|--|--|---|---|-------------------|----------------------|---------------|---------------------------------------|
| Eile Edit Vies Broject Fi | igen Debig Pegipheral | a joole gv | os ≣indom Help | | | | | |
| | 70 000 10 | 8 B B | 律 注 //i //ii Mi | | M | 1 m 🔍 | | a 🖬 • 🔦 |
| 🕸 🕮 🖉 🔤 🕎 🔤 | 1_0920 🙀 🖌 | 6 6 3 | | | | | | |
| Project 🧭 | Sep1_Drv6 | 10. c 🔛 | | | | | | |
| 🖯 🔛 Sapl_6910 | 196 SYSC | R-SPURCOR. | XTL128 FN = 1: | | | | - | 1 |
| E CMSIS files | Options for Targe | t Supl_GP | 10' | | | | 2 | 7 |
| rystes_BUCinx, s rystes_BUCinx, s rystes (Blas) rystes | Bevire Target Desp Destere 14, Vender Nucraten Dewe NUC120LE3M Tablet ARM C NUC1000C O NUC100C O N | K I III KIN AUCTON AUCTON VOTON AUCTON IIII IIIII IIIII IIIII IIIII IIIII IIII | CPU Cole ARM 32bit Contex 40 Vences ARM 32bit Contex 40 Vences 1286 Embedded A1 1566 Embedd | Lisker Bebu CPU dook up to 5 Right Right AM | E Unilinie | | | +/1n, 1 (/nu, 1 |
| | | | | | | | is pi | n then close it |
| | | OK | funcel | Jefailt | ts | Halp | 1 | |
| | | | | and an order | stormed . | | | <u>1</u> |
| ullé Cetpet | | | | | | | | 19 |
| <pre>iver command #2: frome!f '.\obj\CodeTemplate.axf' .cep "E:\\wusengjun\\\F9</pre> | text -= ".\sbj - 0 Error(s), 0 般文档\\芯樓\\SUCL | <pre>\CodeTemp1 Warning(s) 00SeriesBS</pre> | ste.axf"out - F_CMSIS_v1.02. | 105//NAC7008 | odeTemplate | e.txt" NuvetonPla | tform_Kei | 1/\Sample/\Driver/\ |
| \$ (| | | | | | 1 | | 8 |
| 1 11 14 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | | | H- | Link Debugger | De la composition | | Lan | - CAP 10 4 400 |
| 1 719E S S S | | 6 | | | - Bar | M 10. | 10 M W | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| <i>4开始</i> ◎ 6 ▲ 把型号给选好。 | >>> | 1.0 | r_vasis_v1.02. | Link Debugger | | arecone 18 | as a constant | |

2) 点击"Debug"选项,并选择 Nu-link Debugger

| Options for Target 'Smpl_GPIO' | |
|--|--|
| Device Target Output Listing User C/C++ A | sm Linker Debug Utilities |
| C Use Simulator Settings | |
| Load Application at Startup Run to main() Initialization File: | Altera Blaster Cortex Debugger Stellaris ICDI Initializatio Signum Systems JTAGjet Cortex-M/R J-LINK/J-Trace Edit. |
| Restore Debug Session Settings | Restore ULINK Pro Cortex Debugger Wetch Windows Memory Display |
| CPU DLL: Parameter: | Driver DLL: Parameter: SARMCM3.DLL |
| Dialog DLL: Parameter: | Dialog DLL: Parameter: |
| OK Can | cel Defaults Help |

选好后,确保 DZ-Link 和 MO 板子已连接好,点击红色圈的"Settige 也 如 DZ-Link 跟 MO 板子连接正常,则会出现 DeviceID 号,如板子连接不正常,则报错且 Device ID 的栏里没有序列号。

| Options for Target 'Smpl_GPT Device Target Output Listing C Use Simulator Limit Speed to Real-Time V Load Application at Startup Initialization File: | USB SW Adapter UCE Version: 4033 Device Family: Cortex-M Device ID: 0BB11477 | ies ✓ Settings |
|---|---|-------------------|
| Restore Debug Session Settings Breakpoints IV Toolbox Watch Windows & Performance . Memory Display CPU DUL: Parameter: | Port: SW V Max Clock: 1MHz V Chip Select Chip Type: NUClxx V | olbox |
| Dialog DLL: Parameter: | Reset Options Reset: Autodetect | |
| OK | Cancel Defaults | Help |

3) FLASH Programing 设置,点 "Utilite题 🖞 ---选择 "Use Target Driver for Flash Programming" ---选择 Nu-Link Debugger---点击 "Settigs 🛛 🖞

| Options for | Target 'Smpl_GPIO' | × |
|-----------------|--|---|
| Device Target | Output Listing User C/C++ Asm Linker Debug Utilities | |
| Configure Flash | n Menu Command | 1 |
| Use Targe | t Driver for Flash Programming | |
| | Nu-Link Debugger Settings Update Target before Debugging | |
| Init File: | ULINK Cortex Debugger Signum Systems JTAGjet | |
| C Use Extern | Contex-M/R J-LINK/J-Trace ST-Link Debugger | |
| Command: | | |
| Arguments: | | |
| | - Fran Independent | |
| | | |
| | | |
| | | |
| | | |
| | | |
| <u>y</u> | | |
| | OK Cancel Defaults Help | |

则出现以下界面

| Iash Download Flash Select Flash | For NUClass RAM for Alg Start Ox20 | orithm | - Download Functio C Erase Full (C Erase Sector | n Chip 🔽 Program Flash 's 🔽 Verify Flash |
|---|--|---------------------|--|---|
| APROM LDROM -Programming Algor Description NUC1xx 128kB | ithm Device Type ONCHIP Flash | Device Size 128K | Address Range 000000000H - 0001: | Flash Breakpoint Flash Breakpoint Flash BP Config0/1 |
| < | OK | Ca | ncel | Configure |

界面默认已选择 "Program fts f2" 和 "Verify Flash", 和 "Erase Sector", 若要擦除整个 FLASH, 则可选择 "Erase full chip"; 若要下载完程序后芯片复位跑程序,则需选择 "Reset and run"。 "Flash Select"为可选择程序烧入到 APROM 或 LDROM 中。 接着配置 Configur 经,点击 "Configur 经",出现界面

| Config Bit | |
|---|--------------------------|
| -Clock Source Selection- C external 12MHz crys | tal c 🗭 internal rc 22M |
| Brown Out Voltage | |
| ● 4.5V ○ 3.8V | C 2.6V C 2.2V |
| Boot Select | |
| • APROM | C LDROM |
| T Watch Dog Enable | 🔽 Clock Filter Ens |
| 🦵 Brown out reset enal 🥅 Data Flash Enabl | 🦵 Brown out detector ens |
| Data Flash Base | 0×1F000 |
| Wirte Config Bit | Exit |

此时可根据需要,可选择外部 12MHZ 的晶振或是内部 22MHZ 的晶振;可选择相应的掉电检测电压;可选择程序复位后在 APROM 或是在 LDROM 开始执行等等。配置好后,点击"Wirte Confg B fo",即可把配置位的相关信息写入 IC。

4)回到程序主界面,点击下图的"LOAD"图标就可以下载了!



5)进入仿真——以上均正常后,点击下图所示红圈按钮,即可进入仿真进行断点和单步等 调试!



三. 如何使用 NuMicro ICP Programming Tool 工具对程序区、数据区、配置位进 行烧写?



1) 连好开发板,打开 ,出现界面,选择好目标芯片系列。

点击"继续",出现界面

| uner. | 芯片(C |) 语i | | | | | | | | | | | | | | | | | | |
|--|--|---|---|--|---|--|---|--|--|--|-------------|---|---|---|--|--|-------------|------------|---------------------------------------|-------------------------------|
| าบ | vo | T | Dr | 1 | | | - | - | - | | | | | | | | | | | |
| 连线状 | 态检测 É接 | 一己間 | f开 | | | | | | | | | | | | | | | | | |
| 芯) | 片型号 | | | | | | | | | | | | | | | | | | | |
| 载入文 | 件 | | | | | | | | | | | | | | | | | | | |
| | DROM | 文件 | ‡名: | | E:\ | wus | ong | jun\ | 大中 | 中科技 | 大/支 | :中₹ | 詳共 | 享()) | NC1; | ex IS | SP V1 | . 08\ | NUC1xx | ISP V: |
| | | | | | 大小 | h: 3 | 3944 | 字节 | 5, 1 | 交验 | 值: | e70 | 8 | | | | | | | |
| AP | ROMO | 文件 | ‡名: | | E: \ | wus | ong | jun \ | 下载 | 战文相 | 当いだ | 唐 | NUC | 1005 | Gerie | sBSI | _CMS | IS_v | 1.02.0 | 02\NUC: |
| | | | | | 大小 | h: 6 | 3 <mark>99</mark> 2 | 字节 | 5, 1 | 校验 | 值: | £90 | 5 | | | | | | | |
| 数据 | Flash | 文件 | ‡名: | | C:\ | Dat | a. h | ex | - | | | | | | | | | | | |
| | | | | | 未能 | 包加速 | 載文 | 件。 | | | | | | | | | | | | |
| 配置位 | 2 | | | | | | | | | | | | | | | | | | | |
| Ĩ | 1. | | | 1 | | | | - 1 L | | | 10 | | | | | 100 | 02240w | 10 202 | | |
| | X.AE. | HC | 査 値0 |]: | OxFF | FFFF | FFF | | 配首 | 值1 | • | 0x00 | 001F | 000 | | 180 | 日上間 | 比置 | > | × |
| 文件数 | xxe (据 | ЦС | 宜伯(| D: | OxFF 片. | FFFF FFF | FFF .ash | | 配直 | 值1 | : [] | 0x0(| DO1F 寓創 | 000 \$F1: | ash | 5 | F I P | 比查 | > | × |
| 文件数 LDROM | x.AE (据 APROM | 10 数: | 宜伯(居F1a |): .sh | OxFF 片. LDR | FFFF FFF ROM | FFF .ash AB | ROM | | (值1 数据 | : Fla: | 0x00 | DO1F 寓創 LDRO | 000 BF1: | ash APH | XOMO | F 上間 【数携 | 比查 BF1a | sh | ×. |
| 文件数 LDROM 00000 00000 00000 00000 00000 00000 0000 | APROM 0000: 0010: 0020: 0020: 0040: 0050: 0060: 0070: 0080: 0090: | 10 数 C 8 0C 00 00 00 00 AB 01 AB 01 AB 01 AB 01 AB 01 AB 01 | 音伯(居F1a 00 00 00 00 00 00 00 00 00 | 20 00 00 00 00 00 00 00 00 00 00 00 | OxFF 儿DR 9D 000 000 AB AB AB AB AB | COM COM COM COM COM COM COM COM | AI | ROM | 0 A1 00 A7 AB AB AB AB AB | 值1 数据 01 00 01 01 01 01 01 | Flas | 0x00 sh 00 00 00 00 00 00 00 00 00 0 | LDR A3 A5 A9 AB AB AB AB AB AB | 000 影F1: 01 01 01 01 01 01 01 01 01 01 | APF 00 00 00 00 00 00 00 00 00 00 | 000 00 00 00 00 00 00 00 00 00 00 00 00 | 数据 | EF1a | sh ③813 ○16: ○32: | ت ت ت ت |
| 文件数 LDROM 00000 00000 00000 00000 00000 00000 0000 | APROM 0000: 0010: 0020: 000: 000: 000: 000: 000: 000: 000: 0 | 10 数 C 8 0C 00 0C 0 | 音伯(居F1s 00 00 00 00 00 00 00 00 00 00 00 00 00 | 20 00 00 00 00 00 00 00 00 00 | 0xFF 片 LDR 9D 000 000 000 000 000 000 000 000 000 | EFFF EF1 20M 01 01 01 01 01 01 01 01 01 01 | AI | PROM 00 00 00 00 00 00 00 00 00 00 00 00 | 0 A1 00 A7 AB AB AB AB AB AB AB | 值1 数据 01 01 01 01 01 01 01 01 01 01 | F1as | sh 000000000000000000000000000000000000 | DO1F 离射 LDR(A3 005 A8 A8 A8 A8 A8 A8 A8 A8 A8 A8 A8 A8 A8 | 000 \$F1 01 01 01 01 01 01 01 01 01 0 | ash APH 00 00 00 00 00 00 00 00 00 00 00 | 000 00 00 00 00 00 00 00 00 00 00 00 00 | 数据 | | > ③ 8位 ① 16 ③ 32 | で で 位 「新」」 |
| 文件数 LDROM 00000 00000 00000 00000 00000 00000 0000 | APROM 0000: 0010: 0020: 0040: 0050: 0050: 0050: 0080: | 10 数 C 8 0C 00 0C 00 0C AB 01 AB 01 A | 唐F114 000 000 000 000 000 000 000 000 000 | D: sh 20 00 00 00 00 00 00 00 00 00 | OxFF 片 LDR 9D 000 000 AB AB AB AB AB AB | FFFF 上FI 30M 01 00 00 01 01 01 01 01 01 | FFF .ash AI 00 00 00 00 00 00 00 00 00 00 00 00 00 | PROM 00 00 00 00 00 00 00 00 00 00 00 00 00 | A1 AB AB AB AB AB AB AB AB AB AB AB AB AB | 值1 数据 01 01 01 01 01 01 01 01 01 | F1a: | 00x00 sh 000 00 00 00 00 00 00 00 00 00 00 00 0 | A3 A3 A3 A5 A8 A8 A8 A8 A8 A8 A8 A8 A8 A8 X0 X | 000 kF1: 0M 01 01 01 01 01 01 01 01 01 01 | ash APF 00 00 00 00 00 00 00 00 00 00 00 00 00 | COMO 00 00 00 00 00 00 00 00 00 00 00 00 00 | 「上□」 | EFI1a | > sh ③ 8位 ① 16 ③ 32 ² ■ | で 位 (立 (立)新) |

2)点击"连接",如是第一次连接,且固件版本比较低,可能会出现内部版本号不兼容,出现以下界面

| 连线状态 (停止 ⁾ 芯片 | 心 F (E) 参检测 检测 型号 |) iš 1 2 | いまし。 し い い 新 开 | | | | | | | | | | | | | |
|--------------------------------|--|---|--|--|---|---|---|--|--|--|--|--|---|------|------|--|
| 载入文件 | + | Rul | licro | ICP | Pro | gran | min | g Io | 001 | | | | | | × | - |
| LDR | ROM | 1 | | 应用程 | 序内 | 部版本 | 号不 | 匹配 | I. | | | | | | | NUC1xx ISP V: |
| 数据F | Flash | | | 如果您 1. 请· 2. 请· | 、要更到 不要在 拔下US | 新韧体 EICP 。 SB,按 | ,在 longl 住IC | - 更新 e上列 P dos | 韧体之 ─接任(ngle上 | .前 可设备 的按键 | 建时 | 再重新 | 新插。 | λuse | 8. | |
| 配置位 设计 文件数据 | 定居 | | | 您想开 | ·始更新 【 | 新初体 是(| :程序(<u>Y</u>) | ₽?][| 否(| <u>I)</u> |) | | | | | > |
| 配置位 设 文件数据 LDROM | 定 居 APROF | NO 3 | 数据Fla | 您想开 ash Li | ·始史 【 DROM | 新韧体 是(APR | :程序(<u>Y</u>) | 吗?][数据: | 否 (Flash | |) | APR | OMO | 数握 | Fla | sh |
| 配置位 设 文件数据 LDROM | 定 APRO/ 000: 010: 020: 030: 040: 050: 050: 050: 050: 050: | MO 200 (000 (00) (000 (000 (00) (000 (000 (00) (000 (00) (000 (0)))))))))) | 数据F1。 0C 00 00 00 00 00 01 00 01 00 01 00 01 00 | 您想开 20 90 00 00 00 01 00 01 00 01 00 01 00 01 00 01 00 01 | 加度 DROM DROM D 01 D 00 D 00 D 00 D 00 D 01 D 00 D 01 D | 新初体 是(APR(00 0 00 0 00 0 00 0 00 0 00 0 00 0 00 | 程序 (1) 0 A1 0 A1 0 A5 0 A5 0 A5 0 A5 | 马? 数据 01 00 01 01 01 01 | 否 () F1 ash 00 0 00 0 00 0 00 0 00 0 00 0 00 0 | D LDR 0 A3 0 A5 0 A5 0 A8 0 A8 0 A8 0 A8 | 01 01 01 01 01 01 | APR(00 00 00 00 00 00 00 00 | 0M0 00 00 00 00 00 00 00 00 00 | 数捷 | Fla | sh 8位 16位 32位 |
| 配置位 设: 文件数据 LDROM | 定 APRO/ 000: 010: 020: 030: 050: 050: 050: 050: 050: 050: 05 | MO 28 (000 (00) (000 (00) (0) (| 数据F1。 0C 00 00 00 00 00 01 00 01 00 01 00 01 00 01 00 01 00 | 您想开 20 90 00 00 00 AI 00 AI 00 AI 00 AI | 日本 中国 中国 中国 中国 中国 中国 中国 中国 中国 中国 | 新初体 是(APR(000000000000000000000000000000000000 | 程序 1) 0 A1 0 A1 0 A1 0 A1 0 A1 0 A5 0 A5 0 0 0 0 0 0 0 | 吗? 数据: 0100001 01101 01101 01101 | 否 () F1 ash 00 0 00 0 00 0 00 0 00 0 00 0 00 0 00 | D LDR 0 A3 0 00 0 A5 0 A5 0 A8 0 A8 0 A8 0 A8 0 A8 0 A8 0 A8 | 01 01 01 01 01 01 01 01 01 | APR/ 00 00 00 00 00 00 00 00 00 00 00 | 0M0 00 00 00 00 00 00 00 00 00 | 数捷 | Fla | sh 8位 16位 32位 |
| 配置位 设: 文件数据 LDROM | 定 APRO/ 000: 010: 020: 030: 040: 050: 050: 050: 090: 090: 090: | MO 3 CS (OO (AB (AB (AB (AB (AB (AB (AB (AB | 数据F1。 00 00 00 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 | 您想开 20 90 00 00 00 00 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 | 日本 中国 中国 中国 中国 中国 中国 中国 中国 中国 中国 | 新初体 是(APR(00 0 00 0 00 0 00 0 00 0 00 0 00 0 00 | 程序 Y) 0 A1 0 A0 0 AB 0 AB 0 AB 0 AB 0 AB 0 AB 0 AB | 马? 数据 01 01 01 01 01 01 01 | 否 () 00 0 00 0 00 0 00 0 00 0 00 0 00 0 0 | D LDR 0 A3 0 A3 0 A5 0 A5 0 A8 0 A8 0 A8 0 A8 0 A8 0 A8 0 A8 0 A8 | 01 01 01 01 01 01 01 01 01 01 01 | APR(00 00 00 00 00 00 00 00 00 00 | DMO 00 00 00 00 00 00 00 00 00 00 00 | 数据 | ÷F1a | sh 8位 16位 32位 |

在以上的界面点击"是",则更新韧体程序。更新完程序后,出现界面

| 文件 (E) 芯片 (C) | 语言(L) | | | |
|---------------|---|---|--|-----------------------|
| nuvo | NO | | | |
| 连线状态检测 | | | | |
| 断开 | 芯片已连接 | | | |
| 芯片型号 | NUC120LE3A | N LDROM: 4096 Byte | s, APROMO: 131072 Byte | s, Data: O Bytes |
| 载入文件 | | | | |
| LDROM | 文件名: | E:\wusongjun\大中科 | 支\大中群共享\NUClxx IS | P V1.08\NUC1xx ISP V: |
| | | 大小: 3944字节, 校验 | 值: e708 | |
| APROMO | 文件名: | E:\wusongjun\下载文 | 当\芯唐\NUC100SeriesBSF | _CMSIS_v1.02.002\NUC: |
| | | 大小: 6992字节, 校验 | 值: £905 | |
| 数据Flash | 文件名: | C:\Data. hex | | |
| | | 未能加载文件。 | | |
| 配置位 | | | | |
| 设定 | 配置值0: | OxFFFFFFFFF 配置值1 | : 0x0001F000 < | 片上配置 > 🛛 💌 |
| 文件数据 | | 片上Flash | 离线Flash | |
| LDROM APROMO | 数据Flash | LDROM APROMO 数据 | Flash LDROM APROMO | 数据Flash |
| 00000000: C | 8 OC 00 20 0 00 00 00 | 9D 01 00 00 A1 01 00 00 00 00 00 00 | 00 00 A3 01 00 00 00 00 00 00 00 00 | 2 3位 |
| 00000020: 00 | $ \begin{array}{cccccccccccccccccccccccccccccccccccc$ | 00 00 00 00 00 00 00 00 00 00 00 A7 01 | 00 00 A5 01 00 00 00 00 A9 01 00 00 | ○ 16位 |
| 00000040: Al | B 01 00 00 B 01 00 00 | AB 01 00 00 AB 01 AB 01 00 00 AB 01 | 00 00 AB 01 00 00 00 00 AB 01 00 00 | ○ 32位 |
| 00000070: 8 | 5 07 00 00 5 01 00 00 | AB 01 00 00 AB 01 AB 01 00 00 AB 01 AB 01 00 00 AB 01 | 00 00 AB 01 00 00 00 00 AB 01 00 00 | |
| 00000090: Al | B 01 00 00 B 01 00 00 | AB 01 00 00 AB 01 AB 01 00 00 AB 01 | 00 00 7D 07 00 00 00 00 AB 01 00 00 | - 刷新 |
| | 0 01 00 00 | AP 01 00 00 AP 01 | 00 00 48 01 00 00 | |
| | APRONO | □ 数相Flach | 回 配 罢 区 计选 面 | 田田 |
| | [▼] AT YOMO | T SXJELISU | | |
| | | | | 构建: 4136rc |

3) 在"载入文件"中的"LDROM"、"APROMO"、"数据 Flash"中选择要烧写的程序和数据。 (注: LDROM 为实现 ISP、IAP 用的程序区, APROMO 为用户的应用程序区,数据 Flash 为内 部的数据存储区),在"编程"的区中,根据需要把"LDROM"、"APROMO、"数据 Flash"给 选上,然后点击"选项",出现界面

| 编程选项 | | X |
|--|-------|---|
| ✓ 擦除 ✓ 烧写 ✓ 验证 □ 离线模式 ✓ 对离线模式数据力 | r密 | |
| 输入密码 | ***** | |
| 重复输入密码 | ***** | |
| () 默认 | 确定 取消 | |

为保证程序和数据能正确烧写,"擦除"、"烧写"、"验证"这三项最基本的功能要勾上。 点击"确定"。出现刚才的界面

| (件 @) 芯片 (C) | 语言① | - 14- | | |
|--|--|---|---|--|
| NUVO | ION | | | |
| 连线状态检测 | 芯片已连接 | | | |
| 芯片型号 | NUC120LE3A | JN LDROM: 4096 Byte | s, APROMO: 131072 F |)ytes, Data: O Bytes |
| 载入文件 | | | | |
| LDROM | 文件名: | E: \wusongjun\大中科 | 支\大中群共享\NUClx: | x ISP V1.08\NUC1xx ISP V: |
| | | 大小: 3944字节, 校验 | 值: e708 | |
| APROMO | 文件名: | E:\wusongjun\下载文标 | 当\芯唐\NUC100Serie | BSP_CMSIS_v1.02.002\NUC: |
| | | 大小: 6992字节, 校验 | 值: £905 | |
| 数据Flash | 文件 名· | C. Mata her | - | |
| Styler rash | 200 | →. (butth inth +能加載文件。 | | |
| 配置位 | | -7€ HE/Jµ#32×17 € | | |
| 设定 | 配置值0: | OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | : 0x0001F000 | < 片上配置 > 🗸 🗸 🗸 |
| | 1 | like Lawrence | THE ALE NO. | |
| 又行致活 | 1 xx | F_Flash | 出版Tash | |
| LUKUM APROMO | 一数据Flash | LDROM APROMO 数据 | Flash LDROM APRO | DMO 数据Flash |
| 000000000: C 00000010: 0 | 8 OC 00 20 0 00 00 00 | 9D 01 00 00 A1 01 00 00 00 00 00 00 00 | 00 00 A3 01 00 00 00 00 00 00 | 00 🔷 🙆 8位 |
| 00000020: 0 | 0 00 00 00 00 | 00 00 00 00 00 00 00 00 00 00 00 00 00 | 00 00 A5 01 00 0 00 00 A9 01 00 0 | 00 〇16位 |
| 00000040: A | B 01 00 00 | AB 01 00 00 AB 01 | 00 00 AB 01 00 / | 00 |
| 100000050: A | B 01 00 00 | AB 01 00 00 AB 01 | 00 00 AB 01 00 4 | 00 032112 |
| 00000050: A 00000060: A 00000070: 8 | B 01 00 00 B 01 00 00 5 07 00 00 | AB 01 00 00 AB 01 AB 01 00 00 AB 01 AB 01 00 00 AB 01 | 00 00 AB 01 00 00 00 AB 01 00 00 00 AB 01 00 | 00 00 |
| 00000050: A 00000060: A 00000070: 8 00000080: A | B 01 00 00 B 01 00 00 5 07 00 00 B 01 00 00 | AB 01 00 00 AB 01 | 00 00 AB 01 00 00 00 AB 01 00 00 00 AB 01 00 00 00 AB 01 00 00 00 AB 01 00 | |
| 00000050: A 00000060: A 00000070: 8 00000080: A 00000090: A 00000090: A | B 01 00 00 B 01 00 00 5 07 00 00 B 01 00 00 B 01 00 00 B 01 00 00 B 01 00 00 | AB 01 00 00 AB 01 | 00 00 AB 01 00 00 00 AB 01 00 | 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| 000000000: A 000000000: A 00000070: 8 00000080: A 00000080: A 00000000: A | B 01 00 00 B 01 00 00 5 07 00 00 B 01 00 00 B 01 00 00 B 01 00 00 B 01 00 00 | AB 01 00 00 AB 01 | 00 00 AB 01 00 00 00 AB 01 00 00 00 AB 01 00 00 00 AB 01 00 00 00 7D 07 00 00 00 AB 01 00 00 00 AB 01 00 | 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| 00000050: A 00000060: A 00000070: 8 00000080: A 00000090: A 00000090: A 000000000: A 000000000: A 000000000: A 000000000: A 000000000: A | B 01 00 00 B 01 00 00 5 07 00 00 B 01 00 00 B 01 00 00 B 01 00 00 C 01 00 00 C 01 00 00 | AB 01 00 00 AB 01 AB 01 00 00 AB 01 | 00 00 AB 01 00 00 00 00 AB 01 00 00 00 00 00 00 AB 01 00 00 00 00 00 00 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 00 00 00 00 00 |

此时,点击"开始",如果选择了"配置区",则会出现下面的界面

| Nullicro ICP Programming Tool | × |
|---------------------------------|-----|
| 因为要擦除配置区,整片Flash都必须要被打 确定 取消 | 察除! |
| 点击 "确定",出现界面 | |
| Nullicro ICP Programming Tool | |
| (2) 使用批量编程模式吗? | |
| 是似一香砚 | |

如果只是编程一片 M0,则选择"否";如果是批量烧写,则点击"是"。(注:如选择了批量烧写,则拔下已烧写的板子,插入新板则会自动检测烧入。)

四. 如何脱离上位机,利用 DZ-LINK 配合按键进行离线烧写?

如果需要离线烧入,则可先把程序烧入 DZ-LINK 里的 EEPROM,应用到工具 NuMicro ICP Programming Tool,步骤为

1) 点击下图的"选项",

| 文件(22) 芯片(C) | 语言(L) | | | |
|--|--|---|--|--|
| nuvol | noī | | | |
| 连线状态检测 | 芯片已连接 | | | |
| 芯片型号 | NUC120LE3A | N LDROM: 4096 Byte | s, APROMO: 131072 Bytes, Da | ta: O Bytes |
| 载入文件 LDROM | 文件名: | E:\wusongjun\大中科 | 技\犬中群共享\WUC1xx ISP V1. | 08\NUC1xx ISP V: |
| | | 大小: 3944字节, 校验 | 值: e708 | |
| APROMO | 文件名: | E:\wusongjun\下载文 | 垱\芯唐\WC100SeriesBSP_CMSI | S_v1.02.002\NUC: |
| | | 大小: 6992字节, 校验 | 值: f905 | |
| 数据Flash | 文件名: | C:\Data.hex | | |
| #399/2- | | 未能加载文件。 | | |
| 设定 | 配置值0: | OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | : 0x0001F000 < 片上配 | 置 > 🔽 |
| 文件数据 | | 片上Flash | 离线Flash | |
| LDROM APROMO | 数据Flash | LDROM APROMO 数据 | Flash LDROM APROMO 数据 | Flash |
| 00000000: C3 00000010: 00 00000020: 00 00000030: 00 00000040: A4 00000050: A4 00000050: A4 00000050: A4 | 8 0C 00 20 0 00 00 00 00 0 00 00 00 00 0 00 00 00 00 0 00 00 00 00 0 01 00 00 00 0 01 00 00 00 0 01 00 00 00 5 07 00 00 00 5 01 00 00 00 | 9D 01 00 00 A1 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 A7 01 AB 01 00 00 A8 01 | 00 00 A3 01 00 00 00 00 00 00 00 00 00 00 A5 01 00 00 00 00 A9 01 00 00 00 00 A8 01 00 00 | ● 8位 ○ 16位 ○ 32位 |
| 00000000: AB | B 01 00 00 B 01 00 00 P 01 00 00 | AB 01 00 00 AB 01 AB 01 00 00 AB 01 AB 01 00 00 AB 01 | 00 00 7D 07 00 00 00 00 AB 01 00 00 00 00 AB 01 00 00 | ✓ 刷新 |
| 编程 I LDROM | APROMO | 一数据Flash | | 开始 |
| | | | | 构建: 4136rc |

出现界面

| 编程词 | 先项 | | |
|------------------------|--|-------|--|
| ব <mark>ব</mark> ব ব ব |] 擦除] 烧写] 验证] 薀线模式] 对离线模式数据加 | u密 | |
| | 输入密码 | ***** | |
| | 重复输入密码 | ***** | |
| | 默认 | 确定取消 | |

2) 把"离线模式"选上,如要对离线模式的数据加密,则把"对离线模式数据加密"选上,同时写上密码(如果对离线模式的数据加密,则点击主界面的"刷新"时,需要填对密码,才能把 EEPROM 里的离线程序数据读出)。按"确定",回到主界面,点击"开始",按照提示操作即可烧写成功,出现界面



2)此时,可以把 NuMicro ICP Programming Tool 软件给关掉了!把仿真器与板子接上,同时 仿真器的 PIN2 和 PIN10之间接上按键(参考管脚定义图),按下按键,其中一个灯出现 红灯闪三下,然后变常红状态;另一个灯出现红灯保持一会,然后出现常绿状态,即表 示离线烧写成功!把烧好的板子取下,接另一块待烧写的板子,按下按键……,重复这 个步骤即可实现离线批量生产!

五. 如何调节 DZ-LINK 仿真器的输出电压?

利用 DZ-LINK 调试 M0, 仿真器可直接给 M0 供电,而不需另接电源。观察附录的原理图可 以看出,如果接上 R4,不接 R3,则输出电压为 3.3V;如果接上 R3,不接 R4,则输出电压为 5V; 而芯唐 IC 支持的电压范围在 2.5V—5.5V,用户可根据需要来选择输出电压。(注: R3 和 R4 为 0 欧电阻)



六. 附录:DZ-LINK 原理图