

主控板 1756-IB32 增重编码器模块

产品名称	主控板 1756-IB32 增重编码器模块
公司名称	厦门盈亦自动化科技有限公司
价格	977.00/件
规格参数	品牌:A-B 型号:1756-IB32 产地:美国
公司地址	厦门市集美区宁海三里10号1506室
联系电话	0592-6372630 18030129916

产品详情

主控板 1756-IB32 增重编码器模块

1756-A10	1756-IF16	1794-IM16	1756-HSC
1756-A13	1756-IF16H	1794-IM8	1756-IA16
1756-A17	1756-IF8	1794-IR8	1756-IA16I
1756-A4	1756-IF8H	1794-IRT8	1756-IA32
1756-A7	1756-IF8I	1794-IT8	1756-IB16
1756-BA1	1756-IF6I	1794-IV16	1756-IB16D
1756-BA2	1756-IF6CIS	1794-IV32	1756-IB16I
1756-BATA	1756-IT6I	1794-OA16	1756-IB32
1756-CN2	1756-IR6I	1756-M03SE	1756-BATA
1756-CN2R	1756-IR12	1756-M08SE	1756-CNB
1756-CNB	1756-IRT8I	1756-M16SE	1756-IC16
1756-CNBR	1756-IT6I2	1756-N2	1756-IB16

1756-DHRIO	1756-IM16	1756-OA16	1756-IB32
1756-DNB	1756-L61	1756-OA16I	1756-IF16
1756-EN2T	1756-L62	1756-OB16D	1756-IR61
1756-EN2TR	1756-L63	1756-OB16E	1734-ACNR
1756-EN3TR	1756-L64	1756-OB16I	1734-ADN
1756-ENBT	1756-L65	1756-OB32	1734-AENT
1756-ENET	1756-L71	1756-OF4	1734-AENTR
1756-EWEB	1756-L71S	1756-OF8	1734-APB
1756-TBS6H	1756-PA75R	1756-OF8I	1746-IA16
1756-TBSH	1756-PB72	1756-OW16I	1746-IB16
1757-SRM	1756-PB75	1756-PA72	1746-IB32
1746-N2	1756-RM	1756-PA75	1746-IM16
1746-NI16I	1756-IB16	1794-OA8	1746-IO12DC
1746-NI4	1746-IV32	1794-OA8I	1746-ITB16

主控板 1756-IB32 增重编码器模块

D9-Pro处理器开发指南

为了让更多客户快速使用D9-Pro核心板，米尔电子在推出开发板的同时就推出了配套的开发资料。

对于零基础的朋友，可能需一点时间熟悉一下相关知识。但对于有一定基础的开发者而言，基于D9平台开发还是相对不难。这里结合官方资料给大家分享一下 D9 Linux 软件开发主要流程：

0、概述MYD-JD9X 搭载基于 Linux 4.14.61 版本内核的操作系统。开发板出厂附带嵌入式 Linux 系统开发所需要的交叉编译工具链，U-boot 源代码，Linux 内核和各驱动模块的源代码等资料包，以及适用于 Windows 桌面环境和PC Linux 系统的各种开发和调试工具，应用开发例程等。

1、开发环境

D9处理器：D9310、D9340、D9350、D9360等。

Linux开发主机：Debian, Ubuntu, RHEL等，Ubuntu18.04 64bit 桌面版系统。

安装米尔定制的 SDK

在使用 SDK 构建这个系统镜像过程中，还需要安装交叉工具链，米尔提供的这个 SDK 中除了包含各种源代码外还提供了必要的交叉工具链，可以直接用于编译应用程序等。SDK 的安装步骤，如下：拷贝 SDK 到 Linux 目录并解压新建工具链目录

```
PC$ sudo mkdir -p /tool
```

解压编译链

```
PC$ cd $HOME/MYD-JD9XPC$ tar -xvf gcc_linaro_7.3.tar.gz -C /tool
```

安装、测试编译链为方便设置编译链，他们提供一个设置脚本 env.sh

2、构建开发板镜像第1步：获取源码购买开发板会提供源码和配套的工具。第2步：快速编译镜像这里我们需要使用米尔提供的 envsetup.sh 脚本进行环境变量的设置。进入 SDK 目录，执行 ./build.sh config 工程配置，选择 board、project、启动方式和内核版本。然后执行 ./build.sh 完成编译和打包，镜像生成到 out 目录。只需执行 build.sh 就可以一键编译镜像：

```
PC$ ./build.sh
```

镜像编译完成之后，在 SDK 源码会生成 out 目录，out 目录是执行 build.sh 一键编译脚本后自动生成的目录，包含所有的镜像组件。主要组件介绍：

images_d9_ref 板子的镜像主目录

binary 目录，包含所有组件的二进制文件

build_object 目录，包含 BSP 以及 buildroot 系统等组件的编译信息

images 目录，最终的镜像目录，.pac 格式的镜像烧录包

第3步：构建 SD 卡烧录器镜像构建 sd 卡烧录镜像之前，需要先构建完 full 镜像。准备 full 镜像，进入 SDK 目录，执行 build.sh config 命令，选择 D9 为 SD 卡启动。然后，编译 SD 卡镜像，编译更新镜像包。

3、烧录系统镜像D9系列微处理器启动方式有多种，用户可以根据需求选择不同的方式进行更新。更新方式主要有以下几种：.

USB 烧写：适用于研发调试，测试等场景。

制作 SD 卡启动器: 适用于研发调试，快速启动等场景

制作 SD 卡烧录器：适用于批量生产烧写 eMMC

官方资料给出了两种方法：SDFactoryTool 烧录、制作 SD 卡烧录。方法一：SDFactoryTool 烧录也就是使用SDFactoryTool工具，通过USB进行烧录。通过USB将电脑和开发板连接，选择启动模式,将拨码开关拨到 USB 下载模式。准备烧录镜像，启动 SDFactoryTool 工具进行下载即可。

方法二：SD卡烧录为满足生产，大批量烧录，也可以使用SD卡烧录的方法。这里通过制作 TF 卡烧录器镜像包实现这种方法，为节省时间，米尔已经把 sd 卡烧录镜像包编译完成，用户只需编译更新镜像包就行。（具体可以参考开发手册）

4、修改板级支持包这一节应该是相对比较重要的，也是相对比较难的，包括U-boot、kernel等相关内容的编译与更新。a.板载 SSDK 编译与更新SemiDrive MCU SDK（以下简称 SSDK）用于 SemiDrive 各芯片平台的 MCU 软件开发，例如 D9 系列 SOC 的 Safety R5 软件。编译 SSDK：

```
wuji@system2-ubuntu1804:/media/wuji/D9/d9_ssd$ ./build.sh ssdk
```

编译完成后，你可以在 source/ssdk/boards/d9_ref/build/ref/gcc/secure/目录找到 safety.bin 和 secure.bin 二进制文件。b.板载 uboot 编译与更新U-boot 是一个功能非常丰富的开源启动引导程序，包括内核引导，下载更新等众多方面。编译 uboot：

```
wuji@system2-ubuntu1804:/media/wuji/D9/d9_ssd$ ./build.sh uboot
```

编译完成后，你可以在 out/images_d9_ref/binary/download_img/目录找到 bootloader.img 二进制文件。更新 uboot：拷贝编译的镜像（bootloader.img）到开发板，uboot 对应的分区为/dev/mmcblk0p26，执行下面的命令更新：

```
root@myd-jd9x:~# echo 0 > /sys/block/mmcblk0boot0/force_ro
root@myd-jd9x:~# echo 0 > /sys/block/mmcblk0boot1/force_ro
root@myd-jd9x:~# dd if=bootloader.img of=/dev/mmcblk0p26 bs=1M conv=sync 0+1 records in1+0 records out
```

c.板载 Kernel 编译与更新编译 linux：

```
wuji@system2-ubuntu1804:/media/wuji/D9/d9_ssd$ ./build.sh linux
```

编译完成后，你可以在 out/images_d9_ref/binary/download_img 目录找到 kernel.img和 dtb.img 二进制文件。更新内核和设备树：拷贝编译的镜像（kernel.img）到开发板，kernel 对应的分区为/dev/mmcblk0p38，dtb 对应的分区为/dev/mmcblk0p34，执行下面的命令更新：

```
root@myd-jd9x:~# echo 0 > /sys/block/mmcblk0boot0/force_ro
root@myd-jd9x:~# echo 0 > /sys/block/mmcblk0boot1/force_ro
root@myd-jd9x:~# dd if=kernel.img of=/dev/mmcblk0p38 bs=1M conv=sync 9+1 records in10+0 records out
root@myd-jd9x:~# dd if=dtb.img of=/dev/mmcblk0p34 bs=1k conv=sync 113+1 records in114+0 records out
```

5、适配硬件平台这一节相当于是芯片底层驱动相关的内容，可通过芯驰提供的引脚配置的工具，快速定义各个节点的管脚，属性，电源等（官方提供了对应工具）。这里配置的具体细节，可以参看《MYD-JD9X_Linux软件开发指南》第6节。接下来，就可以添加你的应用程序了，官方提供了“基于 Makefile 的应用”的介绍，大家可以参看一下。

主控板 1756-IB32 增重编码器模块