# 1.单片机介绍

#### 教学要求

1.了解单片机的基本概念及应用

2.熟练掌握单片机的软硬件知识

3.能应用单片机开发板完成基本实验

4.能完成简单开发单片机应用

**单片机介绍**

1.单片机（Microcontrollers）是一种集成电路芯片，是把CPU、RAM、ROM、多种I/O口和中断系统、定时器/计数器等功能集成到一块硅片上,构成的一个小而完善的微型计算机系统。在工业控制领域广泛应用。

2.单片机为工业测量控制而设计，又成为微控制器MCU.

3.具有三高优势（集成度高、可靠性高、性价比高）

4。单片机的使用领域已十分广泛，如智能仪表、实时工控、通讯设备、导航系统、家用电器等。各种产品一旦用上了单片机，就能起到使产品升级换代的功效，常在产品名称前冠以形容词——“智能型”，如智能型洗衣机等。

**目前常用单片机**

1. 51系列单片机，有很多公司生产，用得人最多，价格最便宜，资料丰富，对性能要求不高的场合大量被采用
2. ATMEL公司AVR单片机，其显著的特点为高性能、高速度、低功耗，
3. PIC单片机系列是美国微芯公司（Microship）的产品，共分三个级别，即基本级、中级、高级
4. ST厂商推出的STM32系列单片机，性价比超高的系列单片机，ARM Cortex-M内核

**单片机的历史及发展**

1.第一阶段（1974-1978）：初级单片机阶段。以Intel公司MCS-48为代表，8位单片机,RAM 64Byte,ROM 1KB , 主频 6MHz

2.第二阶段(1978-1982)：高性能单片机阶段。以Intel公司MCS-51为代表，8位单片机,RAM 128Byte,ROM 4KB , 主频 12MHz

3.第三阶段（1983-1990）：巩固发展阶段。8位单片机全面发展，16位单片机的出现。

4.第四阶段（1990至今）:MCU全面发展阶段.出现了高速、大寻址范围、强运算能力的8位/16位/32位/64位通用型单片机

**单片机的发展趋势**

1.低功耗化

2.低电压化

3.大容量化

4.高性能化

**单片机主要技术指标**

1.字长

8位/16位/32位/64位

2.内存容量

64Byte,1KB,20KB,64KB

3.运算速度

可以是主频，每秒运算次数

6MHz,12MHz,72MHz

# **STM32单片机介绍**

STM32 单片机是意法半导体（ST）公司的生产的基于 ARM Cortex M3 架构的 32 位精简

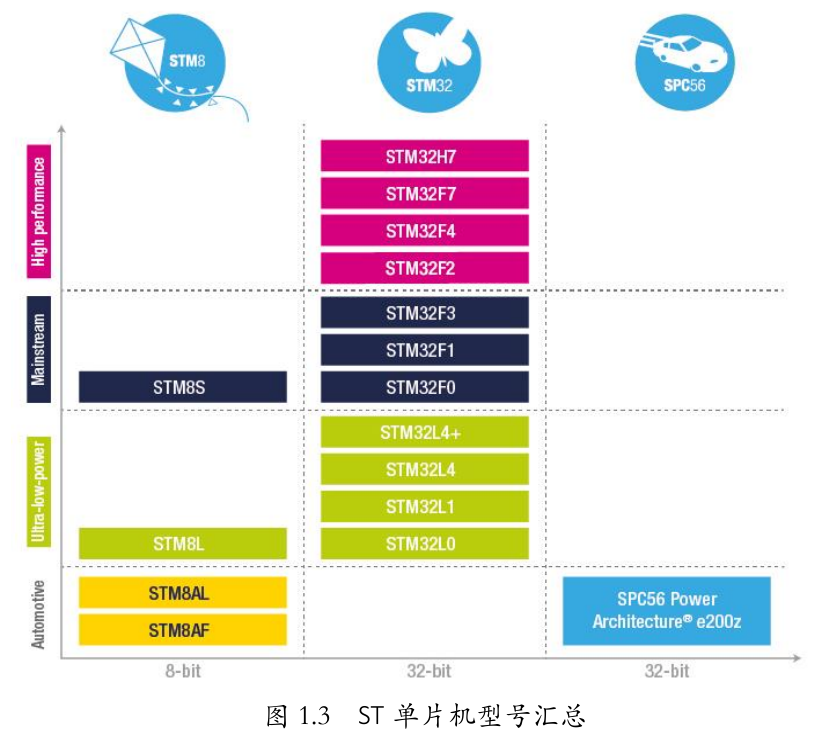
指令集单片机。

ARM可以有两个含义，一是指英国ARM公司，它是单片机设计公司。二是指ARM公司设计的CPU内核与CPU架构。

1.STM32单片机分类：

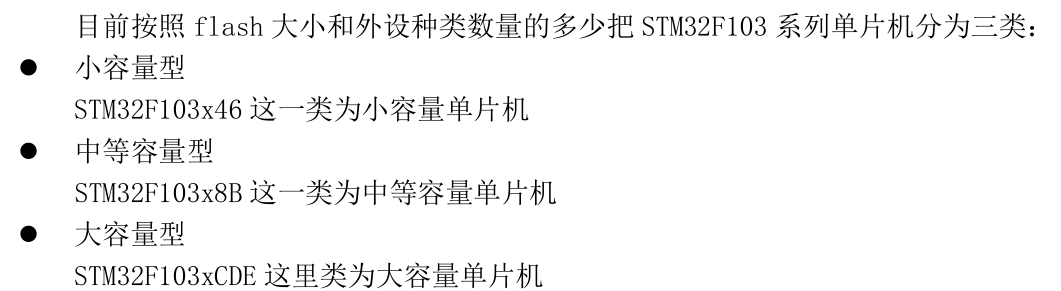


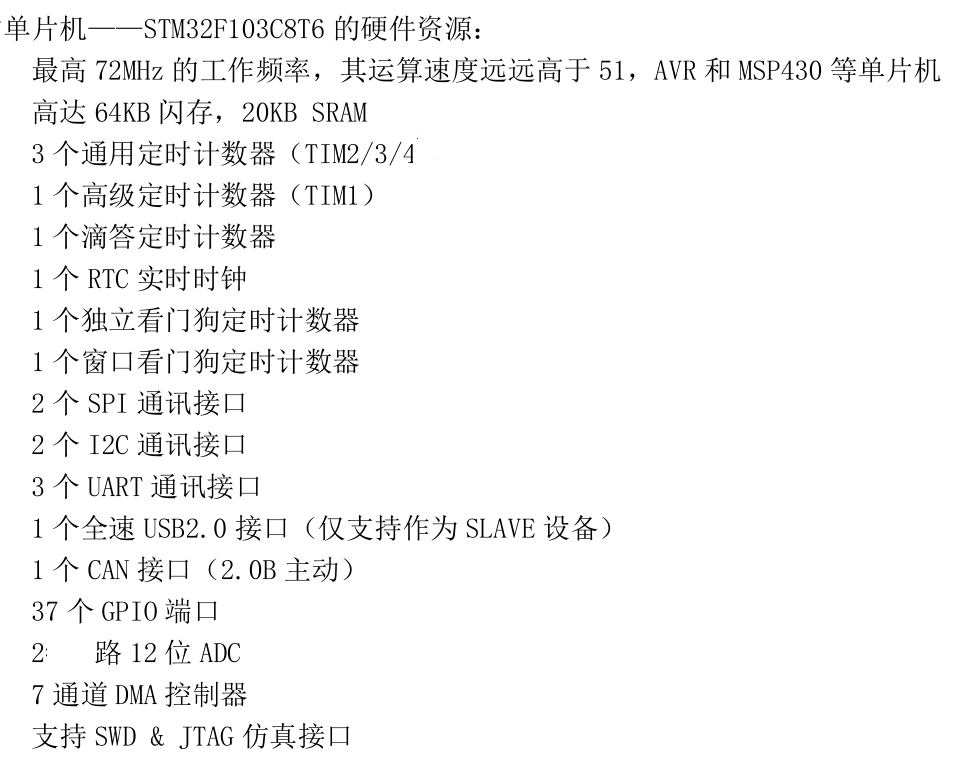
2.STM32单片机型号汇总：



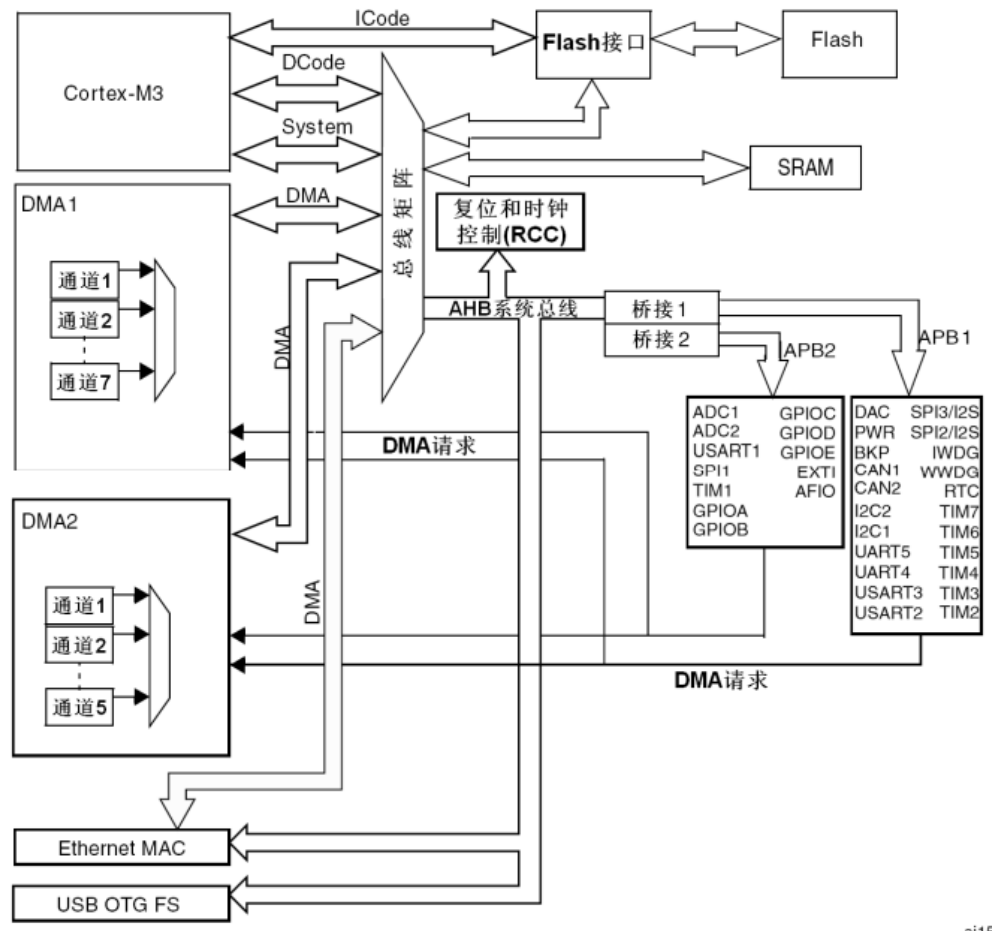
1. ST 单片机的命名规则



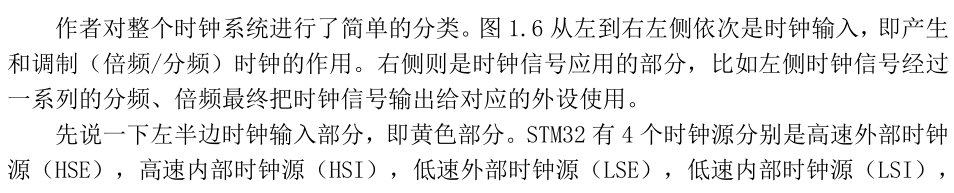


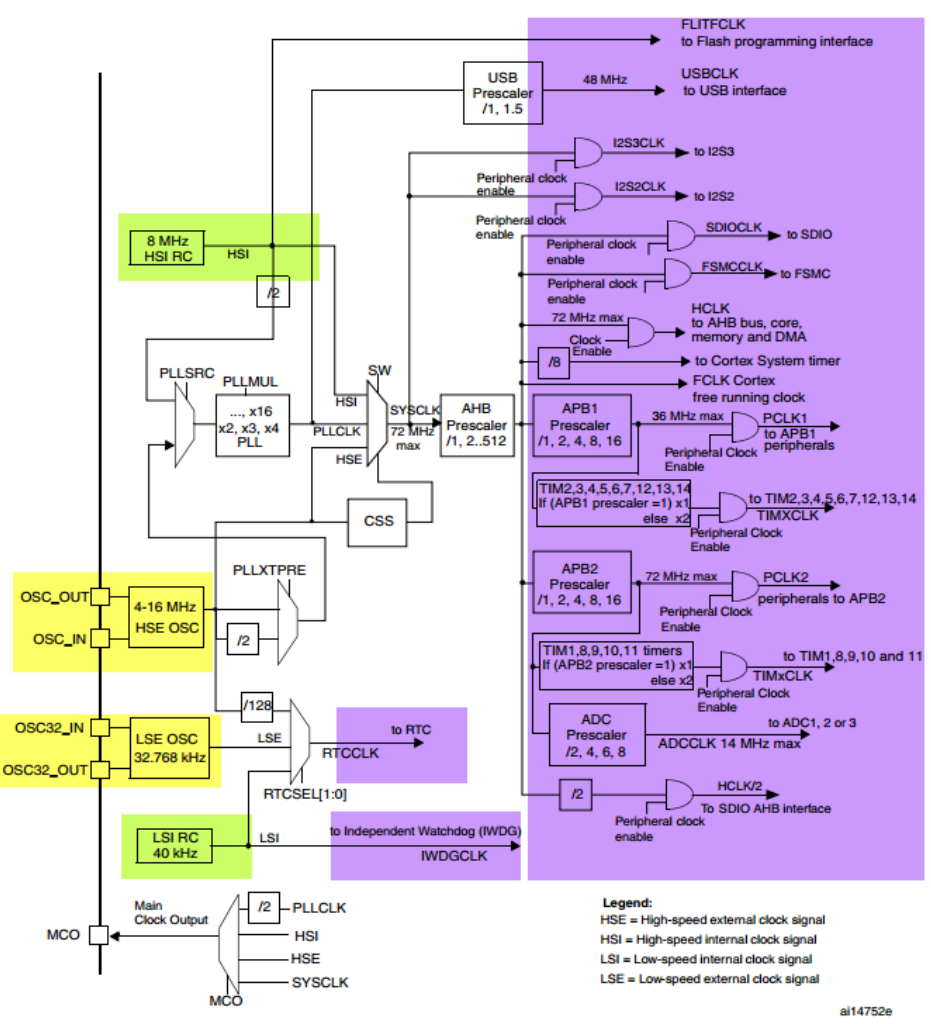


STM32 整体的架构框图



STM32F103 单片机时钟系统





# **3.C语言介绍**

C 语言是一种通用的、面向过程式的计算机程序设计语言。1972 年，为了移植与开发 UNIX 操作系统，丹尼斯·里奇在贝尔电话实验室设计开发了 C 语言。

C 语言是一种广泛使用的计算机语言，它与 Java 编程语言一样普及，二者在现代软件程序员之间都得到广泛使用。

当前最新的C语言标准为 C11 ，在它之前的C语言标准为 C99。

# C 程序结构

C 程序主要包括以下部分：

* 预处理器指令
* 函数
* 变量
* 语句 & 表达式
* 注释

让我们看一段简单的代码，可以输出单词 "Hello World"：

## 实例

#include <stdio.h>

int main()

{

/\* 我的第一个 C 程序 \*/

printf("Hello, World! \n");

return 0;

}

接下来我们讲解一下上面这段程序：

1. 程序的第一行 *#include <stdio.h>* 是预处理器指令，告诉 C 编译器在实际编译之前要包含 stdio.h 文件。
2. 下一行 *int main()* 是主函数，程序从这里开始执行。
3. 下一行 /\*...\*/ 将会被编译器忽略，这里放置程序的注释内容。它们被称为程序的注释。
4. 下一行 *printf(...)* 是 C 中另一个可用的函数，会在屏幕上显示消息 "Hello, World!"。
5. 下一行 **return 0;** 终止 main() 函数，并返回值 0。

## 编译 & 执行 C 程序

# C 基本语法

1. 在 C 程序中，分号是语句结束符。也就是说，每个语句必须以分号结束。
2. **C 语言有两种注释方式：**

// 单行注释

以 **//** 开始的单行注释，这种注释可以单独占一行。

/\* 单行注释 \*//\*

多行注释

多行注释

多行注释

\*/

**/\* \*/**这种格式的注释可以单行或多行。

您不能在注释内嵌套注释，注释也不能出现在字符串或字符值中。

1. 标识符:  
   是用来标识变量、函数，或任何其他用户自定义项目的名称。一个标识符以字母 A-Z 或 a-z 或下划线 \_ 开始，后跟零个或多个字母、下划线和数字（0-9）。

C 标识符内不允许出现标点字符，比如 @、$ 和 %。  
  
4.C 是区分大小写的编程语言。因此，在 C 中，Name 和 name是两个不同的标识符。

## 保留字

下表列出了 C 中的保留字。这些保留字不能作为常量名、变量名或其他标识符名称。

|  |  |
| --- | --- |
| **关键字** | **说明** |
| auto | 声明自动变量 |
| break | 跳出当前循环 |
| case | 开关语句分支 |
| char | 声明字符型变量或函数返回值类型 |
| const | 声明只读变量 |
| continue | 结束当前循环，开始下一轮循环 |
| default | 开关语句中的"其它"分支 |
| do | 循环语句的循环体 |
| double | 声明双精度浮点型变量或函数返回值类型 |
| else | 条件语句否定分支（与 if 连用） |
| enum | 声明枚举类型 |
| extern | 声明变量或函数是在其它文件或本文件的其他位置定义 |
| float | 声明浮点型变量或函数返回值类型 |
| for | 一种循环语句 |
| goto | 无条件跳转语句 |
| if | 条件语句 |
| int | 声明整型变量或函数 |
| long | 声明长整型变量或函数返回值类型 |
| register | 声明寄存器变量 |
| return | 子程序返回语句（可以带参数，也可不带参数） |
| short | 声明短整型变量或函数 |
| signed | 声明有符号类型变量或函数 |
| sizeof | 计算数据类型或变量长度（即所占字节数） |
| static | 声明静态变量 |
| struct | 声明结构体类型 |
| switch | 用于开关语句 |
| typedef | 用以给数据类型取别名 |
| unsigned | 声明无符号类型变量或函数 |
| union | 声明共用体类型 |
| void | 声明函数无返回值或无参数，声明无类型指针 |
| volatile | 说明变量在程序执行中可被隐含地改变 |
| while | 循环语句的循环条件 |

# C 数据类型

在 C 语言中，数据类型指的是用于声明不同类型的变量或函数的一个广泛的系统。变量的类型决定了变量存储占用的空间，以及如何解释存储的位模式。

C 中的类型可分为以下几种：

|  |  |
| --- | --- |
| **序号** | **类型与描述** |
| 1 | **基本类型：** 它们是算术类型，包括两种类型：整数类型和浮点类型。 |
| 2 | **枚举类型：** 它们也是算术类型，被用来定义在程序中只能赋予其一定的离散整数值的变量。 |
| 3 | **void 类型：** 类型说明符 *void* 表明没有可用的值。 |
| 4 | **派生类型：** 它们包括：指针类型、数组类型、结构类型、共用体类型和函数类型。 |

## 整数类型

下表列出了关于标准整数类型的存储大小和值范围的细节：

|  |  |  |
| --- | --- | --- |
| **类型** | **存储大小** | **值范围** |
| char | 1 字节 | -128 到 127 或 0 到 255 |
| unsigned char | 1 字节 | 0 到 255 |
| signed char | 1 字节 | -128 到 127 |
| int | 2 或 4 字节 | -32,768 到 32,767 或 -2,147,483,648 到 2,147,483,647 |
| unsigned int | 2 或 4 字节 | 0 到 65,535 或 0 到 4,294,967,295 |
| short | 2 字节 | -32,768 到 32,767 |
| unsigned short | 2 字节 | 0 到 65,535 |
| long | 4 字节 | -2,147,483,648 到 2,147,483,647 |
| unsigned long | 4 字节 | 0 到 4,294,967,295 |

## 浮点类型

下表列出了关于标准浮点类型的存储大小、值范围和精度的细节：

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **存储大小** | **值范围** | **精度** |
| float | 4 字节 | 1.2E-38 到 3.4E+38 | 6 位小数 |
| double | 8 字节 | 2.3E-308 到 1.7E+308 | 15 位小数 |
| long double | 16 字节 | 3.4E-4932 到 1.1E+4932 | 19 位小数 |

## void 类型

void 类型指定没有可用的值。它通常用于以下三种情况下：

|  |  |
| --- | --- |
| **序号** | **类型与描述** |
| 1 | **函数返回为空** C 中有各种函数都不返回值，或者您可以说它们返回空。不返回值的函数的返回类型为空。例如 **void exit (int status);** |
| 2 | **函数参数为空** C 中有各种函数不接受任何参数。不带参数的函数可以接受一个 void。例如 **int rand(void);** |
| 3 | **指针指向 void** 类型为 void \* 的指针代表对象的地址，而不是类型。例如，内存分配函数 **void \*malloc( size\_t size );** 返回指向 void 的指针，可以转换为任何数据类型。 |