1. 从0开始学ARM-安装Keil MDK uVision集成开发环境

原创 土豆居士 一口Linux 2020-11-20 11:55

收录于合集

#所有原创 206 #arm 22 #从0学arm 27

关于ARM的一些基本概念,大家可以参考我之前的文章:《0.到底什么是Cortex、ARMv8、arm架构、ARM指令集、soc?一文帮你梳理基础概念【科普】》

0. 如何学习arm?

ARM的知识点很多很杂,很多同学都遇到过类似的问题,单个的汇编指令一看就懂,但是结合uboot代码后,却一段都看不下去,网上搜资料,一大堆,屡不清楚条理,什么原因呢?

主要是各个知识点之间关系千丝万缕,融汇成一个整体,要想看懂类似于uboot这样的汇编代码,除了了解汇编指令,还要要了解整个SOC的架构,了解各个控制器的使用原理以及各个寄存器的使用方法,代码编译原理等等。如果不按照一定顺序学习,肯定会走很多的弯路。

一口君为了方便众多初学者方便入门,过滤掉一些枯燥的理论知识,一切以实用为主出发,下面会从环境安装开始,一步步教大家如何学习arm,最终能够学会分析uboot代码。

首先我们从安装环境开始吧。

一、KEIL、MDK、uVision、ARM之间的关系

1、KEIL

- . 1) 既是公司名称 , 同时也是KEIL公司所有的开发工具。
- . 2) 2005年被ARM收购。

2, uVision

- . 1) KEIL公司开发的集成开发环境(IDE)。
- . 2) 共有4个版本: uVision2、uVision3、uVision4、uVision5。



3、MDK

- . 1) 英文全称: Microcontroller Development Kit。
- . 2) MDK-ARM = KEIL MDK = RealView MDK = KEIL For ARM, 统一用 MDK-ARM 称呼。



MDK-ARM 为基于Cortex-M、Cortex-R4、ARM7、ARM9处理器设备提供了一个完整的开发环境。MDK-ARM专为微控制器应用而设计,不仅易学易用,而且功能强大,能够满足大多数苛刻的嵌入式应用。

MDK-ARM有四个可用版本,分别是MDK-Lite、MDK-Basic、MDK-Standard、MDK-Professional。所有版本均提供一个完善的C / C++开发环境,其中MDK-Professional还包含大量的中间库。

二. 安装

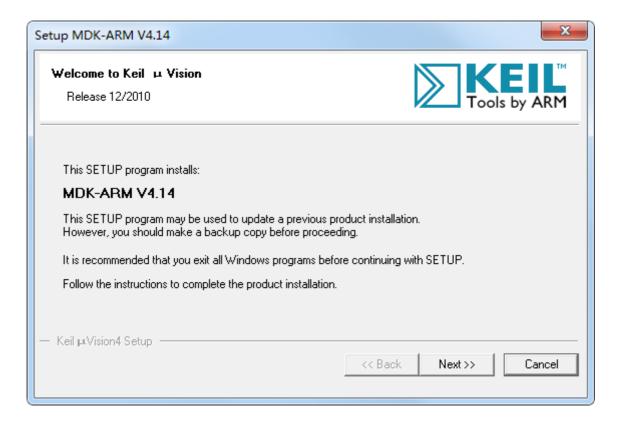
我们采用的安装包是Keil MDK-ARM 4.14版本,包括ARM的编译器和uVision 4集成开发环境。

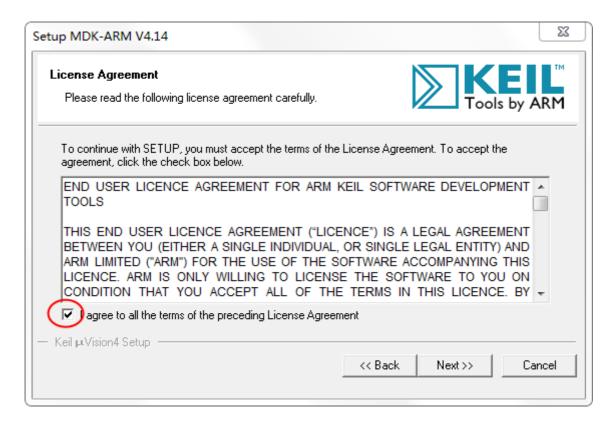
该环境具有良好的模拟ARM指令环境,非常适合大家学习ARM指令。

软件下载,后台回复【mdk】

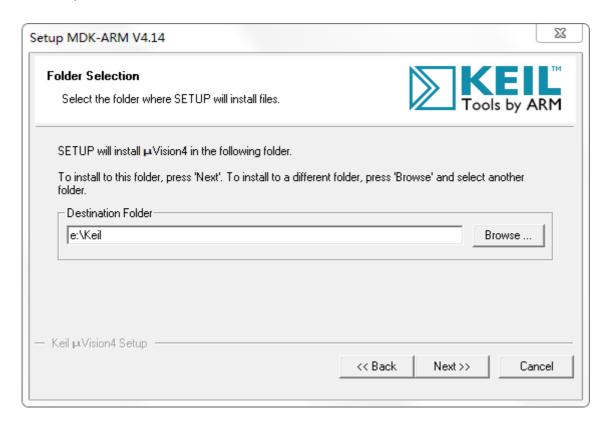


右边点击mdk414.exe -->以管理员身份运行

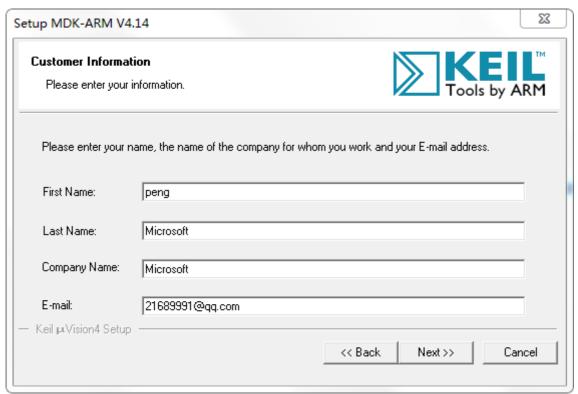


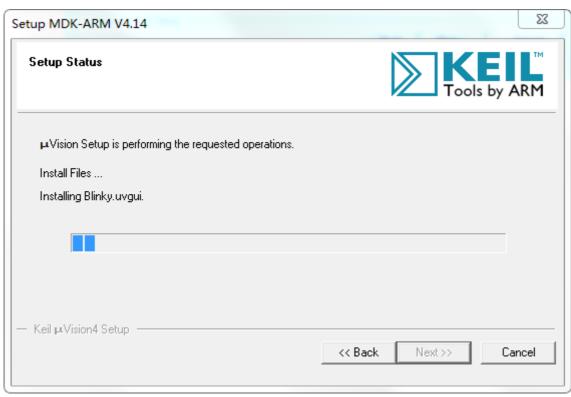


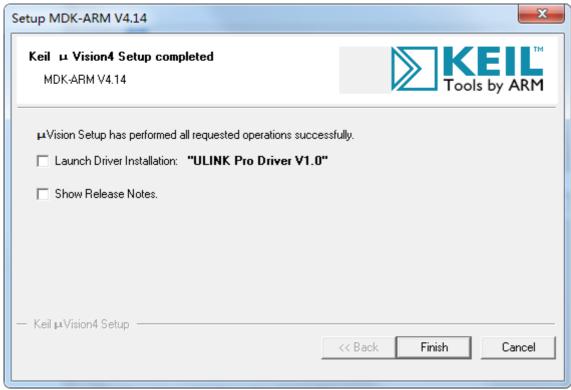
选择安装目录,尽量不要有中文目录:



输入任意的name、e-mail:





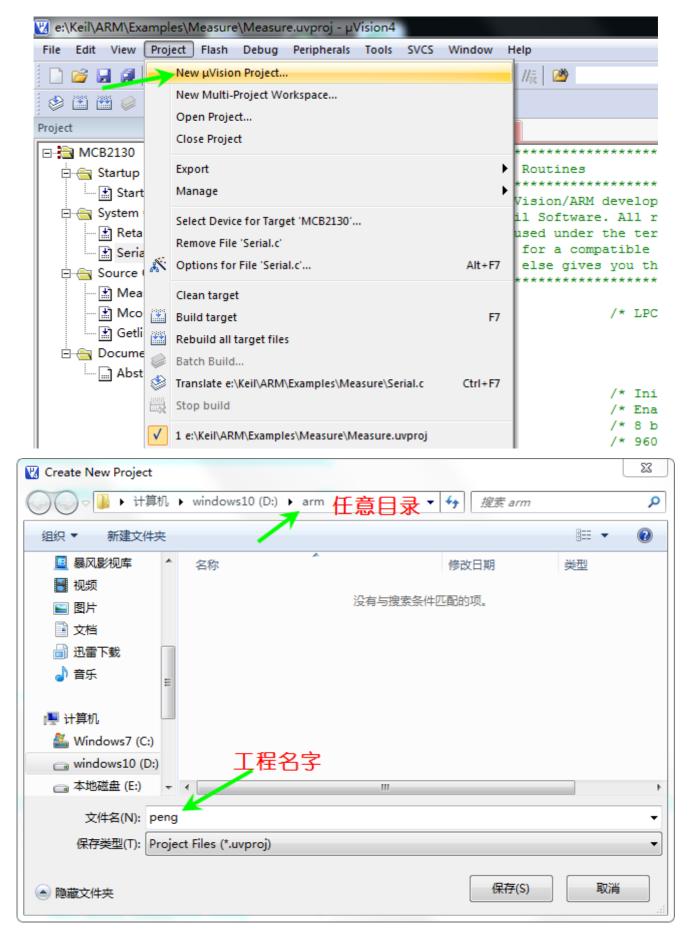


点击Finish即可:

最终在桌面上会有以下图标:

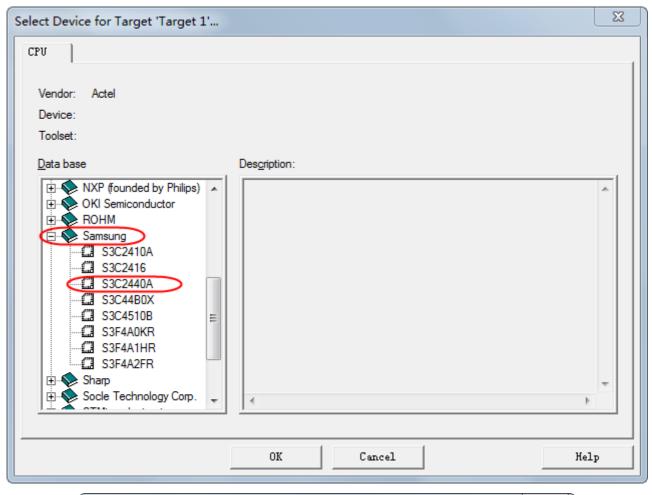


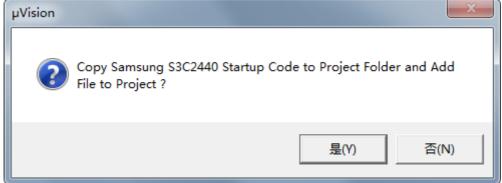
三、创建第一个工程



选择CPU->Samsung->S3C2440A 因为ARM指令集大部分的指令并没有太多差别,所以我们选择S3C2440A。

datasheet 下载,后台回复【exynos】





在这里插入图片描述

点击是,进入以下界面:

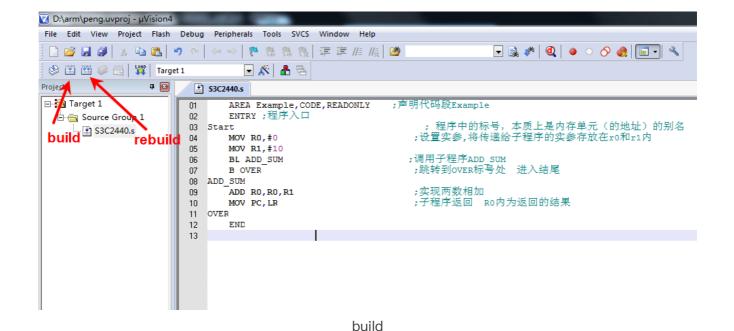
```
☑ D:\arm\peng.uvproj - µVision4
 File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
                                                                                     □ 🐸 🖫 🗿 🐰 🛍 🕮 😕 🖭 🗠 🗠 🕪 🥬 🍇 🍇 🏥 準 准 版 💁
 🏖 🖺 🥔 誤 Longet 1
                            ⅓ $3C2440.s
⊡- 🛅 Target 1
                                ;/* S3C2440.S: Startup file for Samsung S3C440
                            0002
   🖹 🔄 Source Group 1
      --- S3C2440.s
                            0004
                                 ;/* <<< Use Configuration Wizard in Context Menu >>>
                            0005
                                ;/* This file is part of the uVision/ARM development tools.
                            0007
                                ;/* Copyright (c) 2005-2008 Keil Software. All rights reserved.
                                ;/* This software may only be used under the terms of a valid, current, :/* end user licence from KEIL for a compatible version of KEIL software
                            0008
                            0009
                                ;/* development tools. Nothing else gives you the right to use this software. ^{*}/
                            0011
                            0012
                            0014 :/*
                                ; * The S3C2440.S code is executed after CPU Reset. This file may be
                            0015
                                ; * translated with the following SET symbols. In uVision these SET
                                ; * symbols are entered under Options - ASM - Define.
                            0017
                            0018
                                ; * NO_CLOCK_SETUP: when set the startup code will not initialize Clock
                            0019
                                ; * (used mostly when clock is already initialized from script .ini ; * file).
                            0021
                            0022
                                ; * NO_MC_SETUP: when set the startup code will not initialize Memory
                            0024 ; * Controller (used mostly when clock is already initialized from script
                            0025 ; * .ini file).
                                ; * NO_GP_SETUP: when set the startup code will not initialize General Ports
                            0027
                            0028 ; * (used mostly when clock is already initialized from script .ini
                            0030
                                ; * RAM_INTVEC: when set the startup code copies exception vectors
                            0031
                                ; * from execution address to on-chip RAM.
                            0032
                            0033
                            0034
                            0035
                                 ; Standard definitions of Mode bits and Interrupt (I & F) flags in PSRs
                            0037
                            0038 Mode USR
                                                 EQU
                                                         0x10
                                Mode_FIQ EQU 0x11
                           0040 Mode IRO
                                                 EQU
                                                         0x12
```

代码区域的代码是KEIL提供的基于处理器S3C2440A的一些测试代码,我们暂时先不关心这些代码,直接删除该文件的所有代码,拷贝如下代码【注意缩进】:

```
AREA Example, CODE, READONLY ;声明代码段Example
ENTRY;程序入口
              ;程序中的标号,本质上是内存单元(的地址)的别名
Start
MOV R0,#0
          ;设置实参,将传递给子程序的实参存放在r0和r1内
MOV R1,#10
BL ADD SUM
           ;调用子程序ADD SUM
           ;跳转到OVER标号处 进入结尾
B OVER
ADD_SUM
ADD R0,R0,R1 ;实现两数相加
          ;子程序返回 R0内为返回的结果
MOV PC, LR
OVER
END
```

编译

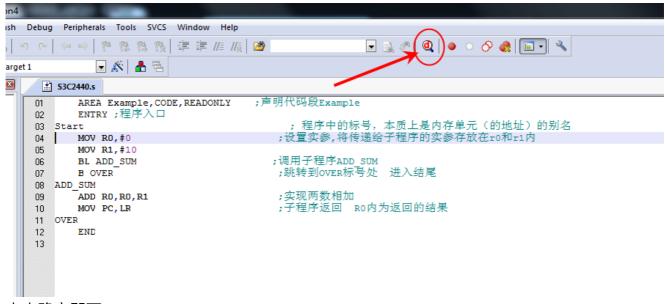
点击下图的两个按钮都可以实现代码编译:



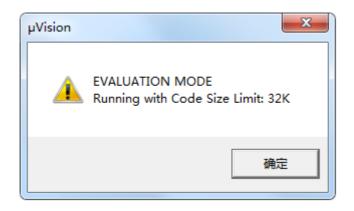
编译成功后,软件下方会有 **0 Error(s)**字样。

四、debug代码

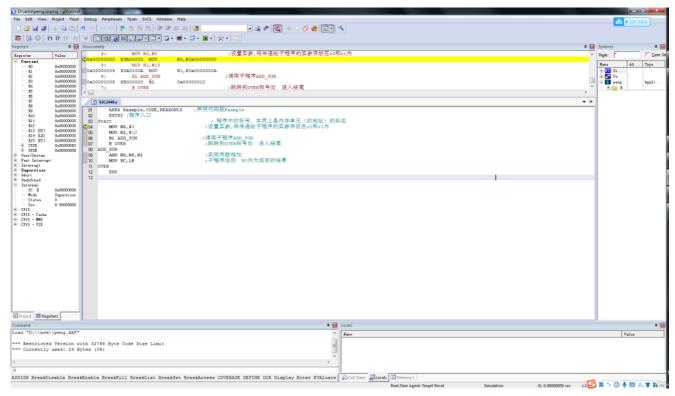
为方便讲解代码,我们后面调试全部进入debug界面,可以点击下面图标或者用快捷键ctrl+F5:



点击确定即可

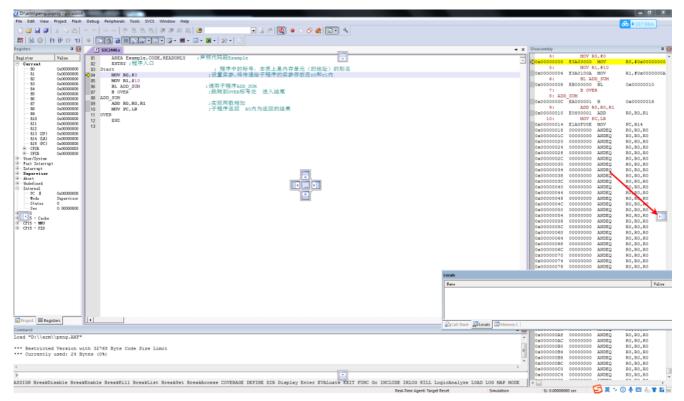


进入以下界面:



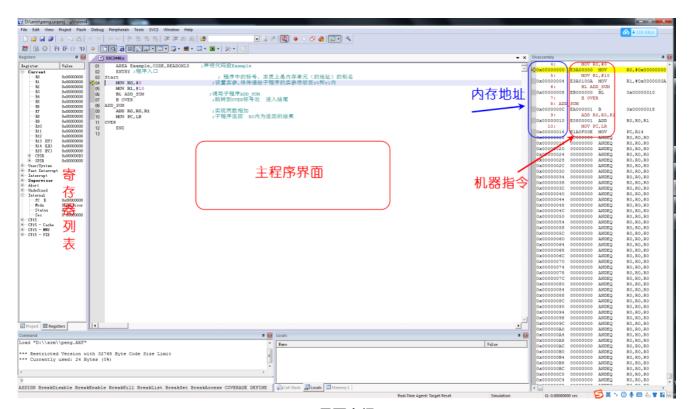
debug界面

调整界面布局,点击每个标签页面的标题栏拖拽,鼠标放到对应的边框三角形位置即可吸附上去:



在这里插入图片描述

调整界面如下:



界面介绍

- 1. 程序左侧的黄色箭头,表示程序指令执行的位置;
- 2. 左边的R0-R15等等,是ARM在不同模式下的寄存器列表;
- 3. 右边是汇编指令对应的内存地址和机器码
- 4. 单步执行F10, 进入某个函数执行F11

这个代码的含义,我们就暂时先不补详细说明,留待后面详细讲解。相信有汇编基础的同学对这个代码应该比较熟悉。

环境安装完毕,下面我们就可以开始ARM的理论学习了。

下一篇预告《2. 从0开始学ARM-CPU原理,基于ARM的SOC讲解》

其他网友提问汇总

- 1. 两个线程,两个互斥锁,怎么形成一个死循环?
- 2. 一个端口号可以同时被两个进程绑定吗?
- 3. 一个多线程的简单例子让你看清线程调度的随机性
- 4. 粉丝提问|c语言:如何定义一个和库函数名一样的函数,并在函数中调用该库函数

推荐阅读

- 【1】到底什么是Cortex、ARMv8、arm架构、ARM指令集、soc? 一文帮你梳理基础概念【科普】 必读
- 【2】Linux面试题100道,看看会多少?
- 【3】Modbus协议概念最详细介绍必读
- 【4】I2C基础知识入门
- 【5】多路复用IO内幕
- 【6】又一华为程序员进了ICU:压垮一个家庭,一张结算单就够了!必读

点击"阅读原文"查看更多分享,欢迎点分享、收藏、点赞、在看。

收录于合集 #从0学arm 27

下一篇·2. 从0开始学ARM-CPU原理,基于ARM的SOC讲解

阅读原文

喜欢此内容的人还喜欢

终结篇_万字总结Spring声明式事务

小东方不败



Konva实现图片自适应裁剪

A逐梦博客



理论先行-CAP定理

架构染色

