

# V9801Test\_Keil\_学习板工程说明

本文档为 SD502 工具包套配学习板的说明文档,主要介绍了如何利用万高科技提供的代码,在 Keil C51 的环境下建立学习板工程 V9801Test,并利用该工程验证 SD502 是否正常工作。

V3.0

2014-09-22

本文档也描述了如何使用 Keil C51 IDE 的编译器/汇编器和链接器/定位器(LX51),对程序进行代码分页 (Code Banking),进而完成大于 64KB 的 V9801 应用程序的编译、调试,并将其烧写入 Flash。

## 1. 准备

1. 按照《SD502 用户手册》的说明按装 SD502 的 USB 驱动和万高科技专用 DLL (Keil 版);
2. 硬件: SD502;
3. 芯片: V9801 (8052 MCU 内核, 128KB Flash, 4KB RAM);
4. 源文件: main.c/delay.c/STARTUP.A51/L51\_BANK.A51/bank1.c/bank2.c/bank3.c;
5. 参考资料: V9801 数据手册、 $\mu$ Vision® IDE User's Guide

## 2. 关于代码分页程序

V9801 的片上 Flash 存储器大小为 128KB, 可用于存储大于 64KB 的应用程序。但是, 8051 内核 MCU 的地址总线宽度为 16-bit, 最多只能寻址 64KB 的程序存储空间。所以, 对于超过 64KB 的 V9801 的应用程序, 用户可采用代码分页 (Code Banking) 技术, 将应用程序分成多个小于或等于 64KB 的代码段, 通过硬件额外增加的逻辑扩展地址线, 并结合软件, 将这些代码段烧写入 Flash 中, 供 MCU 访问。

对应用程序进行代码分页时, 应将代码分成两类: 公共代码和段代码。公共代码为所有段代码所共用, 包括一些所有程序都会调用的函数, 比如: 复位向量 (Reset Vector)、中断向量 (Interrupt Vector) 和中断进程 (Interrupt Routine) 等, 而且为了实现不同段代码的跳转, 那些控制段代码跳转的程序也应归为公共代码。段代码一般是各功能模块相应的代码。段代码编号从 0 开始, 即 Bank0.....Bankn。不同段代码内的程序互相调用, 必须先进行段代码切换, 增加了程序执行的时间, 所以, 为了避免通过切换不同段代码来实现同一个功能, 同一个功能模块的代码一般被归为一个段代码。

8051 的 64KB 的地址空间被划分成两部分, 公共代码区域 (Common Code Area) 和段代码区域 (Bank Code Area)。在地址空间中, 公共代码区域一般从 0x0000 开始, 至段代码区域开始。CPU 执行程序时, 公共代码映射到公共代码区域。根据程序执行的需要, 不同的段代码将轮流映射到段代码区域。在 V9801 中, 片上 8051 的地址空间的低 32KB (0x0000~0x7FFF) 为公共代码区域, 而高 32KB (0x8000~0xFFFF) 为段代码区域。

## 3. 硬件支持

V9801 的片上 8051 额外增加了两条逻辑扩展地址线, 即, SFR 0xA0 (CBANK) 的 bit1 和 bit0。这两个位与地址总线的最高位 (A15) 一起, 以 “与” 逻辑的关系, 将 8051 连接到片上 Flash 地址线的高 2 位, 使 8051 可以访问 128KB 的 Flash 存储器地址范围。

只要 A15 为低电平, 8051 即可访问 Flash 存储器的低 32KB 地址 (0x0000~0x7FFF), 该地址范围用于存储公共代码。



当 A15 为高电平时，根据寄存器 CBANK SFR 的配置值，8051 访问 Flash 存储器的 3 个区域，即，V9801 支持用户将应用程序分成 3 个段代码，分别被存储于 Flash 的这 3 个区域，其中，CBANK SFR 的配置值被译码为段代码的编号，如表 1 所示。用户可通过配置 CBANK SFR 的值进行段代码切换。CBANK SFR 不可被配置为 0x00，所以，在 Keil C51 环境进行工程配置时，不应将程序分配至 Bank0 中。

表 1 当 A15 为高电平时，CBANK SFR (0xA0) 配置与段代码各种地址的关系

CBANK	映射到地址空间 (0x8000~0xFFFF) 的段代码	物理地址	逻辑地址
0x01	Bank1	0x08000~0x0FFFF	0x818000~0x81FFFF
0x02	Bank2	0x10000~0x17FFF	0x828000~0x82FFFF
0x03	Bank3	0x18000~0x1FFFF	0x838000~0x83FFFF

发生 POR、恢复供电复位、IO/RTC 休眠唤醒复位、RSTn 低电平有效或 WDT 溢出复位时，CBANK SFR 被复位为 0x01，即，此时，根据 A15 的电平，8051 可以访问 Flash 存储器的地址范围 0x0000~0xFFFF。

在 Keil IDE 中，逻辑地址的最高字节 (MSB) 表示段代码编号，比如 0x81 表示 Bank1，0x82 表示 Bank2；低 2 个字节表示地址空间中的段代码区域 (0x8000~0xFFFF)。

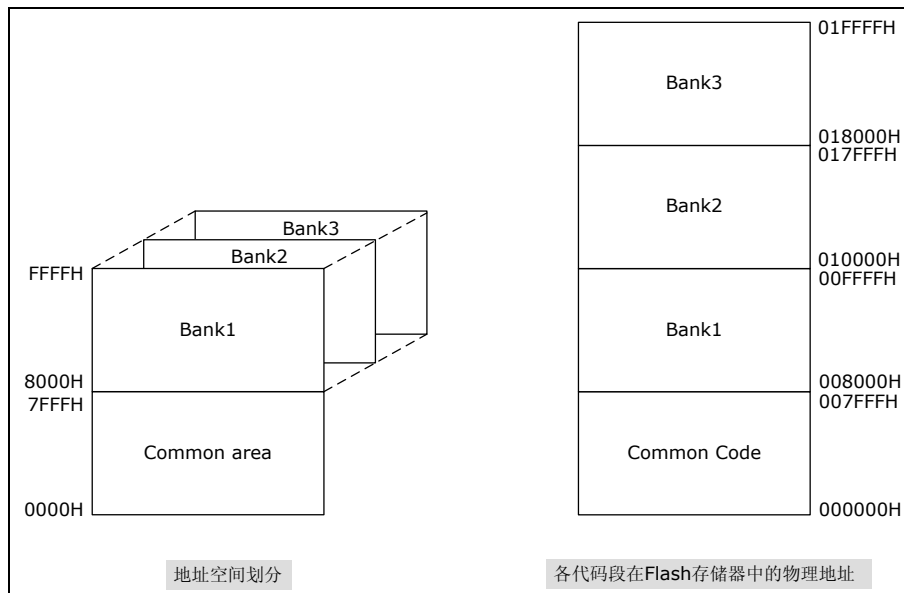


图 1 8051 地址空间划分及各代码段在 Flash 存储器中的物理地址 (V9801)

## 4. 在 Keil C51 IDE 中建立代码分页的应用程序

1. 点击 **Project ->New ->µVision Project**，并将工程保存为 V9801Test；
2. 在 **Select Device for Target 'Target 1'**对话框中，选择 MCU 内核；
3. 选择 **Use Extended Linker (LX51) instead of BL51**，使用链接器/定位器 LX51，以使能生成 **HEX-386** 格式的 HEX 文件选项；

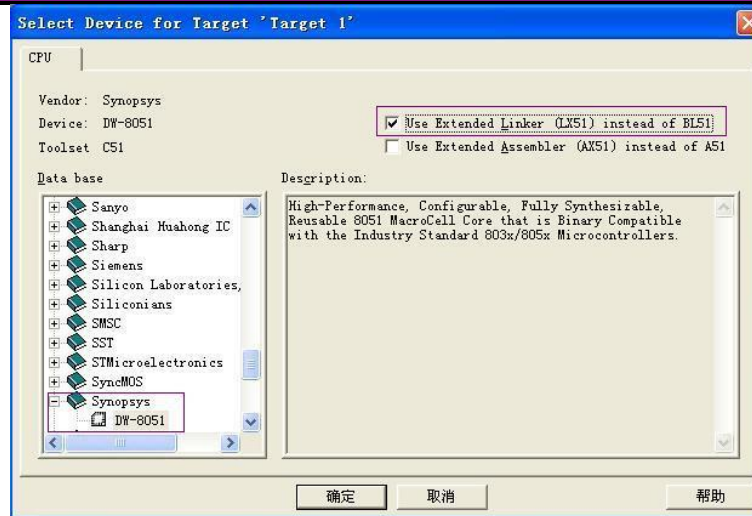


图 2 选择 MCU 内核和链接器/定位器

4. 不需要将标准 **8051** 的启动代码添加到工程中;
5. 双击 **Source Group 1**, 或右键点击 **Source Group 1 -> Add Files to Group 'Source Group 1'**, 添加由万高科技提供的学习板配套源文件:
  - 5.1 **main.c**: 主程序, 必须被归为公共代码;
  - 5.2 **bank1.c/bank2.c/bank3.c**: 段代码;
  - 5.3 **delay.c**: 包括两个延时函数, 分别延时 100ms 和 1ms;
  - 5.4 **STARTUP.A51**: 万高科技提供的启动代码;
  - 5.5 **L51\_BANK.A51**: 万高科技根据 V9801 的硬件机制修改的代码分页配置文件, 设置了段代码的数量和段代码的切换机制等。文件的主要修改如下:

```
$NOCOND
;-----
; This file is part of the BL51 / LX51 Banked Linker/Locater package
; Copyright (c) 1988 - 2001 Keil Elektronik GmbH and Keil Software, Inc.
; Version 2.21 (Code and Variable Banking for Classic 8051 Derivatives)
;-----
;***** Configuration Section modified by Vango*****
?B_NBANKS      EQU 4      ; Define maximum Number of Banks                *
;                                                         *
?B_MODE        EQU 0      ; 0 for Bank-Switching via 8051 Port            *
;                                                         *
?B_RTX         EQU 0      ; 0 for applications without real-time OS        *
;                                                         *
?B_VAR_BANKING EQU 0      ; 0 Variable Banking does not use L51_BANK.A51    *
;                                                         *
?B_RST_BANK    EQU 0xFF   ; specifies the active code bank number after CPU      *
;                                                         *
;                                                         *
;                                                         *
; INTERBANK CALL TABLE. The value 0xFF disables          *
; this LX51 linker/locater optimization.                  *
;-----*
;                                                         *
IF ?B_MODE = 0;
;-----*
; if ?BANK?MODE is 0 define the following values          *
```

```

; For Bank-Switching via 8051 Port define Port Address / Bits      *
;                                                                *
P1          DATA    0A0H      ; I/O Port Address                *
;                                                                *
?B_PORT     EQU      P1       ; default is P1                    *
?B_FIRSTBIT EQU      0        ; bit0 is the first address line    *
;-----*
ENDIF;
;                                                                *
;                                                                *

```

6. 选择 **Target 1**，并点击 **Project ->Options for Target 'Target 1'**，或者，双击 **Target 1 ->Options for Target 'Target 1'**，打开 **Options for Target 'Target 1'**对话框，进行 **Target** 选项卡设置。

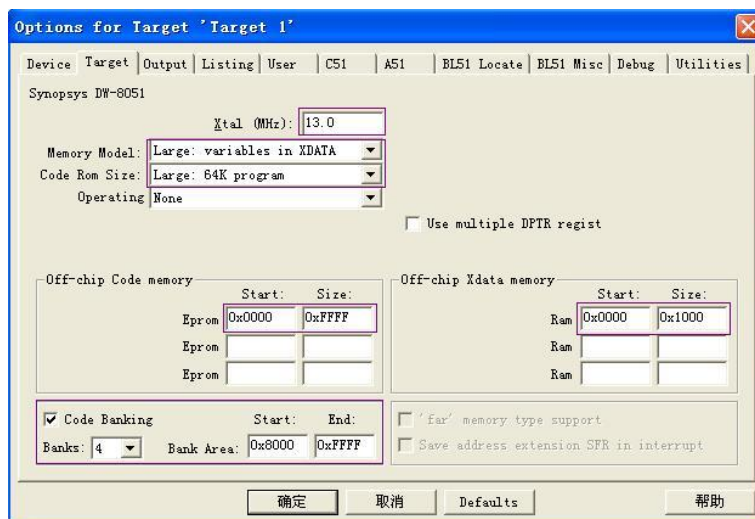


图 3 Target 选项卡设置

7. 在 **Project Workspace** 窗口，右键分别点击 **bank1.c/bank2.c/bank3.c/delay.c/main.c /STARTUP.A51/L51\_BANK.A51 ->Options for File 'xx.x'**，在打开的文件配置对话框中，在 **Properties** 选项卡下，将各源文件分配到相应的段代码和公共代码中。其中，**STARTUP.A51** 和 **L51\_BANK.A51** 必须归为公共代码。

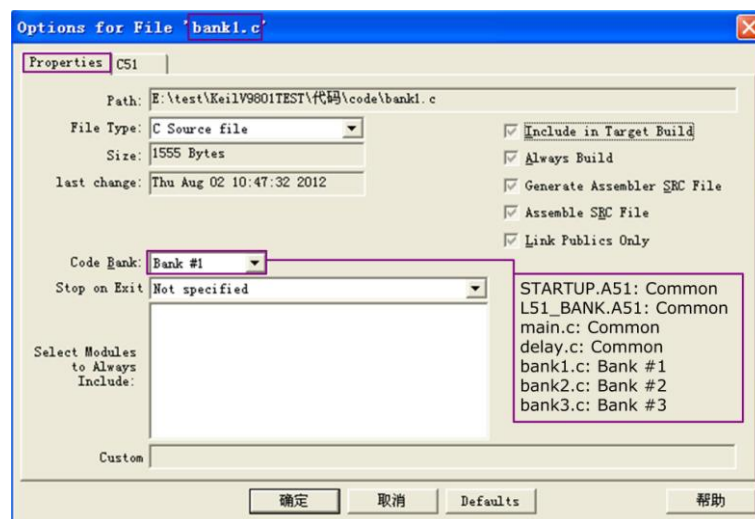


图 4 源文件配置

8. 选择 **Target 1**，并点击 **Project ->Options for Target 'Target 1'**，或者，双击 **Target 1 ->Options for Target 'Target 1'**，进行工程设置：

8.1 在 **Output** 选项卡上设置文件输出；

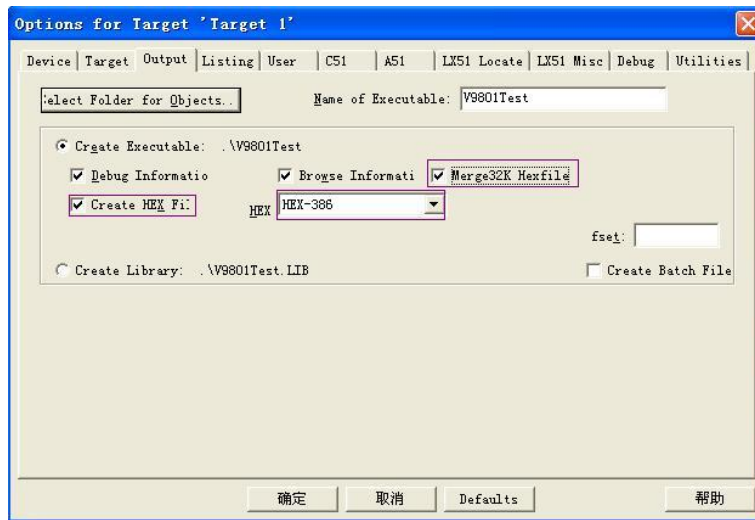


图 5 选择生成 HEX-386 格式的 HEX 文件，并选择 Merge32K Hexfile

如果使用链接器/定位器 BL51，那么，在 **Output** 选项卡上，只能选择生成 HEX-80 格式的 HEX 文件，此时，每个代码段生成一个扩展名为.h0X 的 HEX 文件（X 为各段代码编号），包括了公共代码和相应段代码的信息。比如，扩展名为.h01 的 HEX 文件，在该文件中，0x0000~0x7FFF 为公共代码的信息，而 0x8000~0xFFFF 为 Bank1 的信息。将.h01/.h02/.h03 这三个文件烧写入目标板 Flash 时：

.h01 文件应烧写入 0x010000~0x01FFFF；.h02 文件应烧写入 0x020000~0x02FFFF；.h03 文件应烧写入 0x030000~0x03FFFF。上述地址已经超出了 V9801 片上 Flash 的实际地址范围，所以不便于使用。

为了解决上述问题，我们应使用 Keil 提供的扩展链接器/定位器 LX51，并在 **Output** 选项卡中，选择生成 **HEX-386** 格式的 HEX 文件，并选择 **Merge32K hexfile**，从而生成一个包括了 32KB 公共代码和所有段代码（不包括 Bank0）的.hex 文件，其组成如下：

0x0000~0x7FFF: Common code

0x8000~0xFFFF: Bank1

0x10000~0x17FFF: Bank2

0x18000~0x1FFFF: Bank3

这个 HEX 文件可直接写入 V9801 的片上 128KB Flash 中。

HEX-386 与 HEX-80 文件类似，只是它扩展了相应的线性地址，地址段可以大于 2 个字节，从而满足了寻址大于 64KB 程序的需要。

- 8.2 在 **Debug** 选项卡上，选择 **Use**，并在该下拉列表中选择 **SD502 配套 dll**，之后，点击 **Settings**，在 **Vango 芯片调试接口设置**对话框中勾选 **Code Banking**。在 **Debug** 选项卡上，取消 **Breakpoints**，并点击**确定**完成 **Debug** 的设置；

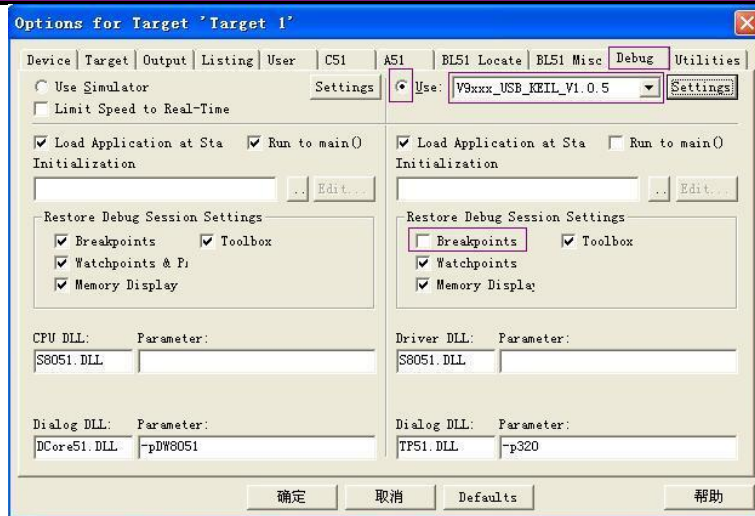


图 6 设置 Debug 选项卡

取消选择 **Breakpoints**，使得每次进入 Debug 调试时，所有断点均会被清除，可以有效防止在进入调试时出现断点过多的错误。如果仍然勾选了该项目，则必须保证工程文件单步和全速运行时断点不超过 3 个。



图 7 Vango 芯片调试接口设置


如果创建的 V9801 的应用程序不需要进行代码分页，那么，在 **Vango 芯片调试接口设置**对话框中，不需要选择 **Code Banking** 选项。

- 8.3 在 **Utilities** 选项卡上，选择 **Use Target Driver for Flash Programming**，在其下拉框中选择 **SD502 配套 dll**，并点击 **Settings** 进行 Flash 烧写设置，并点击**确定**，从而完成 **Utilities** 的设置。





图 8 Utilities 设置

9. 点击工具栏中的  按钮，编译和链接程序。下方 **Output Window -> Build** 窗口将显示编译和链接过程。如果最后显示 **0 Error(s), 0 Warning(s)**，则表明一个代码分页的 V9801Test 工程已经成功建立，并输出一个 HEX 文件（V9801Test.hex）。

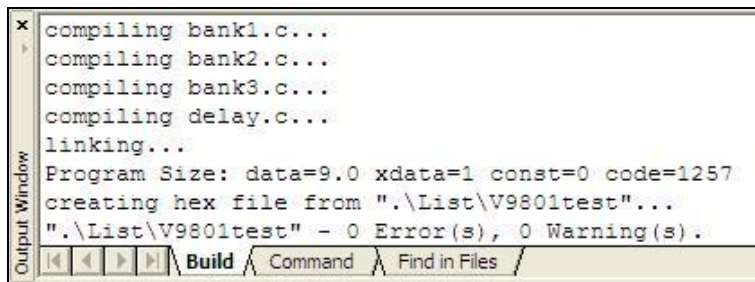


图 9 Output Window – 代码编译和链接

## 5. 烧写 Flash

正确连接 PC、SD502 和学习板后，点击工具栏中的  按钮，将上述编译通过的代码烧写入学习板中。烧写过程中，**Output Window** 窗口将显示烧写过程与结果。

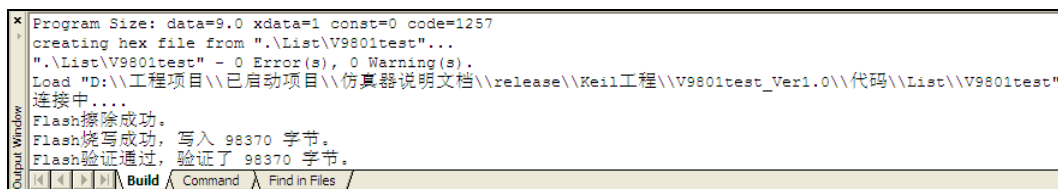


图 10 Output Window 显示烧写过程与结果

## 6. 验证

程序烧写完成后，复位学习板，并观察学习板上的 LED 指示灯。如果观察到如图 11 所示的现象，表明程序下载成功。

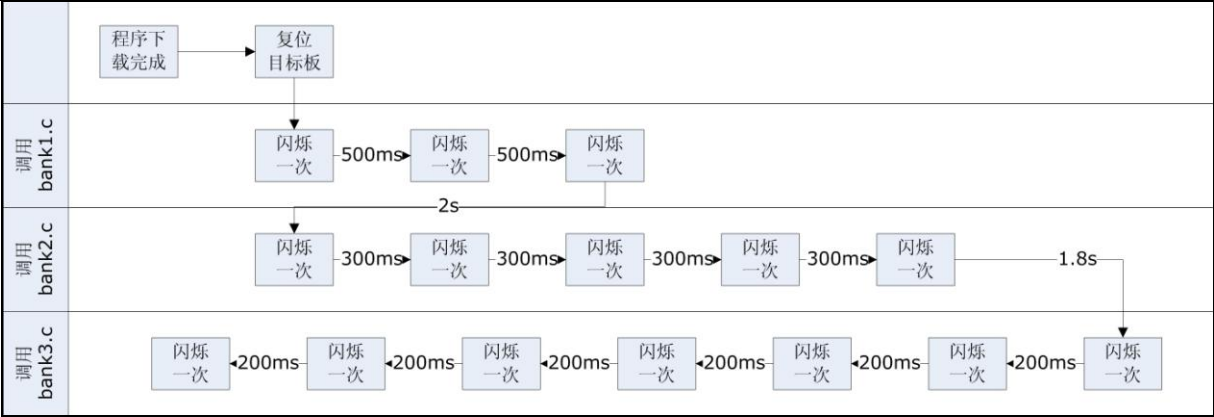



图 11 验证现象

7. 注意

1. 在调试过程中，当遇见函数 `SetPLL(SETPLL_6_4M);`时，不可点击  按钮，或按 **F11** 键进入此函数调试。因为在切换 PLL 过程中，芯片速度会下降，此时若使用单步调试会失去调试连接。
2. 如果创建一个不需要进行代码分页的 V9801 应用程序，请参照 **V9811Test\_Keil\_学习板工程说明.pdf** 进行设置。



## 版本修订历史

日期	版本	说明
2014-09-22	V3.0	仿真器改为 SD502
2012-08-28	V2.0	文档结构/格式修改 修改关于工程设置的描述
2012-06-29	V1.0	正式发布

杭州万高科技有限公司保留对本手册所涉及的产品及相关的技术信息进行补正或更新的权利。使用本手册时，请您从我们的销售渠道或登录公司网站 <http://www.vangotech.com> 获取最新信息。