

# ULINK 简明使用手册

Heciang

QQ:176780142 Email:heciang@126.com Tel:13811129591

ARM7 TDMI 结构的 Keil 开发套件采用最新设计的超豪华 uVision3 集成开发环境，内嵌 C 编译器/汇编器/工程管理器/调试器等功能模块，是一款稳定/可靠/高效的开发工具，适用于不同层次的用户，完全满足从专业的应用开发工程师到初学嵌入式软件开发的学生的所有使用要求。类似于 8051 的智能平台将大幅度缩短您的开发周期，各大半导体厂商的所有 ARM 型号将逐一得到全面支持。

Keil uVision 调试器可以帮助用户准确地调试 ARM 器件的片内外围功能(I<sup>2</sup>C、CAN、UART、SPI、中断、I/O 口、A/D 转换器、D/A 转换器和 PWM 模块等功能)。ULINK USB-JTAG 转换器将 PC 机的 USB 端口与用户的目标硬件相连(通过 JTAG 或 OCD)，使用户可在目标硬件上调试代码。通过使用 Keil uVision IDE/调试器和 ULINK USB-JTAG 转换器，用户可以很方便地编辑、下载和在实际的目标硬件上测试嵌入的程序。


支持 Philips、Samsung、Atmel、Analog Devices、Sharp、ST 等众多厂商 ARM7 内核的 ARM 微控制器。

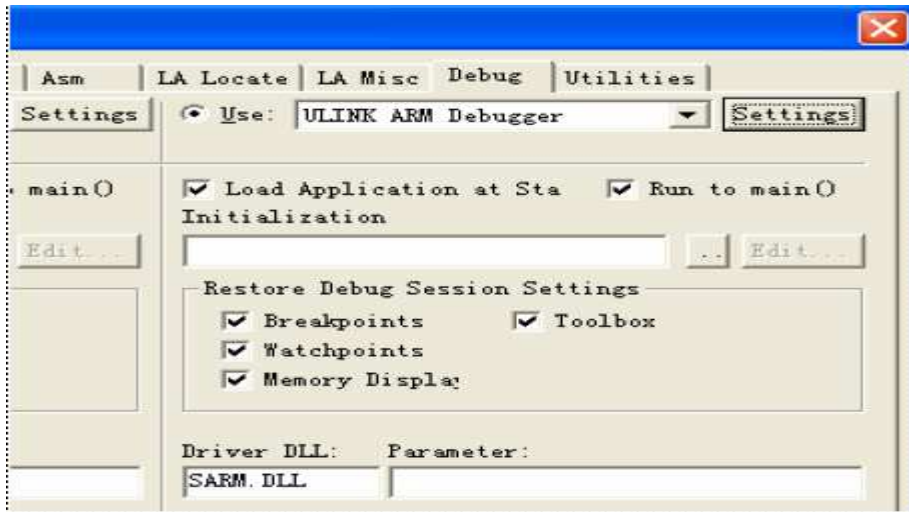


Keil 的开发环境用过 51 的都熟悉，keil 的模拟仿真及硬件仿真 (Monitor51) 功能是极其强大的，现在 ARM 的也不例外，在 ARM 是使用 Ulink 仿真器，使用 Ulink 有一点要注意的是，在配置完 Debug 环境为 ULINK ARM Debugger 外还要配置 Utilities 的选项，目的是为了把程序通过 Ulink 或 LPC210x\_ISPEXE 工具对 ARM Flash 先进行编程。

以下设置基于 KEIL FOR ARM 2.42A 及 ARM KIT LPC2142 实验板。

## 配置 Debugger

点击面板上的  按钮进入 Debug 选项设置。按下图一步一步进行设置。

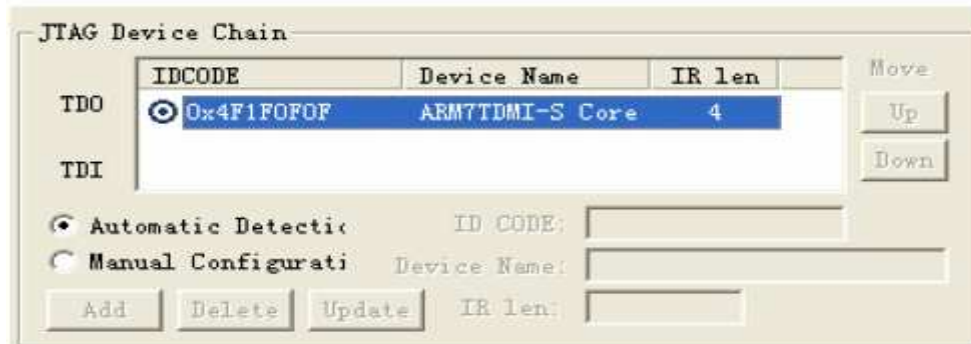


选中 ULINK ARM Debugger



点击 Settings 按钮进行设置。

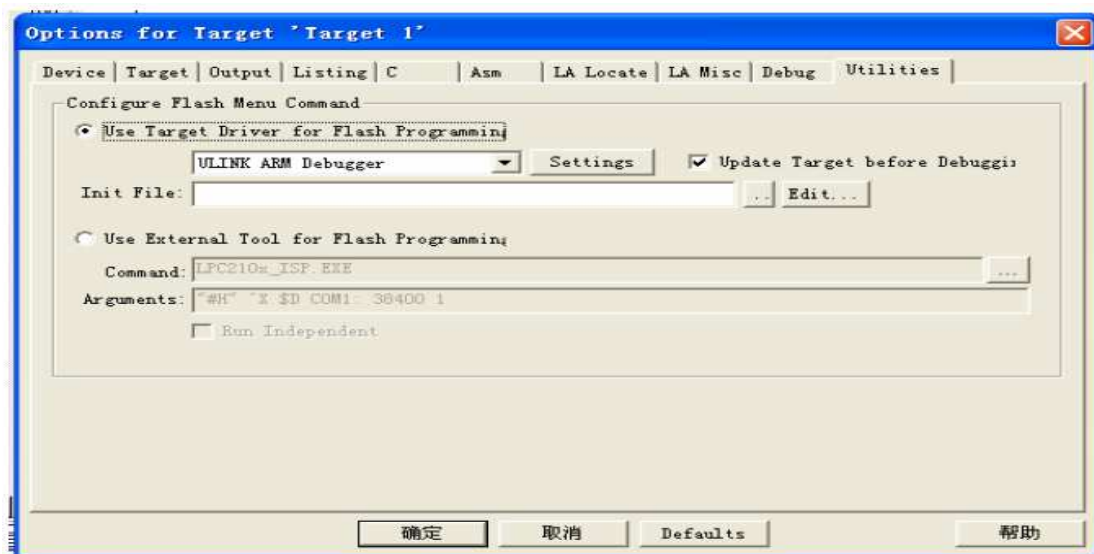
如果已经连上仿真器及目标板会显示如下信息，请用下图的默认设置。



点击代码校验

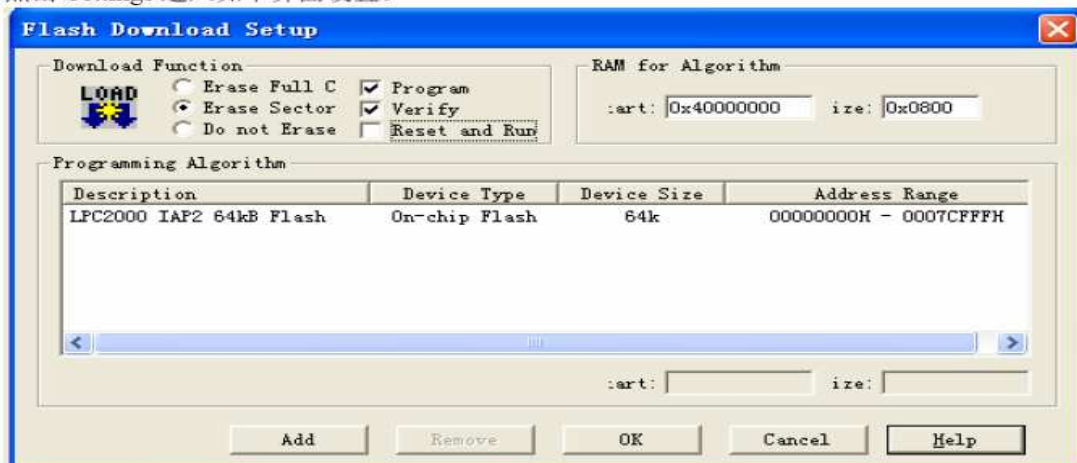


配置好上面的内容记得还要配置下面的否则无法进入仿真环境的（与 51 仿真环境的不同之处）



注意：Update Target before Debugging 这个选项要打勾。

点击 Settings 进入如下界面设置。

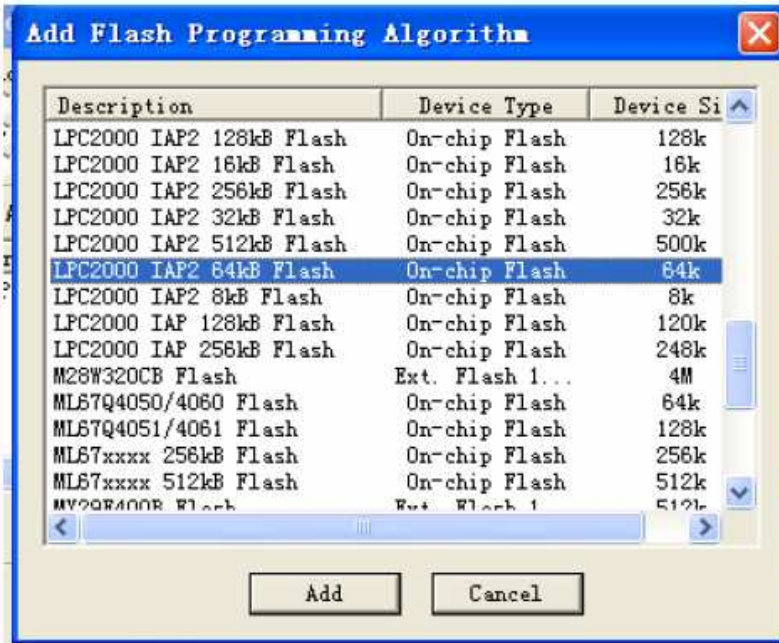


Download Function 按照用户需要进行设置，RAM for Algorithm 需要注意，START 是指目标芯片的RAM 起始地址，对于LPC2142 是0x40000000，SIZE 是指RAM 的大小，

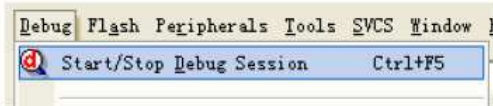
0x800,

Programming Algorithm 是指FLASH 编程算法，需要手工根据目标芯片进行添加，电击

 添加即可。



按照上面的设置进行设置后，即可完成硬件仿真环境的设置，只要程序修改编译后点击  或



会自动装载最新程序到目标芯片中并进入仿真环境，无需人工对 FLASH 进行烧录。

key - ARMion3 - [E:\work\ARM\_KIT\例程\key\_led\_Demo\key.c]

File Edit View Project Debug Flash Registers Tools DPC5 Window Help

Project Workspace

Register	Value
Current	
R0	0x0...
R1	0x40...
R2	0x00...
R3	0x00...
R4	0x00...
R5	0x00...
R6	0x00...
R7	0x00...
R8	0x00...
R9	0x00...
R10	0x00...
R11	0x00...
R12	0x00...
R13 (FP)	0x40...
R14 (LR)	0x00...
R15 (PC)	0x00...
CPSR	0x40...
SPSR	0x00...

```

098 int i=0;
099     break;
100 }
101     return i;
102 }
103
104 //+
105
106 //+
107 void delay (unsigned int i)
108 unsigned int n;
109 while(i>1)
110 {
111     for(n=65535;n>1;n--);
112     i--;
113 }
114
115
116
117
118 void main(void)
119 {
120     unsigned char i;
121     init_port();
122     while(1)
123     {
124         i=Scan_Key();
125         if(i!=0)IOCLR1 = (0x00000000<<i);
126         delay(100);
127         IOSET1 = 0x00ff0000; //ALL LED OFF
128         //delay(100)
129     }
130 }

```

key.c

Load "E:\work\ARM\_KIT\例程\key\_led\_Demo\Obj\key.ELF"

Output Window

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakS...

Command / Find in Files /

General Purpose Input/Output 1 (GPIO 1) - Slow Interface

GPIO	24	Bits	23	Bits	16	15	Bits	0	7	Bits	0
IO1DIR	0x00ff0000										
IO1SET	0x00000000										
IO1CLR	0x00000000										
IO1PIN	0x00000000										

A/D Converter 0

A/D Control

AD0CR: 0x00000000 SEL: 0x00  FSW

CLKS: 11.0k/10k CLKDIV: 0x00  BURST  EDGE

START: None A/D: 15000000

A/D Global Data & Status

AD0GR: 0x00000000 V/VIA: 0x0000  DONE  OVERSH

AD0STAT: 0x00000000 CDR: 0x00  ADINT

A/D Channel Data

AD0DR0	0x00000000	RESULT0	0x0000	<input type="checkbox"/> DONE0	<input type="checkbox"/> OVERSH0
AD0DR1	0x00000000	RESULT1	0x0000	<input type="checkbox"/> DONE1	<input type="checkbox"/> OVERSH1
AD0DR2	0x00000000	RESULT2	0x0000	<input type="checkbox"/> DONE2	<input type="checkbox"/> OVERSH2
AD0DR3	0x00000000	RESULT3	0x0000	<input type="checkbox"/> DONE3	<input type="checkbox"/> OVERSH3
AD0DR4	0x00000000	RESULT4	0x0000	<input type="checkbox"/> DONE4	<input type="checkbox"/> OVERSH4
AD0DR5	0x00000000	RESULT5	0x0000	<input type="checkbox"/> DONE5	<input type="checkbox"/> OVERSH5
AD0DR6	0x00000000	RESULT6	0x0000	<input type="checkbox"/> DONE6	<input type="checkbox"/> OVERSH6
AD0DR7	0x00000000	RESULT7	0x0000	<input type="checkbox"/> DONE7	<input type="checkbox"/> OVERSH7

A/D Interrupt Enable

AD0INTER: 0x00000100  ADINTEN0  ADINTEN4

ADINTEN1  ADINTEN5  ADINTEN6  ADINTEN7

ADINTEN2  ADINTEN3  ADINTEN7