

# K60开发环境的安装与配置简介

胡春旭

华中科技大学  
瑞萨高级嵌入式控制器实验室

April 15, 2012

# 为什么要用开源？

## windows or Linux

windows 说 我的图形界面漂亮！

linux说 我不用独显.. ..

windows 说 我的安全性强！

linux说 我不用杀毒.. ..

windows 说 我的代码编辑器丰富！

linux说 我只有vim和emacs.. ..

windows 说 我的游戏多！

linux说 我是技术男.. ..

windows 说 我的应用程序多！

linux说 我不用破解.. ..

windows 说 我可以挂QQ！

linux说 好吧，我输了

# 怎么开源?

我要hello world!!

- printf?!
- vim or emacs?!
- 工具链?!
- 交叉编译?!
- Open.. ..?

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍



# ARM Cortex处理器

## ARM(Advanced RISC Machines)

**设计产品** 高性能、廉价、耗能低的RISC处理器

**技术特点** 性能高、成本低、能耗省

**应用领域** 嵌入控制、消费/教育类多媒体、DSP和移动式应用等

## ARM Cortex

**Cortex-A** 开放式操作系统的高性能处理器(手机、掌上电脑)

**Cortex-R** 面向实时应用的卓越性能(打印机、数码相机、汽车)

**Cortex-M** 面向具有确定性的微控制器应用的成本敏感型解决方案(手表、智能玩具、MCU)

# ARM Cortex-M4处理器

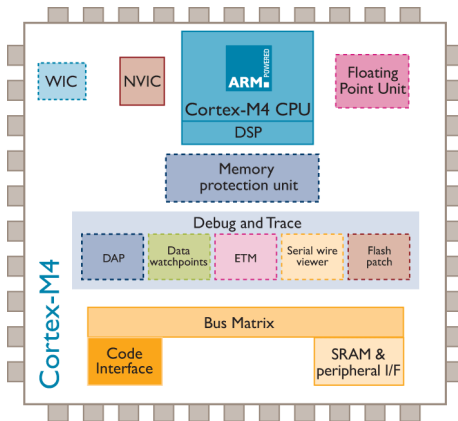


Figure: Cortex-M4内核

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍

# 基于Cortex-M4的处理器

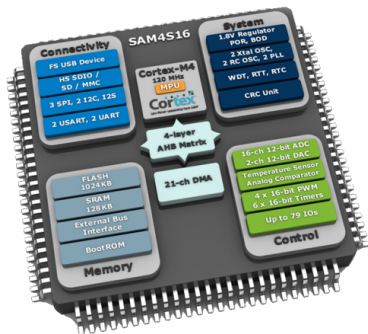


Figure: 基于Cortex-M4内核的一款处理器

# Kinetis60单片机

## 简介

飞思卡尔半导体推出Kinetis系列，这是基于新ARM®Cortex-M4处理器的90纳米32位MCU，开创了其微控制器(MCU)领先地位的新纪元

## 性能特点

- 具有超强可扩展性的低功耗、混合信号微控制器
- 每个系列提供了不同的性能，存储器和外设特性
- 引脚、外设和软件上可兼容
- ARM、飞思卡尔以及第三方的技术支持

# Kinetis60单片机

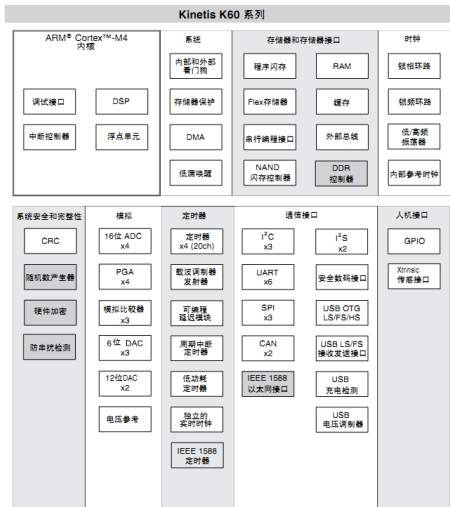


Figure: k60单片机模块图

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍

# 开发环境的四个部分

## ① 交叉编译工具链

- 核心部分，编译链接代码

## ② 守护进程

- 接收各种操作命令(复位、暂停、读/写等)

## ③ 硬件接口板

- 连接PC与单片机

## ④ 集成开发环境

- 整合工具链，提供图形界面



# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍

# 交叉编译工具链

## 简介

嵌入式设备由于不具备一定的处理器能力和存储空间，程序开发一般用PC来完成，然后将可执行文件下载到嵌入式系统中运行。

## CodeSourcery工具链

- 一个基于GNU工具链的完整的软件开发环境
- 包括GNU C/C++ 编译器、GNU汇编程序和连接器、运行时间库、源码级和汇编级调试器、支持微处理器调试接口与MIPS系统Navigator探针和MIPSSim仿真器的连接
- 支持Windows以及Linux平台上开发，目标平台已经有很多，支持最新ARM内核

# 编译系统

## ① 预处理器

- 根据以字符#开头的命令，修改原始程序

## ② 编译器

- 将程序翻译成.s文本文件，包含汇编语言程序

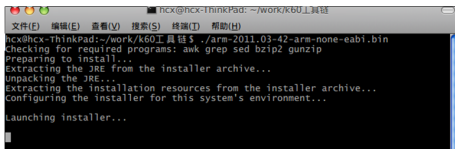
## ③ 汇编器

- 将.s文件翻译成机器语言指令

## ④ 链接器

- 将文本中的引用进行链接，得到可执行文件

# CodeSourcery工具链的安装



```
hcx@hcx-ThinkPad: ~/work/k60工具链
文件(E) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
hcx@hcx-ThinkPad:~/work/k60工具链$ ./arm-2011.03-42-arm-none-eabi.bin
Checking for required programs: awk grep sed bzip2 gunzip
Preparing to install...
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...
Launching installer...
```

Figure: 工具链的安装



```
hcx@hcx-ThinkPad: ~
文件(E) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
hcx@hcx-ThinkPad:~$ arm-none-eabi-g++ --version
arm-none-eabi-g++ (Sourcery G++ Lite 2011.03-42) 4.5.2
Copyright (C) 2010 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
hcx@hcx-ThinkPad:~$
```

Figure: 工具链的版本信息

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍

# Openocd简介

## 简介

OpenOCD是德国人的一个开源项目，主要目的是实现一个在线调试器，同时也能够实现程序烧写等目的，目前支持很多种芯片。

## 什么是在线调试器？

- 可以读/写内存
- 可以读/写CPU寄存器
- 可以单步执行和实时执行
- 具有硬件断点和触发器功能

# Openocd安装

## 1 下载源码

- `sudo apt-get install git`
- `git clone git://openocd.git.sourceforge.net/gitroot/openocd/openocd`

## 2 编译

- `./bootstrap` //这个命令主要是用于生成configure文件的,同时git出一些相关的代码
- `./configure - --enable-maintainer-mode - --enable-ft2232_libftdi - --enable-maintainer-mode`
- `make`

## 3 安装

- `sudo make install`

## 4 版本信息



```
hcx@hcx-ThinkPad: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
hcx@hcx-ThinkPad:~$ openocd -v  
Open On-Chip Debugger 0.6.0-dev-00477-gf1c0133 (2012-03-26-21:15)  
Licensed under GNU GPL v2  
For bug reports, read  
http://openocd.sourceforge.net/doc/doxygen/bugs.html  
hcx@hcx-ThinkPad:~$
```

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - **OpenJtag**
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍



# OpenJtag简介

## Jtag

JTAG也是一种国际标准测试协议（IEEE 1149.1兼容），主要用于芯片内部测试。基本原理是在器件内部定义一个TAP（Test Access Port测试访问口）通过专用的JTAG测试工具对内部节点进行测试。

## OpenJtag

Open JTAG是一个开源项目，它的目标是使嵌入式开发者能够通过JTAG接口，用开放的硬件和软件系统进行烧写、校验和调试。



# OpenJtag和Jlink的比较

## 操作系统

*OpenJtag* 可以用在Windows、Linux下

*JLink* 只能用在Windows下，在Linux下它的调试功能无法使用

## 集成开发环境

*OpenJtag* 能用在所有支持GDB调试协议的工具上

*JLink* 支持多种调试协议RDI、GDB调试，几乎所有Windows下的工具都支持

## 对Flash的烧写

- 对FLASH的烧写功能几乎一样强

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4 简介
  - Kinetis60 简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery 工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile 配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍

# Eclipse简介

## Eclipse

Eclipse是著名的跨平台的自由集成开发环境（IDE），Eclipse的本身只是一个框架平台，但是众多插件的支持，使得Eclipse拥有较佳的灵活性。许多软件开发商以Eclipse为框架开发自己的IDE。

## Eclipse版本

Eclipse基金会每年都会安排同步发布。

版本代号	平台版本	主要版本发行日期	SR1发行日期	SR2发行日期
Callisto	3.2	2006年6月26日	N/A	N/A
Europa	3.3	2007年6月27日	2007年9月28日	2008年2月29日
Ganymede	3.4	2008年6月25日	2008年9月24日	2009年2月25日
Galileo	3.5	2009年6月24日	2009年9月25日	2010年2月26日
Helios	3.6	2010年6月23日	2010年9月24日	2011年2月25日
Indigo	3.7	2011年6月22日	2011年9月23日	2012年2月24日
Juno（预定）	3.8及4.2	2012年6月27日	2012年9月28日	2013年2月22日
Kepler（预定）	4.3	2013年6月26日	2013年9月27日	2014年2月28日

# Eclipse安装

## ① 下载JDK



<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

## ② 安装JDK



`./jdk-6u26-linux-i586.bin`

## ③ 下载Eclipse



<http://www.eclipse.org/downloads/download.php?file=/technology/epp/c/cpp-helios-SR2-linux-gtk.tar.gz>

## ④ 安装Eclipse



Eclipse是不需要安装的，直接解压缩后就可以使用了

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍

# 开发流程

## ① 安装插件

- 安装CDT插件

## ② 建立工程

- 建立一个使用Sourcery G++工具链的Makefile工程

## ③ 代码编译

- Build Project

## ④ 工具配置

- 在“Debug Configurations”中配置gdb选项

## ⑤ 程序调试

- 打开openocd，点击调试

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 **Eclipse开发k60**
  - 开发流程
  - **makefile配置**
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍



# makefile简介

## make

Linux下的一款程序自动维护工具，配合makefile的使用，就能够根据程序中模块的修改情况，自动判断应该对那些模块重新编译，从而保证软件是由最新的模块构成。

## makefile

整个工程的编译规则，一个工程中的源文件不计数，其按类型、功能、模块分别放在若干个目录中，makefile定义了一系列的规则来指定，哪些文件需要先编译，哪些文件需要后编译，哪些文件需要重新编译，甚至于进行更复杂的功能操作

# makefile配置

我们使用的是makefile的通用模板，主要注意以下几点：

## 需要编译的main函数位置

- `DIR_TARGET = app/test_ntshell_cmd`

## 需要编译的源代码位置

- `DIR_SRC = startup common drivers modules library/ntshell`

## 优化等级

- `OPT = -O0`

## 编译需要用到的通用头文件位置

- `EXTRAINCDIRS = include (DIR_SRC)(DIR_TARGET)`

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍

# 链接脚本

链接脚本主要是对生成代码段放置位置的内存配置。我们提供了两套链接脚本:

## K60\_RAM.ld

- 程序调试使用，烧写速度较快
- 所有程序和数据存放在RAM中

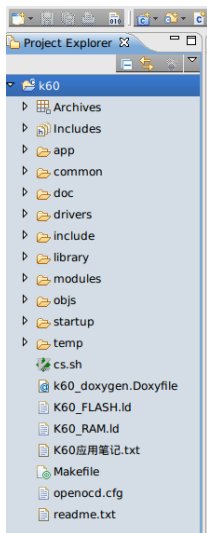
## K60\_FLASH.ld

- 离线调试使用，烧写速度较慢
- 将代码和数据放在flash中

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4 简介
  - Kinetis60 简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery 工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile 配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍

# 工程结构



# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍

# 代码规范

## 变量定义

- 全部采用小写字母，单词之间采用下划线连接
- 数据类型在头文件中t\_stddef.h进行定义，方便移植

## 函数

- 全部采用小写字母，单词之间采用下划线连接
- 采用宾动结构，将模块名或者对象名放在前面，操作放在后面

## 文件

- 全部采用小写字母，单词之间采用下划线连接
- .c文件内部用到的宏定义和声明都放在.c文件中，.h文件中只放对外接口和全局可见的宏定义和声明。



# 注释规范

```

gpio.c  ftm_decoder.h  ftm_decoder.c
/**
 * =====
 *  COPYRIGHT NOTICE
 *  Copyright (c) 2012  HUST-Renesas Lab
 *  ALL rights reserved.
 *
 *  @file    gpio.c
 *
 *  @brief   gpio驱动函数
 *
 *  @version 0.1
 *  @date    2012/2/14 15:35:40
 *
 *  @author: Hu Chunxu, hcx196@gmail.com
 *  =====
 *  @0.1  Hu Chunxu 2012/2/14 create original file
 *  =====
 */

```

Figure: 文件头注释

```

/**
 * @brief  lptmr  定时器初始化
 *
 * @param  clock  时钟选择
 *
 * @return  E_OK      初始化成功
 * @return  E_INIT    初始化失败
 *
 * @see    lptmr_registers_clear()
 * @note   在这个定时器中, 由于产生不了小于35.6ms的基准定时, 所以统一选择基准为50ms的定时。
 *         在移植过程中应该相应调整COMPARE寄存器和PSR寄存器中的预分频系数。
 */
ER lptmr_timer_init(uint32_t clock)

```

Figure: 函数注释

```

/** enable BE signals - note, not used in this example */

```

# 注释规范

```
/** enable BE signals - note, not used in this example */
FB_CSPMCR = FB_CSPMCR_GROUP2(2) | FB_CSPMCR_GROUP3(2);

/**
 * address(1~18)
 */
PORTD_PCR8 = PORT_PCR_MUX(6);          /* fb_a[16] */
PORTD_PCR5 = PORT_PCR_MUX(5);          /* fb_ad[1] */
PORTD_PCR4 = PORT_PCR_MUX(5);          /* fb_ad[2] */
PORTD_PCR3 = PORT_PCR_MUX(5);          /* fb_ad[3] */
```

Figure: 一般注释

```
/**
 * 常见的数据类型
 */
typedef int      bool_t;      /**< bool布尔值 */
typedef unsigned char  char_t;  /**< 无符号字符类型 */

typedef signed int    int_t;    /**< 自然大小的符号整数（32位） */
typedef unsigned int  uint_t;   /**< 自然大小的无符号整数（32位） */
```

Figure: 宏定义注释

# 主要内容

- 1 K60单片机简介
  - ARM Cortex-m4简介
  - Kinetis60简介
  - 开发环境简介
- 2 开发环境的安装与配置
  - CodeSourcery工具链
  - Openocd
  - OpenJtag
  - Eclipse
- 3 Eclipse开发k60
  - 开发流程
  - makefile配置
  - 链接脚本
- 4 k60软件平台介绍
  - 工程结构
  - 代码规范
  - 功能介绍

# 单片机启动

- ① 放置中断向量表
  - 中断向量表的前两个向量必须放在0x00000000和0x00000004地址
- ② 设置堆栈指针
  - 修改堆栈的初始值
- ③ 设置程序起始地址
  - 设置程序复位后的起始地址
- ④ 硬件配置（时钟、WDOG等）
  - 利用C语言进行硬件配置
- ⑤ 跳入main函数
  - 开始用户的函数

# 中断处理

K60支持120个中断和异常，其中前十六个是系统内核异常和中断，其他的是普通中断。

## 中断和异常

中断 功能调用

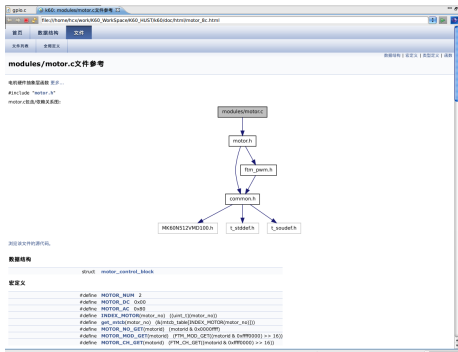
异常 错误退出

```
/**
 * @brief install an exception handler
 *
 * @param excno exception no.
 * @param exchdr exception handler
 */
void exc_install(uint32_t excno, FP exchdr)
{
    *((FP*)vector_table) + excno = exchdr;
}

/**
 * @brief set the priority of exception
 *
 * @param excno exception no.
 * @param pri exception priority
 */
void exc_set_pri(uint32_t excno, uint8_t pri)
{
    if (excno <= 15 && excno >= 4) { /* system exceptitons */
        *((volatile uint8_t *) (0xE000ED18 + excno)) = (pri & 0xF) << 4;
    }
    else if (excno > 15) { /* target define exceptions */
        NVIC_IP(excno) = (pri & 0xF) << 4;
    }
}
```

# Doxygen自动生成文档

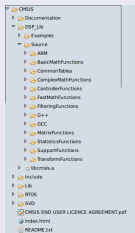
Doxygen是一个程序的文件产生工具，可将程序中的特定批注转换成为说明文件。



# 库

## CMSIS

- ARM® Cortex™ 微控制器软件接口标准 (CMSIS) 是 Cortex-M 处理器系列的与供应商无关的硬件抽象层。
- 包含DSP应用库（科学计算，信号处理，控制算法）



## ntshell

- 小型的嵌入式shell脚本，PC可以通过串口与单片机进行通讯

# End

谢谢大家的关注！