

# 第一个ARM实验（跑马灯）

原创

[LiuSuperfan](#) 于 2017-03-19 14:16:17 发布 6141 收藏 7

分类专栏: [arm](#) 文章标签: [arm 硬件 跑马灯实验 arm环境设置 第一次ARM](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/Superfan67/article/details/63684379>

版权



[arm](#) 专栏收录该内容


1 篇文章 0 订阅

订阅专栏

要成功完成基于ARM板的跑马灯实验首先需要: 硬件 (ARM板), 软件 (ADS1\_2, JLINK, GIVEIO)

现在我们准备好了软件, 先对软件进行安装!

一. 对ADS1\_2的安装:

1. 点击  进行安装,  
一直点击下一步, 出现下图, 可以对安装目录进行更改, 然后点击下一步。

2. 一直点击下一步, 遇到下图的提示框点击确定。


3. 一直点击下一步, 等待安装。

等待安装结束, 弹出下图, 如果有的同学安装停到100%很长时间, 不要惊慌, 请强制关闭, 然后到安装目录的Bin目录找到  
这个卡通人 (licwizard), 点击卡通人也会弹出下框。

4. 然后点击下一步。

5. 出现下图点击Browse...

6. 选择正确的license (这个杨老师给了, 也可以在网上下载)。

7. 然后你的ADS1\_2就可以完美运行啦。On, 不! 还有一点请对你的软件进行兼容性设置! 找到Bin目录下的  (或者开始  
程序直接能找到) 点击右键属性, 选到兼容性选项卡。将“以兼容模式运行这个程序 (XP)”, “以管理员身份运行次程  
序” 前面打 。

二. GIVEIO驱动的安装

1. 首先将GIVEIO下载下来，然后复制到C盘windows/system32下。
2. 然后打开控制面板，寻找，点击添加硬件。
3. 会弹出下面的框框，点击下一步，等待。
4. 选择“是，我已经连接了次硬件（Y）”，点击下一步。
5. 找到添加新的硬件设备，点击下一步！
6. 选择“安装我手动从列表选择的硬件（高级）”，点击下一步。
7. 找到“串口（COM和LPT）”，点击下一步。
8. 然后看图片，不说了。。。。
9. 从磁盘安装，找到你下载的GIVEIO驱动，添加进来。
10. 然后一直下一步，下一步，下一步。（好几个下一步），直到出现提示成功，这样你的GIVEIO驱动安装成功了，是不是很开心呢，已经完成2/3了。

### 三. JLINK的安装

点击这个就可以，这个安装很简单，一直下一步就可以。

现在，我们所需的环境及软件全部准备好，现在就开始我们的第一个跑马灯实验吧。

### 四. 创建项目

1. 打开我们下载成功的ADS，点击File→New, Project中的第一个ARM Executable Image 选择Location: 选择安装目录，然后直接输入项目名称。  
这里有一点注意，路径一定是没有中文的路径。  
这样我们的项目创建成功了，会生成一个.MCP的文件。
2. 接下来我们需要写代码了，再次File→New，选择File选项卡，输入你的File name，注意这里的文件名一定带格式，我们LED跑马灯需要三个文件：init.s, main.c, laday.s。。 创建好这三个文件，然后将代码写人。
3. 接下来选择Add Files...（在项目空白处点右键）将刚刚写的三个文件加进来。
4. 选择OK，你的项目就写完了，接下来，调试，烧写。
5. 简单的配置：点击上面图标的第一个或者菜单栏的Edit→DebugRel Settings

Target Settings中的Output Directory的ARM from ELF

6. 将Language Settings都设置为ARM920T

7. 接下来设置ARM Linker, Output选项卡里的simple image R0 Base将地址设置为0X30000000

8. 继续设置ARM Linker的Options选项卡。设置Image entry point设置为0x30000000

8. 设置最后一步, 将你开始的文件(初始化文件) init.o设置进去。这样设置结束了。

9. 点击上面图标的第三个(小刷子) make一下, 哇, 好开心, 没有报错, 说明我们的程序没有问题。(代码见附录)

10. 点击RUN(绿色三角) 出现如图下问题, 说明你的USB—JLINK没插好, 或者说你的JLINK驱动没有下载成功。找到安装目录打开JLink.exe, 输入USB命令, 如果提示Can not connect to J-Link via USB. 将JLINK拔掉重插, 或者卸载重下JLINK。(我的截图是没有插JLINK, 我选现在没有, 如果成功连接, 会有不同的提示。)

11. 选择调试方式, 老师上课讲过主要有两种调试方式, 一种是仿真调试, 一种是JTAG调试, 我们选择JTAG调试, 点击ADD, 将我们下载好的JLINK驱动加载进来。(选择JLinkRDI.dll), 点击OK。(如果没有自动弹出来Choose Target窗口, 请选择Options→Configure Target)

12. 然后在AXD进行调试配置, 选择Options→Configure Interface的选项卡Session file(这里我们注意一下, 在此之前我们应该写好Session file, 见附录) Browse将Session file添加进来。

到这里, 我们会看到ARM开发板的LED灯会如我们所愿的跑起来, 修改代码, 想让她怎么跑就怎么跑。这是我做实验的一个体会心得, 肯定有遗漏之处(我已经尽量细化每一个步骤), 请大家多多包涵。

现在, 我们动手试试吧!!!

附录:

Init.s:

```
AREA INIT2440, CODE, READONLY
ENTRY
ldr sp, =0x34000000
IMPORT xmain
bl xmain
loop
b loop
END
```

Main.c:

```
#define GPBCON (*(volatile unsigned long *)0x56000010)
#define GPBDAT (*(volatile unsigned long *)0x56000014)
#define LEDES (1<<5|1<<6|1<<7|1<<8)
#define DELAYVAL (0xffff)

extern int delay(int time);

int xmain()
{
GPBCON = 0x00015400;
while(1)
{
GPBDAT = (GPBDAT&(~LEDES)) | (1<<6|1<<7|1<<8);
delay(DELAYVAL);
GPBDAT = (GPBDAT&(~LEDES)) | (1<<5|1<<7|1<<8);
delay(DELAYVAL);
GPBDAT = (GPBDAT&(~LEDES)) | (1<<5|1<<6|1<<8);
delay(DELAYVAL);
GPBDAT = (GPBDAT&(~LEDES)) | (1<<5|1<<6|1<<7);
delay(DELAYVAL);
}
```

```
}  
}
```

Delay.s:

```
EXPORT delay  
AREA DELAY, CODE, READONLY  
delay  
sub r0, r0, #1  
cmp r0, #0x0  
bne delay  
mov pc, lr  
END
```

Init2440.exe:

```
Setmem 0x53000000 0x00000000 32  
Setmem 0x4A000008 0xFFFFFFFF 32  
Setmem 0x4A00001C 0x000007FF 32  
Setmem 0x53000000 0x00000000 32  
Setmem 0x56000050 0x000055AA 32  
Setmem 0x4C000014 0x00000007 32  
Setmem 0x4C000000 0x00FFFFFF 32  
Setmem 0x4C000004 0x00061012 32  
Setmem 0x4C000008 0x00040042 32  
Setmem 0x48000000 0x22111120 32  
Setmem 0x48000004 0x00002F50 32  
Setmem 0x48000008 0x00000700 32  
Setmem 0x4800000C 0x00000700 32  
Setmem 0x48000010 0x00000700 32  
Setmem 0x48000014 0x00000700 32  
Setmem 0x48000018 0x0007FFFC 32  
Setmem 0x4800001C 0x00018005 32  
Setmem 0x48000020 0x00018005 32  
Setmem 0x48000024 0x008E0459 32  
Setmem 0x48000028 0x00000032 32  
Setmem 0x4800002C 0x00000030 32  
Setmem 0x48000030 0x00000030 32
```