

LGT8P653A 开发指南

概要

文档主要用于介绍如何搭建 LGT8P653A 的开发平台，并同时介绍开发必须的工具/硬件的使用方法。

硬件准备：

LGT8P653A 编程板(提供 OTP 编程所需高压)
SWDICE mkII 调试器(更新到最新固件)

软件准备：

LGTMix_ISP v2.5 – 在线烧写工具
MIC8S_ASM – MIC8S 汇编器(可选)

XIC2MIC – 代码转换工具
MPLAB IDE v8.92 – 开发环境
HI-TECH PICC v9.83 – PIC 内核 C 编译器

以上所需软件，可以通过 LGT 官网或其他网络途径获取；所需硬件可以通过 LGT 官方淘宝商店购买。



MIC8S 内核系列

LGT8P653A

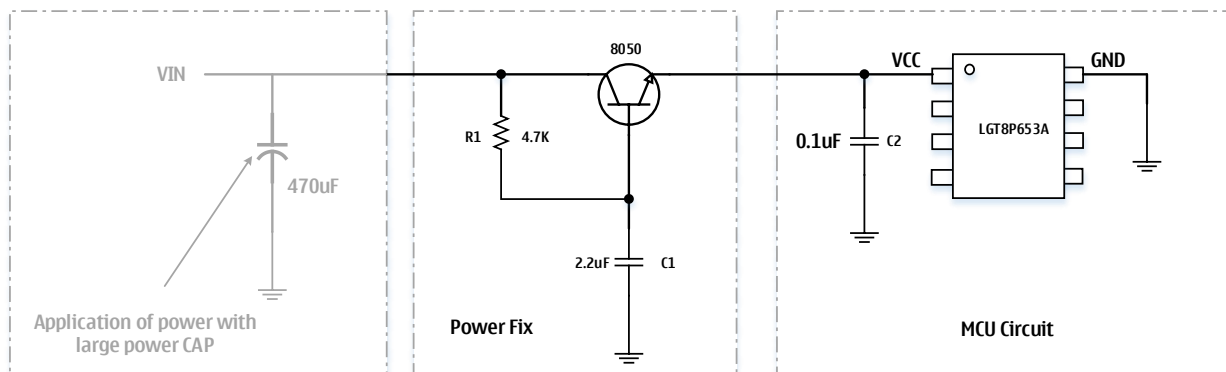
开发指南

Version 1.0

Date 2015/06/24

电源问题

LGT8P653A 目前芯片对电源上电有一定的要求， 请设计芯片电源时， 参考以下电源方案：



(注：此问题只针对 LGT8P653A，其他类型或后续版本请忽略)

软件平台

MIC8S 内核指令兼容 PIC14 基本指令集(35 条), 因此可以使用支持 PIC14 的开发环境为 MIC8S 内核的处理器开发软件。但 MIC8S 与 PIC14 采用完全不同指令编码格式, 因此需要做相应的转码。同时 MIC8S 实现了 11 条扩展指令, 如果要使用 MIC8S 的专有扩展指令, 需要使用 MIC8S 的专用汇编器编译。

LGT8P653A 支持两种运行模式: 8P53A 模式与 8P609A 模式。其中 8P609A 模式为默认模式, 可以访问到 LGT8P653A 内部所有资源。8P609 模式下, LGT8653A 的资源配置与 PIC12F609 兼容。因此建议在使用 LGT8P653A 开发项目时, 使用 8P609A 模式, 简化软件开发调试流程。

基于 MIC8S_ASM 的汇编开发

MIC8S_ASM 为 MIC8S 内核专用汇编器。可以使用 MIC8S_ASM 完成基于汇编的软件开发工作。

MIC8S_ASM 支持 MIC8S 内核所有指令, 包括 11 条扩展指令。MIC8S_ASM 直接生成可以在 MIC8S 内核处理器上运行的代码(HEX), 无需转码。

MIC8S_ASM 的使用方法以及 MIC8S 汇编指令定义, 请参考 MIC8S 使用文档;

MIC8S_ASM 为开源项目, 文档和代码可以在如下地址获得:

<https://github.com/LGTMCU/MIC8S-Assembly>

基于 MPLAB IDE + PICC 的 ASM/C 开发

MPLAB IDE 为 PIC 系列单片机开发环境, 可以实现项目管理, 代码编写, 仿真调试以及在线下载等功能。MPLAB IDE 本身只集成了 MPASM 汇编器, 可以实现基于汇编语言的程序开发。MPLAB IDE 对于 C 语言的支持需要专用的 C 语言编译器。这里我们介绍比较常用的 HI-TECH PICC 编译器。

基于 MPLAB + PICC 可以完全按照 PIC 单片机的开发流程。但生成的目标代码无法直接在 MIC8S 内核的单片机上运行, 需要进行转码。XIC2MIC 用于完成转码的工作。

通过对 MPLAB IDE 工程环境的设置, 可以让 MPLAB IDE 利用 XIC2MIC 直接生成 MIC8S 目标代码, 我们将通过下面的例子介绍具体的设置方法。

在进行如下的操作前, 请确保系统中已经安装 MPLAB IDE, HI-TECH PICC 以及 XIC2MIC 工具。以上工具可以直接从如下地址获得:

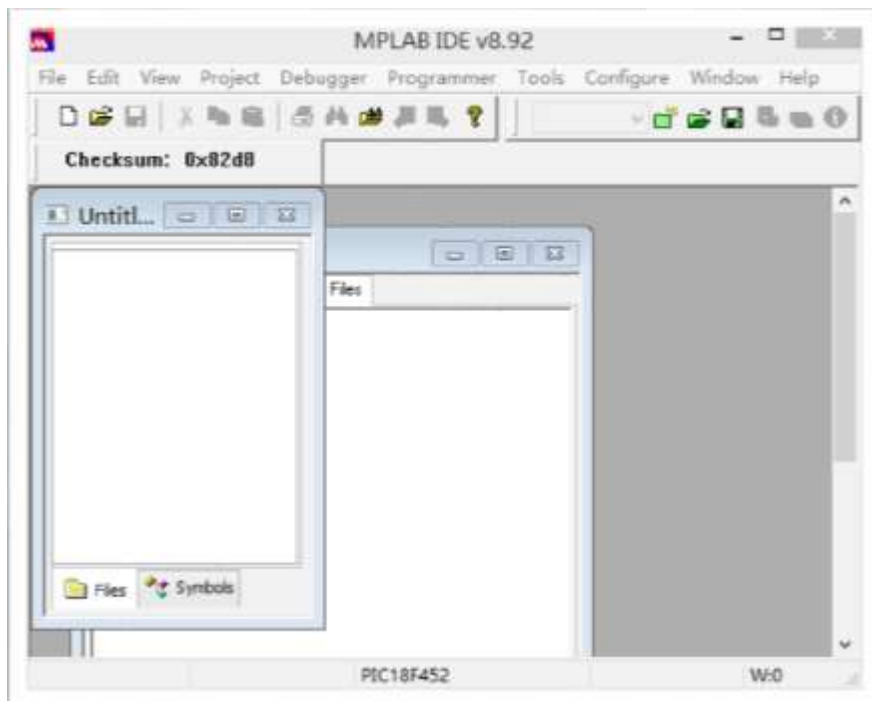
HI-TECH PICC V9.83: <http://pan.baidu.com/s/1c0tYOiW>

MPLAB IDE V8.92: <http://pan.baidu.com/s/1ntqYEEH>

XIC2MIC: <https://github.com/LGTMCU/MIC8S-XIC2MIC>

基于 MPLAB IDE 的汇编开发流程

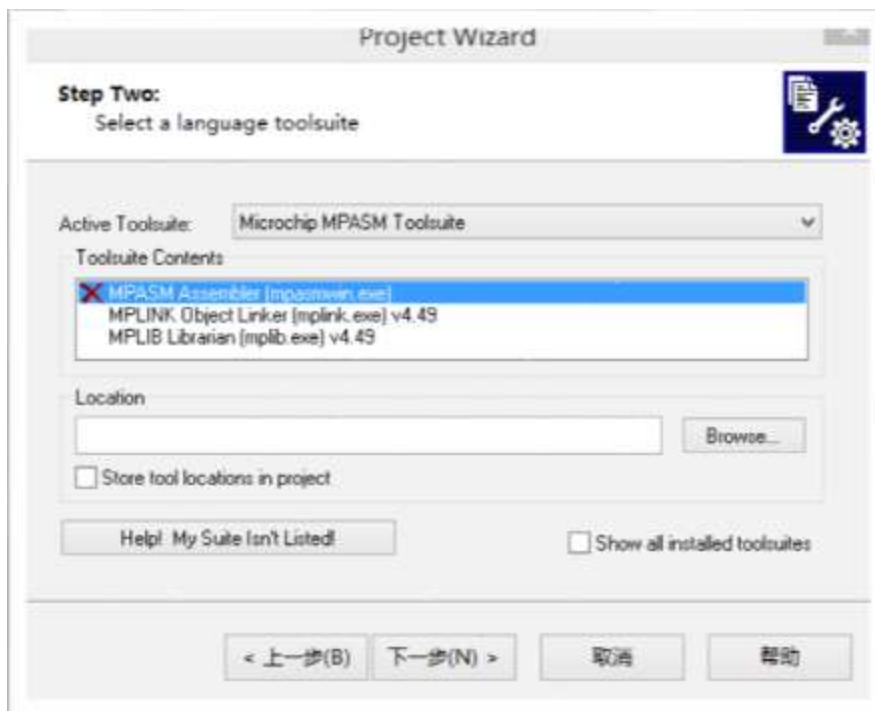
1. 启动 MPLAB IDE



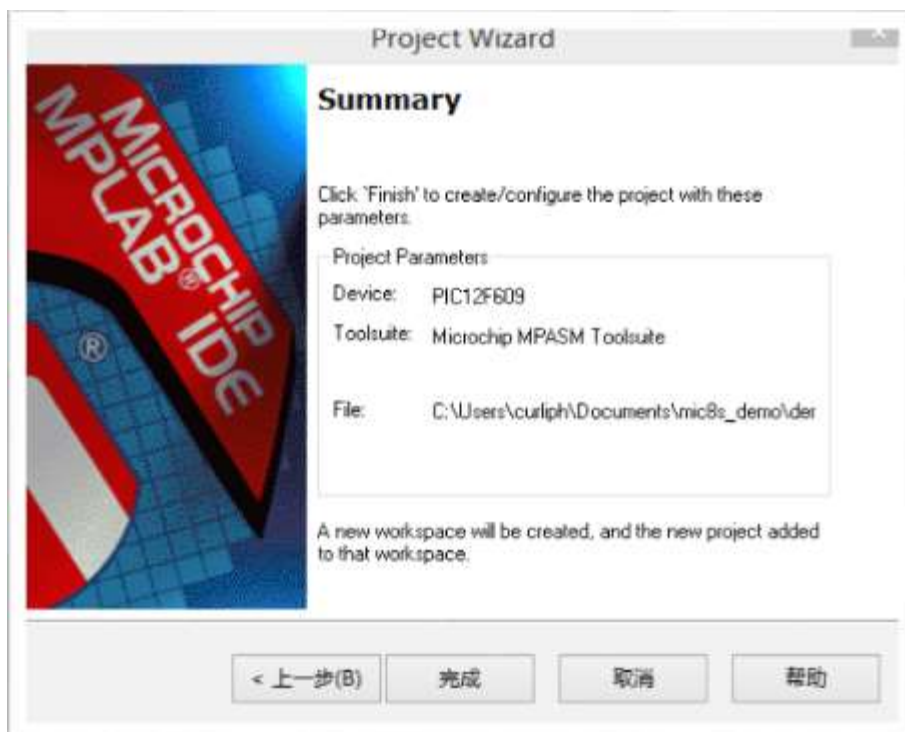
2. 主菜单中选择[Project] -> [Project Wizard...]启动向导，并在设备选择界面选择我们的目标芯片为 PIC12F609:



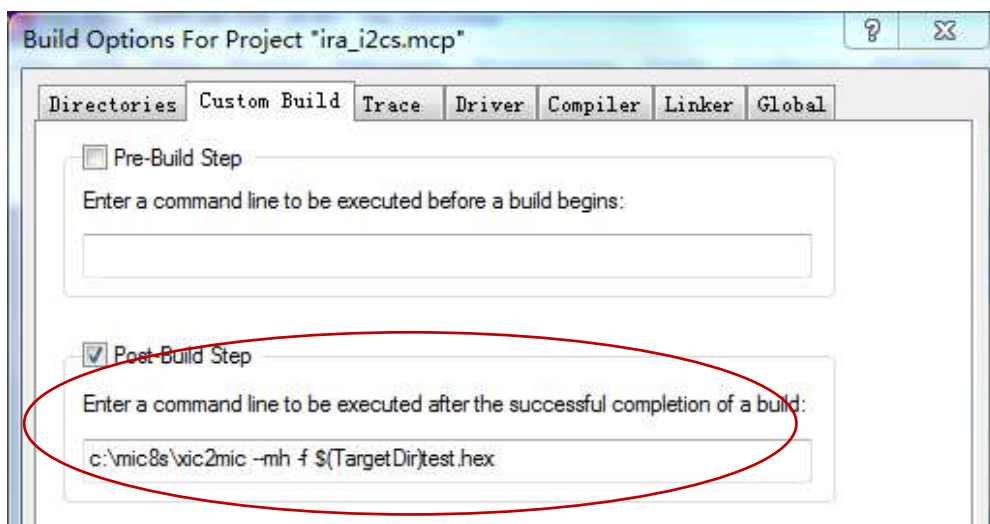
3. 在语言工具窗口选择 MPASM 汇编器作为工具链：



4. 接下来设置工程名称以及项目存放的目录，最后点击[完成]：



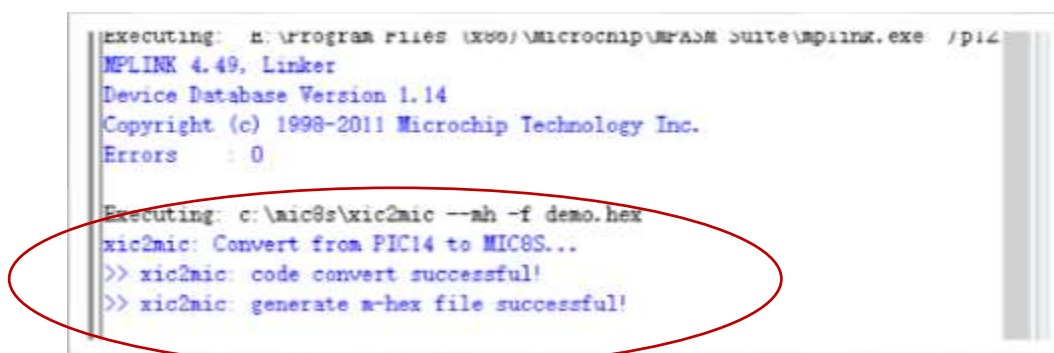
5. 完成后, MPLAB IDE 进入项目管理界面。我们可以在这里添加源文件, 编写代码, 编译以及仿真调试等操作。为了能够让 MPLAB 直接产生 MIC8S 的运行代码, 我们只需要进行如下的设置。
- 在项目树窗口中右击项目, 或在主菜单[Project]下打开[Build Options...], 然后在弹出的设置窗口中选择[Custom Build]标签项, 勾选上[Post-Build Step], 然后填写上需要执行的命令:



将 XIC2MIC 工具的路径和项目名称分别替换为实际有效值即可。

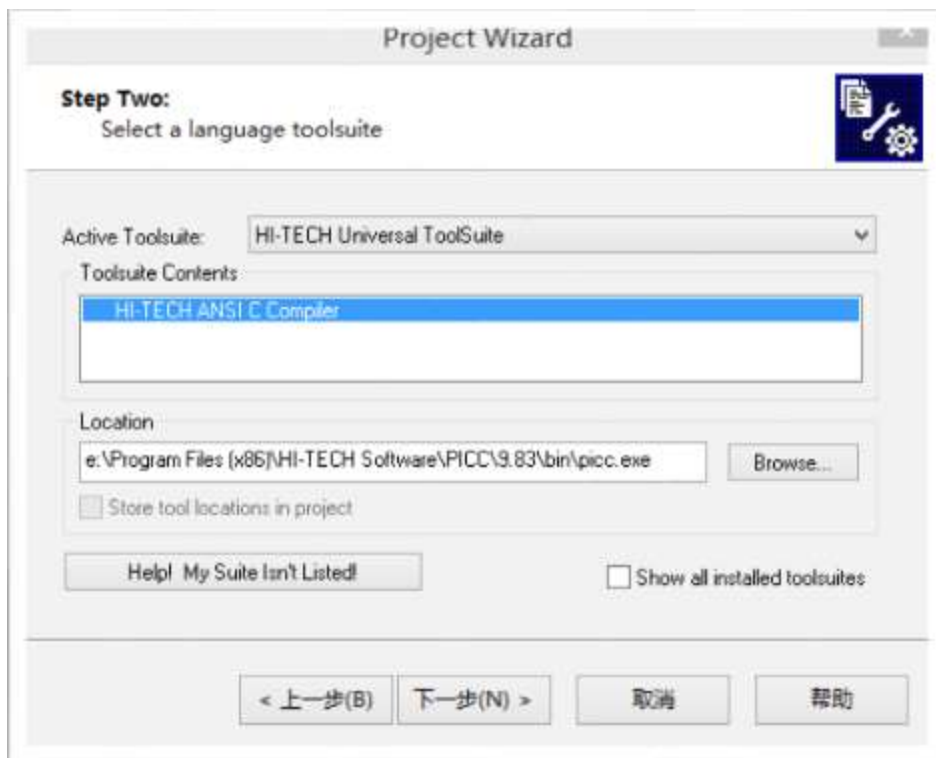
设置完成后, 重新编译工程, 可以在日志输出窗口看到 XIC2MIC 的执行信息, XIC2MIC 工具会在项目目录下生成一个 xxxx_mic.hex 文件, 这个文件可直接烧写到 MIC8S 系列单片机中运行。

【重要提醒】 [Post-Build Step]设置中, 需要将“demo.hex”替换为项目中主汇编文件的名称



基于 MPLAB IDE 的 C 语言开发流程

与 MPLAB IDE 上的汇编语言开发流程类似，我们可以使用[Project Wizard...]向导完成项目的创建。整个创建过程与汇编流程非常相似，不同之处在于对语言工具链的选择：



其他流程包括对 XIC2MIC 的设置与之前的介绍完全一致。

编程烧录

有了可以在 MIC8S 上运行的 HEX 文件，我们就可以将代码烧写到芯片中运行了。LGT8P653A 需要使用专用的烧写软件以及硬件实现。

目前 LGT8P653A 支持三种烧写方式：

1. LGTMCU 通用的调试/烧写方式：LGTMix_ISP 烧写工具 + SWDICE mkII 调试/下载器
2. MIC8S 专用的开源烧写方式：MIC8S_Programmer + 开源下载器硬件
3. 脱机量产烧写方式：LGT 脱机烧写板

以上三种方式，都需要一个专用升压板，为 OTP 烧写提供一个 8.5V 的高压。一般可将这个升压板与芯片的卡座设计为一体。

下面将分别介绍前两种烧写方式。如需脱机量产板可与我们直接联系。

LGTMix_ISP + SWDICE mkII

LGTMix_ISP 为 LGT/MCU 通用在线烧写器工具；最新版本加入了对 LGT8P653A 的支持。可以通过 LGT 官方网站上下载到支持 LGT8P653A 的最新版本，也可以直接与我们联系获得。

SWDICE mkII 为 LGT/MCU 通用调试/下载器。用户可通过我们的官方淘宝店购买，或直接与我们联系获得。购买地址：[SWDICE mkII 调试器购买地址](#)

将所需硬件按下图所示链接，即可实现对 LGT8P653A 的烧写。



MIC8S_Programmer 开源烧写器

为方便广大电子爱好者制作自己的烧写器，我们提供一个开源的烧写器实现。

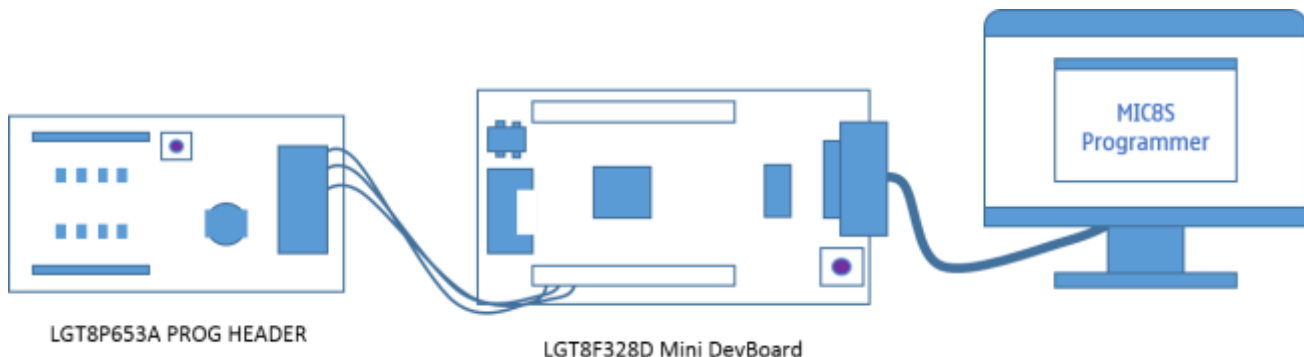
开源烧写器实现一个 MIC8S 专用的在线烧写工具：MIC8S_Programmer

MIC8S_Programmer 与烧写器硬件通过串口通信。烧写器硬件基于 LGT8F328D 最小开发板实现，烧写器的源代码开源，用户可以从中获得 LGT8P653A 的底层烧写通信接口，非常易于移植到其他构架的 MCU 中。

MIC8S 开源烧写器软件以及固件源码，可以从下面的地址获得：

<https://github.com/LGTMCU/MIC8S-Programmer>

将所需硬件按下图所示链接，即可实现对 LGT8P653A 的烧写。



烧写器硬件的固件代码以及烧写器与 LGT8P653A 编程卡座板之间的链接，请参考开源项目中相关文档的介绍。

LGT8P653A 编程注意事项

1. OTP 编程除了需要提供一个 8.5V 左右的 VPP，还需要保证此时芯片的电源 VCC 电压在 4.5~5.5V 左右。VCC 电压偏低容易导致编程数据不稳定；
2. LGT8P653A 使用 GP0/1 作为 SWD/SWC 通讯接口。一般情况下 OTP 为一次可编程，但如果希望在调试过程中通过分区的方法重复利用，需要尽量避免使用 GP0/1 接口，如果需要使用，可将 GP0/1 作为输入功能的数字接口使用。如果需要将 GP0/1 作为输出，或者作为模拟 I/O 使用，当软件运行后，SWD 接口功能无效。这种情况下，可以通过将芯片复位的方法阻止软件的运行。
3. 芯片的外部复位引脚 GP3 与编程高压引脚 VPP 复用，在编程时需要在 VPP 上提供一个 8.5V 的高压，因此复位电路不可以将 GP3 脚直接接地。GP3 作为复位输入时，内部有一个 60K 左右的上拉，需要将 GP3 通过一个合适的电阻(1KΩ)下拉的方式复位。

附录：LGT8P653A 编程底座(PROG_HEADER)原理图

