**各系列MCU芯片基于的内核，不同的内核有不同的汇编**

# 一．MCU芯片编译开发环境及烧录工具

## 1.stc系列

**STC和STC32都是STC（宏晶科技）公司推出的单片机系列**，但它们有显著的不同。STC系列主要基于8051架构，而STC32系列基于ARM Cortex-M0/M3内核。

### STC系列

**开发编译环境：**

STC系列单片机是一种广泛应用的国产单片机，支持丰富的外围接口，适用于各种嵌入式应用。开发STC系列单片机可以使用多种编译环境和工具。以下是常用的开发和编译环境：

#### 1. Keil uVision(免费版与付费版)

**Keil uVision** 是一个强大的集成开发环境（IDE），广泛用于51系列单片机的开发。它支持STC单片机，并且具有强大的调试功能。

##### 安装和配置步骤：

1. **下载和安装Keil uVision**：
   * 访问 Keil官网 下载 Keil uVision。
   * 安装 Keil uVision，按照安装向导进行操作。
2. **安装STC官方提供的开发包**：
   * 访问 STC官方开发工具页面 下载STC-ISP（用于烧录程序）和STC官方提供的库文件和头文件。
   * 解压下载的文件，将库文件和头文件添加到Keil的工程目录中。
3. **创建Keil工程**：
   * 打开 Keil uVision，选择 Project -> New uVision Project 创建一个新工程。
   * 选择目标芯片型号，例如 STC89C52RC。
   * 添加标准51系列单片机的启动文件 STARTUP.A51，可以从安装目录中找到并添加到工程中。
4. **编写代码**：
   * 编写你的C语言或者汇编代码。
   * 添加需要的库文件和头文件，例如 reg51.h。
5. **编译和调试**：
   * 点击 Project -> Build Target 进行编译。
   * 如果使用的是带仿真器的版本，可以进行在线调试。
   * 编译生成的 .hex 文件用于烧录到单片机中。

#### 2. SDCC (Small Device C Compiler)（开源）

**SDCC** 是一个开源的 C 编译器，支持多种 8 位单片机，包括 STC 系列的 8051 单片机。SDCC 可以与其他编辑器如 VSCode 结合使用，形成一个轻量级的开发环境。

官方地址：<https://sdcc.sourceforge.net/>

##### 安装和配置步骤：

1. **下载和安装SDCC**：
   * 访问 SDCC官网 下载并安装 SDCC。
2. **配置环境变量**：
   * 将 SDCC 的安装目录添加到系统的环境变量 PATH 中。
3. **创建工程**：
   * 使用任意文本编辑器（如 VSCode）编写你的 C 代码。
   * 保存文件为 .c 扩展名。
4. **编译代码**：
   * 在命令行中导航到代码所在目录。
   * 使用 sdcc 命令编译代码，例如：

bash

复制代码

sdcc -mmcs51 -o output.ihx your\_code.c

* + 编译生成的 .ihx 文件可以通过转换工具（如 srecord）转换为 .hex 文件，也可以直接使用支持 .ihx 文件的烧录工具烧录到单片机中。

#### 3. Arduino IDE(免费)

**Arduino IDE** 是一种简化的开发环境，支持多种单片机，包括一些STC系列单片机。

##### 安装和配置步骤：

1. **下载和安装Arduino IDE**：
   * 访问 Arduino官网 下载并安装 Arduino IDE。
2. **安装STC支持包**：
   * 安装 Arduino IDE 后，打开软件。
   * 在菜单中选择 File -> Preferences，在 Additional Boards Manager URLs 中添加 STC 的支持包 URL。
   * 访问 [STC-duino](https://github.com/majbthrd/STC-duino) 项目页面，获取具体的支持包 URL。
3. **编写代码**：
   * 使用 Arduino IDE 编写代码，类似于编写 Arduino 程序。
4. **编译和上传**：
   * 选择正确的开发板和端口，点击 Upload 将代码编译并上传到 STC 单片机。

#### 4. PlatformIO（免费）

**PlatformIO** 是一个跨平台的开发环境，支持多种嵌入式开发板和框架，包括STC单片机。

##### 安装和配置步骤：

1. **安装VSCode**：
   * 访问 [VSCode官网](https://code.visualstudio.com/) 下载并安装 VSCode。
2. **安装PlatformIO插件**：
   * 打开 VSCode，进入扩展市场，搜索并安装 PlatformIO 插件。
3. **创建工程**：
   * 使用 PlatformIO 的项目向导创建一个新的项目，选择适当的开发板和框架。
   * 编写你的代码，PlatformIO 将自动处理依赖和编译配置。
4. **编译和上传**：
   * 使用 PlatformIO 提供的编译和上传功能，将编译好的程序烧录到单片机中。

#### 5. STC-ISP 烧录工具(官方提供)

**STC-ISP** 是STC官方提供的用于将编译好的程序烧录到单片机中的工具。

##### 安装和使用步骤：

1. **下载STC-ISP工具**：
   * 访问 STC官方开发工具页面 下载 STC-ISP 工具。
2. **连接单片机**：
   * 使用串口线（USB转TTL线）将电脑和STC单片机连接。
   * 确保电源连接正确，单片机上电。
3. **烧录程序**：
   * 打开 STC-ISP 工具。
   * 选择相应的串口号和波特率。
   * 选择编译生成的 .hex 文件。
   * 点击下载按钮开始烧录程序到单片机中。

#### 总结

STC系列单片机的开发和编译环境多种多样，可以根据具体需求选择合适的工具。Keil uVision 是功能强大的传统选择，适合复杂项目；SDCC 和 PlatformIO 是轻量级且灵活的选择，适合开源和跨平台开发；Arduino IDE 适合快速原型开发。选择合适的工具可以大大提高开发效率和代码质量。

烧录软件

**STC-ISP**

烧录硬件

### STC32系列

开发编译环境

STC32系列单片机是STC公司推出的一系列基于ARM Cortex-M0/M3内核的高性能单片机，广泛应用于各类嵌入式系统中。开发STC32系列单片机可以使用多种开发和编译环境。以下是一些常见的开发环境和工具：

#### 1. Keil MDK (Microcontroller Development Kit)（免费版与付费版）

**Keil MDK** 是一个功能强大的集成开发环境（IDE），支持Cortex-M系列单片机，包括STC32系列。

##### 安装和配置步骤：

1. **下载和安装Keil MDK**：
   * 访问 Keil官网 下载 Keil MDK 安装包。
   * 按照安装向导完成安装。
2. **配置STC32芯片支持包**：
   * 下载STC32的设备支持包，并将其安装到Keil中。具体方法可以参考STC官方文档。
3. **创建工程**：
   * 打开 Keil uVision，选择 Project -> New uVision Project 创建一个新工程。
   * 选择目标芯片型号，例如 STC32G12K128。
4. **编写代码**：
   * 编写你的C代码，Keil提供了丰富的库和头文件支持。
5. **编译和调试**：
   * 点击 Project -> Build Target 进行编译。
   * 使用调试器进行在线调试。

#### 2. IAR Embedded Workbench(收费)

**IAR Embedded Workbench** 是另一个强大的集成开发环境，广泛用于嵌入式系统开发，支持STC32系列单片机。

##### 安装和配置步骤：

1. **下载和安装IAR Embedded Workbench**：
   * 访问 [IAR官网](https://www.iar.com/) 下载 IAR Embedded Workbench 安装包。
   * 按照安装向导完成安装。
2. **配置STC32芯片支持包**：
   * 下载并安装STC32的设备支持包。
3. **创建工程**：
   * 打开 IAR Embedded Workbench，选择 Project -> Create New Project 创建一个新工程。
   * 选择目标芯片型号，例如 STC32G12K128。
4. **编写代码**：
   * 编写你的C代码，IAR提供了丰富的库和头文件支持。
5. **编译和调试**：
   * 点击 Project -> Build All 进行编译。
   * 使用调试器进行在线调试。

#### 3. STM32CubeIDE(免费)

**STM32CubeIDE** 是STMicroelectronics提供的免费IDE，基于Eclipse，支持Cortex-M系列单片机。虽然主要用于STM32系列，但也可以配置支持STC32系列。

##### 安装和配置步骤：

1. **下载和安装STM32CubeIDE**：
   * 访问 STM32CubeIDE官网 下载并安装 STM32CubeIDE。
2. **配置STC32芯片支持包**：
   * 手动添加STC32芯片的启动文件和库文件到工程中。
3. **创建工程**：
   * 打开 STM32CubeIDE，选择 File -> New -> STM32 Project 创建一个新工程。
   * 手动配置目标芯片型号和相关设置。
4. **编写代码**：
   * 编写你的C代码，添加必要的库和头文件。
5. **编译和调试**：
   * 点击 Project -> Build Project 进行编译。
   * 使用调试器进行在线调试。

#### 4. PlatformIO（开源）

**PlatformIO** 是一个跨平台的开发环境，支持多种嵌入式开发板和框架，包括STC32单片机。

##### 安装和配置步骤：

1. **安装VSCode**：
   * 访问 [VSCode官网](https://code.visualstudio.com/) 下载并安装 VSCode。
2. **安装PlatformIO插件**：
   * 打开 VSCode，进入扩展市场，搜索并安装 PlatformIO 插件。
3. **创建工程**：
   * 使用 PlatformIO 的项目向导创建一个新的项目，选择适当的开发板和框架。
   * 编写你的代码，PlatformIO 将自动处理依赖和编译配置。
4. **编译和上传**：
   * 使用 PlatformIO 提供的编译和上传功能，将编译好的程序烧录到单片机中。

#### STC-ISP 烧录工具（官方提供）

**STC-ISP** 是STC官方提供的用于将编译好的程序烧录到单片机中的工具。

##### 安装和使用步骤：

1. **下载STC-ISP工具**：
   * 访问 STC官方开发工具页面 下载 STC-ISP 工具。
2. **连接单片机**：
   * 使用串口线（USB转TTL线）将电脑和STC单片机连接。
   * 确保电源连接正确，单片机上电。
3. **烧录程序**：
   * 打开 STC-ISP 工具。
   * 选择相应的串口号和波特率。
   * 选择编译生成的 .hex 文件。
   * 点击下载按钮开始烧录程序到单片机中。

#### **GNU ARM Embedded Toolchain（补充）**

#### ………

#### 总结

STC32系列单片机的开发和编译环境多种多样，可以根据具体需求选择合适的工具。Keil MDK 和 IAR Embedded Workbench 是功能强大的传统选择，适合复杂项目；STM32CubeIDE 和 PlatformIO 是灵活的跨平台选择，适合开源和快速开发。选择合适的工具可以大大提高开发效率和代码质量。

**烧录软件:**

STC-ISP

**烧录硬件:**

## 2.stm系列

STMicroelectronics（ST）是一家知名的半导体制造商，它生产了两类不同的产品系列：STM（Standard Microsystems）和STM32。

**STM（Standard Microsystems）**

STM系列产品主要是集成了各种标准接口和功能的微控制器（Microcontrollers）。它们通常用于相对简单的控制任务，例如传感器接口、通信接口等。STM系列微控制器广泛应用于各种嵌入式系统中，如家电、消费电子、工业自动化等领域。

**STM32**

STM32则是STMicroelectronics推出的一系列高性能微控制器，主要基于ARM Cortex-M处理器核心。STM32系列微控制器提供了更高的计算性能、更多的外设接口以及更丰富的功能特性，适用于需要复杂算法和高性能处理的应用场景。STM32系列被广泛用于工业控制、智能家居、汽车电子、医疗设备等需要高度可靠性和性能的领域。

### Stm系列

**开发编译环境**

STM系列芯片是STMicroelectronics推出的一系列基于ARM Cortex-M内核的单片机，常见的开发编译环境和工具如下：

#### 1. STM32CubeIDE(官方免费)

**STM32CubeIDE** 是STMicroelectronics官方推出的集成开发环境（IDE），专门用于STM32系列单片机的开发。

* **特点**：
  + 基于Eclipse开发，提供了完整的编码、构建、调试和分析工具。
  + 集成了STM32CubeMX配置工具，支持自动生成初始化代码。
  + 提供了丰富的STM32 HAL库和Middleware组件。

#### 2. Keil MDK (Microcontroller Development Kit)（免费版与付费版）

**Keil MDK** 是ARM官方推出的集成开发环境，支持多种ARM Cortex-M系列处理器，包括STM32系列单片机。

* **特点**：
  + 提供了完整的开发工具链，包括编译器、调试器和仿真器。
  + 支持Cortex-M系列的所有特性和优化。
  + 集成了Pack Installer，方便获取和管理器件支持包。

#### 3. IAR Embedded Workbench for ARM（收费）

**IAR Embedded Workbench** 是一款专业的集成开发环境，支持多种ARM Cortex-M系列处理器，也适用于STM32系列单片机的开发。

* **特点**：
  + 提供了优秀的编译器优化和调试功能。
  + 集成了RTOS支持和代码分析工具。
  + 支持多种调试器和硬件仿真器。

#### 4. PlatformIO（开源）

**PlatformIO** 是一个开源的跨平台IDE，支持多种嵌入式开发板和框架，包括STM32系列单片机。

* **特点**：
  + 基于Atom和VSCode等开发环境。
  + 支持多种开发板和框架，包括STM32Cube。
  + 集成了多种编译器和调试器，支持C/C++语言开发。

#### 5. STM32CubeMX（官方）

**STM32CubeMX** 是STMicroelectronics提供的配置工具，用于快速生成初始化代码和项目配置。

* **特点**：
  + 图形化界面，支持通过拖放配置引脚、时钟、外设和RTOS等参数。
  + 自动生成初始化代码，支持多种开发环境的导出，如Keil MDK、IAR EW等。

#### 6. GNU工具链(开源)

除了商业的集成开发环境，还可以使用开源的GNU工具链进行STM32系列单片机的开发。

* **特点**：
  + 包括GNU编译器集合（GCC）、调试器（GDB）和其他工具。
  + 适合开发者喜欢自定义和集成的需求。
  + 可以与多种开发环境集成，如Eclipse、VSCode等。

下载官方地址：<https://developer.arm.com/downloads/-/gnu-rm>

#### 7. 其他工具和环境

除了上述主流的开发环境外，还有一些其他工具和环境可以用于STM32系列单片机的开发，具体选择取决于开发者的偏好、项目需求和技术栈。

总体来说，STM32CubeIDE、Keil MDK、IAR EW和PlatformIO是最常见和广泛使用的STM32系列单片机开发环境和工具，它们提供了丰富的功能和支持，适合从初学者到专业开发者的不同需求。

**烧录软件**

**烧录硬件**

### Stm32系列

针对STM32系列芯片的开发，常见的开发编译环境和工具如下：

#### 1. STM32CubeIDE

**STM32CubeIDE** 是STMicroelectronics官方推出的集成开发环境（IDE），专门用于STM32系列微控制器的开发。

* **特点**：
  + 基于Eclipse的开源IDE，具有用户友好的图形化界面。
  + 集成了STM32CubeMX，用于配置和初始化代码生成。
  + 内置了STM32Cube库和各种驱动程序，支持快速开发和调试。

#### 2. Keil MDK-ARM

**Keil MDK-ARM** 是一款商业化的开发环境，提供了强大的编译器和调试器，适用于ARM Cortex-M系列微控制器，包括STM32系列。

* **特点**：
  + 提供了高度优化的ARM编译器。
  + 集成了全面的调试工具和仿真器。
  + 支持广泛的外设驱动和库函数，适用于复杂的应用开发。

#### 3. IAR Embedded Workbench for ARM

**IAR Embedded Workbench for ARM** 是另一款广受欢迎的商业化开发环境，支持ARM Cortex-M系列微控制器，包括STM32系列。

* **特点**：
  + 提供了高效的编译器和调试器。
  + 集成了全面的性能分析和优化工具。
  + 支持实时操作系统（RTOS）和多线程开发。

#### 4. GNU ARM Embedded Toolchain

**GNU ARM Embedded Toolchain** 是一个免费开源的工具链，提供了基于GCC的编译器，适用于ARM Cortex-M系列微控制器，包括STM32系列。

* **特点**：
  + 免费开源，适合开发者进行定制和扩展。
  + 可以与多种开发环境集成，如Eclipse、Visual Studio Code等。
  + 提供了广泛的库支持和社区资源。

#### 5. STM32CubeMX

**STM32CubeMX** 是STMicroelectronics官方提供的配置工具，用于快速配置STM32系列微控制器的引脚分配、时钟设置和外设配置。

* **特点**：
  + 可以生成基于HAL库或LL库的初始化代码。
  + 支持各种外设的配置，如UART、SPI、I2C等。
  + 可以与STM32CubeIDE和其他开发环境无缝集成。

#### 6. Visual Studio Code + PlatformIO

**Visual Studio Code** 是一款轻量级的开源IDE，结合 **PlatformIO** 插件可以支持STM32系列的开发。

* **特点**：
  + 提供了丰富的插件支持，包括PlatformIO作为主要的STM32开发工具。
  + 可以跨平台使用，支持多种编程语言和硬件平台。
  + 社区支持活跃，有大量的扩展和库可用。

#### 7. 其他工具和环境

除了上述列出的主流开发环境外，还有一些其他工具和环境可以用于STM32系列微控制器的开发，如Eclipse + GNU ARM Eclipse插件、TrueSTUDIO等。

#### 选择与考虑因素

在选择STM32系列芯片的开发编译环境时，需要考虑以下因素：

* **功能需求**：根据项目的功能需求选择合适的开发工具，例如需要强大的调试功能、优化编译器、集成的库和驱动等。
* **成本**：商业工具如Keil MDK-ARM和IAR Embedded Workbench具有丰富的功能，但有较高的购买成本；而免费开源的工具链如GNU ARM Embedded Toolchain则经济实惠。
* **使用习惯**：个人或团队的开发习惯和技能水平，以及对特定IDE的熟悉度也是选择因素之一。

根据具体的开发需求和项目特点，选择适合的开发环境能够提高开发效率和产品质量。

## 3.esp32芯片

**开发编译环境**

ESP32是一款由乐鑫（Espressif Systems）推出的双核Wi-Fi和蓝牙组合的芯片，通常使用的开发编译环境和工具如下：

### 1. ESP-IDF (Espressif IoT Development Framework)（官方）

**ESP-IDF** 是乐鑫官方提供的开发框架，专门用于ESP32和ESP32-S系列芯片的开发。

* **特点**：
  + 提供完整的开发工具链，包括编译器、调试器和工具。
  + 支持FreeRTOS实时操作系统。
  + 包含丰富的示例和组件库。

### 2. Arduino IDE with ESP32 Core（免费）

**Arduino IDE** 是一个简单易用的开发环境，通过安装ESP32 Core扩展，可以支持ESP32芯片的开发。

* **特点**：
  + 集成了Arduino风格的编程语言和库。
  + 支持通过简单的图形界面和编程接口进行开发。
  + 社区支持丰富，有大量的示例和库可用。

### 3. PlatformIO（开源）

**PlatformIO** 是一个开源的跨平台IDE，支持多种嵌入式开发板和框架，包括ESP32芯片。

* **特点**：
  + 基于Atom和VSCode等开发环境。
  + 支持多种开发板和框架，包括ESP32。
  + 集成了多种编译器和调试器，支持C/C++语言开发。

### 4. Espressif ESP32 Development Boards（官方）

**Espressif提供的开发板**和工具，如ESP-Prog调试器，也是常用的ESP32开发工具。

* **特点**：
  + 提供了与ESP32芯片兼容的开发板和调试器。
  + 集成了必要的调试和烧录功能。
  + 可以直接用于ESP-IDF的开发和调试。

### 5. Visual Studio Code + PlatformIO Extension

**Visual Studio Code** 是一款轻量级的跨平台IDE，通过安装PlatformIO插件，可以进行ESP32的开发。

* **特点**：
  + 强大的插件生态系统，支持丰富的功能扩展。
  + 提供了类似于PlatformIO的集成开发环境。
  + 可以与其他工具和版本控制系统集成。

### 6. Eclipse + ESP32 Tools

对于喜欢使用Eclipse的开发者，也可以使用ESP32 Tools插件进行ESP32开发。

* **特点**：
  + 提供了Eclipse IDE的所有功能和特性。
  + 集成了ESP32的开发工具和调试器。
  + 支持丰富的插件和扩展。

### 7. 其他工具和环境

除了上述列出的主流开发环境外，还有一些其他工具和环境可以用于ESP32的开发，如基于Python的Micropython等。

选择合适的开发环境和工具通常取决于开发者的经验、项目需求和团队偏好。ESP-IDF和Arduino IDE是最为流行和常用的ESP32开发工具，它们提供了丰富的功能和社区支持，适合从初学者到专业开发者的不同需求。

**烧写工具**

## 4.AVR

**开发编译环境**

AVR是一种由Atmel（现在被Microchip收购）推出的8位微控制器系列，常见的开发编译环境和工具如下：

### 1. Atmel Studio

**Atmel Studio** 是由Microchip官方提供的集成开发环境（IDE），专门用于AVR和ARM微控制器的开发。

* **特点**：
  + 提供完整的开发工具链，包括编译器、调试器和仿真器。
  + 集成了Atmel START工具，用于快速配置和初始化代码生成。
  + 支持AVR Studio 4/5/6的项目导入和转换。

### 2. MPLAB X IDE + AVR Plugin

**MPLAB X IDE** 是Microchip官方推出的开源跨平台IDE，通过安装AVR插件，可以支持AVR系列微控制器的开发。

* **特点**：
  + 支持多种Microchip微控制器的开发，包括AVR系列。
  + 集成了丰富的插件和工具支持。
  + 提供了强大的调试功能和性能分析工具。

### 3. CodeVisionAVR

**CodeVisionAVR** 是一款商业的AVR开发环境，提供了专业的开发工具和支持。

* **特点**：
  + 提供了高度优化的AVR编译器和调试器。
  + 集成了图形化配置工具，支持快速开发和调试。
  + 提供了丰富的例程和库文件支持。

### 4. AVR-GCC

**AVR-GCC** 是GNU Compiler Collection (GCC) 的一部分，提供了开源的C/C++编译器，支持AVR系列微控制器。

* **特点**：
  + 免费开源，支持跨平台运行。
  + 提供了高度优化的编译器。
  + 可以与多种IDE集成，如Eclipse、Visual Studio Code等。

### 5. Eclipse + AVR Plugin

**Eclipse** 是一款流行的开源跨平台IDE，通过安装AVR插件，可以支持AVR微控制器的开发。

* **特点**：
  + 强大的插件生态系统，支持丰富的功能扩展。
  + 可以集成AVR-GCC和调试器，支持单步调试和性能分析。
  + 支持多种版本控制系统和项目管理工具。

### 6. Arduino IDE with AVR Core

**Arduino IDE** 是一个简单易用的开发环境，通过安装AVR Core扩展，可以支持AVR系列微控制器的开发。

* **特点**：
  + 集成了Arduino风格的编程语言和库。
  + 支持通过简单的图形界面和编程接口进行开发。
  + 社区支持丰富，有大量的示例和库可用。

### 7. 其他工具和环境

除了上述列出的主流开发环境外，还有一些其他工具和环境可以用于AVR系列微控制器的开发，具体选择取决于开发者的经验、项目需求和技术栈。

总体来说，Atmel Studio和MPLAB X IDE是官方推荐的主要开发环境，提供了完整的工具链和支持。AVR-GCC作为开源的选择，可以与多种IDE集成使用，适合喜欢自定义和跨平台开发的开发者。Arduino IDE则适合初学者和快速原型开发。

**烧写工具**

## 5.PIC

**开发编译环境**

PIC（Peripheral Interface Controller）是由Microchip Technology推出的一系列8位和16位微控制器。常见的PIC开发编译环境和工具如下：

### 1. MPLAB X IDE

**MPLAB X IDE** 是Microchip官方推出的开源跨平台IDE，支持多种Microchip微控制器，包括PIC系列。

* **特点**：
  + 提供完整的开发工具链，包括编译器、调试器和仿真器。
  + 集成了丰富的插件和工具支持。
  + 提供了强大的调试功能和性能分析工具。

### 2. MPLAB XC Compiler

**MPLAB XC Compiler** 是Microchip官方提供的一系列专业的编译器，支持各种PIC系列微控制器的开发。

* **特点**：
  + 提供了针对PIC架构优化的C编译器。
  + 支持高级优化和代码生成。
  + 集成了与MPLAB X IDE无缝连接的功能。

### 3. CCS C Compiler

**CCS C Compiler** 是一款商业的C编译器，专门用于PIC微控制器的开发。

* **特点**：
  + 提供了高度优化的PIC编译器。
  + 支持丰富的PIC特定库和组件。
  + 适合快速开发和原型验证。

### 4. Hi-Tech C Compiler

**Hi-Tech C Compiler** 是Microchip旗下的一款C编译器，专门用于PIC系列微控制器的开发。

* **特点**：
  + 提供了高效的编译器和代码生成。
  + 支持多种PIC系列微控制器。
  + 集成了丰富的标准C库和PIC特定库。

### 5. Proteus Design Suite

**Proteus Design Suite** 是一款集成的电子设计自动化软件，也是一种仿真环境，可以用于PIC微控制器的软硬件开发。

* **特点**：
  + 提供了完整的电路设计和仿真功能。
  + 支持PIC微控制器的模拟和调试。
  + 可以与MPLAB X IDE等工具集成。

### 6. Arduino IDE with PIC32 Core

**Arduino IDE** 是一个简单易用的开发环境，通过安装PIC32 Core扩展，可以支持PIC32系列微控制器的开发。

* **特点**：
  + 集成了Arduino风格的编程语言和库。
  + 支持通过简单的图形界面和编程接口进行开发。
  + 社区支持丰富，有大量的示例和库可用。

### 7. 其他工具和环境

除了上述列出的主流开发环境外，还有一些其他工具和环境可以用于PIC微控制器的开发，具体选择取决于开发者的经验、项目需求和技术栈。

总体来说，MPLAB X IDE和相关的MPLAB XC Compiler是Microchip官方推荐的主要开发工具，适合从初学者到专业开发者的不同需求。其他的商业编译器和仿真工具也提供了丰富的功能和支持，可以根据具体需求进行选择和使用。

**烧写工具**

## 6.XXXX

# 二．项目准备

1. **单片机DIV常见封装形式**

DIP

SIP

TO

SOIC

1. **元器件购买平台**

嘉立创

<https://www.szlcsc.com/>

云汉芯城

<https://www.ickey.cn/>

华强商城

<https://www.hqbuy.com/>

远程通过esp控制家里服务器

单片机小项目(待定)

1. **完成需求**

**Esp32控制家里电脑开关电源，stm单片机小项目**

1. **Xxx**

# 三．XXXXX