|  |
| --- |
| 一、入门总结  1.1为什么要把时间花在“犹豫”上？  ! Q- c  u2 l" U  每当我们在入门之前（ARM是这样，DSP也一样），总会有很多疑问，会有很多顾虑。我们渴望知道学习STM32前景如何？需要啥基础？难不难？适不适合我？但是什么时候能心潮澎湃地、相当着急地开始学STM32？日子在一天一天过去！你开始行动了吗？没有行动的思索，永远都不可能入门！把这些时间用来看书吧，效果能好一万倍。  大家可能是从51单片机过来的，回想一下，我们之前学单片机时如何入门呢？实际上都是先看书（理论），再玩板子（实践）。严格地说，应该是模仿实验。熟悉之后才会自己写程序代码实现某个功能。因此，如果你正在咨询STM32；如果你正对STM32心潮澎湃；如果你想入门STM32；那么，从现在开始，不要犹豫了，不要想再详细地了解STM32的前景了。做一个可能影响你一生的决定吧！不用咨询，不用兴奋，开始看书籍（文档）吧！！每个人都是这么走过来的。  1.2看资料需要计划、耐心和速度  + d' u! ?6 X8 {  这里所谓的“资料”包括STM32书籍、文档。因为STM32有个特点，datasheet很多都是中文的，有些同学就没有去买书籍，直接看STM32的用户手册，也是可以的。但是不管看书籍还是文档，我们是需要计划的。不是今天看3页，明天看5页。一本书看了两个月，还在磨蹭。请记住，你学的不是寂寞，是STM32！  看书或文档不是用来消遣时间的。背水一战吧，给自己规定一个底线：两周内把一定粗略地过一遍！不要求都看懂，事实上，不可能都看懂。但我们必须理解基本知识，对难度高的知识有一个印象，至少以后碰到问题的时候，我们会似曾相识，感觉在哪里见到过，于是翻资料上网找答案——带着问题的时候，效率才是超高的。  两周过去了，STM32的知识你过一遍了吗？没看完？那么，你应该考虑这些天中，你是否尽力了？是否真的想学STM32？或者还是看到STM32人气很高，也想跟下时髦？是不是碰到问题没法解决就想放弃了？现在是你做第二个决定的时候了，请决定你是否继续攻读STM32。如果是一时冲动想跟时髦，请马上放弃——你已经浪费两周时间了，你还想再浪费14天吗？如果想吃得苦中苦，方为人上人，OK，请继续往下看：）  1.3学STM32必备开发板    a6 m& w! x( k& k! }4 f  稍微了解一些理论知识后，现在摆在你面前，有两条路：  第一，自己画PCB制板、焊接、调试。  第二，购买一套性价比高的开发板。  笔者之前做过几次PCB设计，而且当时在实验室可以报销所有费用，就自己做了一个PCB板。建议有报销条件的同学自己设计一块板子学习。毕竟2层PCB的设计不是很难。无法报销的同学，可以选购一款开发板来学。  % ?' A4 ~, M  O. T- w" ]  总之一句话，选购的开发板一定要性价比最高的，最实惠的，而不是最贵的。实际上，大多数同学选购的开发板是用来学习的，不是用来项目开发的。因此，能让你上手STM32，目的已经达到了。  . i7 {: ]1 x1 h: Q8 T  1.4熟悉开发板并试图写程序  4 }; A# c, ]; M0 B- o+ n' }  这时的你，不能再停留在理论上了。你现在的心情，应当是：看到开发板，马上就有一种说不出的兴奋！但是，心急吃不了热豆腐！请耐心把厂家配套的开发板使用手册浏览一遍，熟悉你买的开发板。按照使用手册，把开发板上的相关测试、操作步骤，都动手做一遍，你会发现，原来开发STM32是如此简单！那么就让我们开始STM32的开发之路吧！  " U" V, p, M- F4 d2 r! a6 T  如何进行真正入门STM32，请看下面《STM32入门方法谈》。  . w( C5 F9 g# S  ?  二、STM32入门方法谈  2.1拿到开发板我该做什么？  7 n  d5 [+ s. `6 [6 u! |- \  国内用户有一个普遍现象：用户拿到开发板后，不知道如何下手。由于没有自己的思路，又是初学者，也没有好的突破口，因此，买来开发板后，开机，看看触摸屏，感叹“好漂亮”的屏幕显示，然后就迷茫了。开发板是不会告诉你如何使用它的。开发板的光盘？光盘里放置一大堆资料——我该先看哪个？后看哪个？总之，我如何开始STM32的入门学习？？  从拿到开发板后需要了解如何安装J-Link、MDK（keil）调试软件，一直到STM32上的UCOS移植，配合专门的教程，会在学习STM32的过程中少走许多弯路，提高学习效率了。  2.2我的时间如何安排？  笔者做任何事情前，习惯写一个计划——可能是在公司的时候，每天、每周、每月必写计划和总结养成的“毛病”，呵呵。正如在上一节中提及：我们要用一个月内上手STM32！我们开始考虑，如何更好地利用好这一个月的时间呢？  我想起了第一天公司入职的情景。我的工作导师让我先领一台PC，安装必要的软件，然后告诉我近一个月的任务是什么。当时我很不屑，因为刚参加了公司组织的培训，以为自己啥都会，但是失败了。原因：盲目进行项目，没有一个计划。  # q) C( o9 p" ~3 Q- a  没有计划的日子会很惨。就像每天早上醒来睁开眼睛，却不知道自己今天要干啥。闲话不说，我们谈一谈计划和时间安排。  笔者认为，学习STM32，实际上也是进行项目的过程。进行项目，就必须有项目进度文档，至少需要预计项目的耗时。虽然没有那么规范，但建议大家至少要给自己一个时间表：告诉自己从2012年x月y日至2012年m月n日，必须能够进行独立的项目开发。比如定为30天，把这个时间分三个阶段：  ' \_$ ]8 ^7 |9 B9 d7 l  第一阶段：找感觉——谈及STM32，立即反应到调试的方法如何。  第二阶段：修改例程——按自己的意愿修改例程，让自己有点成就感。  第三阶段：主动攻击——给自己找一个小项目，把它实现出来。  0 o6 F4 h8 B! I9 ?  每个阶段有一个目标后，再细化该目标到每天，写出每天要完成的任务。注意，每天的任务不可太多，否则时间太紧，完成不了当天的任务，反而会进入恶性循环，导致第二天、第三天，以后的任务也将无法完成。最终导致整个计划的失败。  2 o6 m6 g+ @( W# v8 ^  每天的任务需要根据自己的具体情况来定。如果有充足的时间，可以安排多一些任务。如果每天只有2个小时，切记，宁可少安排任务，也不可追求加快项目进度。当然，心中有事情后，每天自然就会忙碌起来！你的学习生活会更加充实！  2.3碰到问题怎么办？  学习难免会遇到问题。在公司参加培训的时候，学员被告知：困难即机会。  9 M, l6 l8 Z\* |4 M\* I/ t$ U  所以希望大家坦然面对问题，不能逃避。一小部分同学在遇到问题的时候，总是想：能否避开走另一条路？比如调试程序，某一个程序的语法不太懂，就想用另一种表示方法来代替，或者直接注释掉。——这是很可怕的。因为你现在一旦放过一个问题，可能意味着你一辈子可能都没有机会去碰这个问题。那么你的技术无法进入最高境界。我们是为学习而来，说难听点，我们现在的付出，是想以后能有更好的回报的。  此外，碰到问题，不能马上就到处求助。  原因：  1、这个问题可能是你的疑问，但它还不具体，你无法用最好的语言让别人接受、理解，并且回答它。很多网友看到模凌两可的问题，是不会回答的。比如Linux如何移植？这个问题，如果详细说，需要用半本书来解释——没有人愿意花两三个小时告诉你如何移植。所以这时候，你要做的，是先去查资料，自己试图解决它。当碰到某一个具体的步骤，实在想不通的时候，比如Linux移植过程中，文件挂载打印卡住了。这个现象，可能对有经验的人来说，一句话就能点破。而你因为有先前研究的基础，一听即可知道哪里出问题。这样的方式，学习效率是最高的。  ) M+ `\* x+ j( o5 u+ b2 h- e4 A  2、CPU是相通的，相信大部分的同学都学习过单片机，是有一定基础的。如果你碰到问题，去尝试了，自己把问题解决了，你会很有成就感！！因为这是你独立思考的结果！！当然，你应该写一个总结文档，收藏在PC中备忘。也可以把总结文档发帖到论坛上，这样的原创文档，跟帖率是最高的。慢慢地，你在这个领域，有了网络知名度。——这绝对是一笔财富。  0 D5 k$ I2 Q; B6 Y' U. k$ x  当然，笔者在此并不是不让大家使用别的方式讨论问题。相反，我提倡大家踊跃发言讨论。刚才是就事论事，也是笔者自己的一点小经历。经历了，才发现如何做才更有效率。  7 Q5 z' ~: }+ f' t6 ]  三、STM32学习步骤  在第二章，我们已经谈及到学习需要计划。那么，在这章节，我们就来详细地阐述这个计划——仅供参考。  3.1关于STM32文档学习  0 l8 X: [% T; W' B  本期教程，我们假定大家已经对STM32的书籍或者文档有一定的理解。如不理解，请立即阅读STM32的文档，以获取最基本的知识点。  STM32的学习与ARM9的学习有一个很大的区别。ARM9的学习一般是需要购买书籍的。比如三星的S3C2440，官方的文档都是英文的，大部分工程师只能去看国内出版的书籍。英文好的同学，请不要以为你很牛，可以只看英文文档。毕竟你是中国人，你最熟悉的，理解最好的还是中文。看英文的速度还是比看中文慢一些，我们要的是最短的时间，而不是追求短时间内记住所有细节。当然，如果是一篇论文，建议看英文原版还是有好处的。  STM32处理器进入国内市场时候，ST官方（或者第三方）的推广工作做的非常好。翻译了大量的英文文档，迎合了国内的很多工程师的思维。所以现在大部分STM32F103xxx的用户datasheet都有中文版。因此可以不用去购买书籍，看电子档即可。当然，有雄厚经济条件的朋友可以购买本书籍哈~  学习的时候，关注两个比较重要的文档：《STM32F103xxx参考手册》，《STM32固件库使用手册》。  是针对有充足的时间、精力的同学，建议去下载需要查阅的文档，以获取更多信息。阅读《STM32F103xxx参考手册》，一定要注意，不需要全部阅读——没有时间的。建议选读，但是前几章必读。存储器和总线架构、电源控制、备份寄存器、复位和时钟控制，通用和复用功能I/O，中断和时间等等部分一定要花时间阅读。后面章节，讲述的是具体的功能模块设计。如果我们用到哪个模块，就可以去阅读哪个模块。比如在使用AD的时候，就需要去阅读第10章ADC。其他不举例。相信每个初学者都有自己的研究方向和判断。  - k' Z4 b+ |# r\* b8 s; x. o  阅读《STM32固件库使用手册》，主要是为了简化编程。STM32给我们提供了一个非常好的固件函数库，我们只要去调用即可。当然，我们也可以不去碰这些固件库——传说使用它会使得代码效率变低，是有道理的。网络上也出现了很多网友自己写的代码，没有使用带固件库函数。如何取舍，在于您的选择。这里我主要强调的是，阅读《STM32固件库使用手册》的时候，前面几章也是必须阅读的。比如第一章文档和库规范中的命名规则，编码规则，这些都是需要注意的。第二章是最关键的，希望大家熟读。第二章描述了固件库的架构，我们如何去使用固件库的步骤等。有了第二章的基础，我们就可以借助固件库写出自己的代码了。第四章开始之后，就可以根据大家的需要来阅读。实际上，后续的章节，都是描述某个模块有什么什么函数，每个函数如何使用等。  关于后面的章节，建议对GPIO库函数、中断部分库函数、复位和时钟设置的库函数要比较熟悉，因为平时经常会用到。  0 G1 H5 N3 Z: ]% m( \_+ n\* j$ K  以上提到的这两个文档，已经足够您看的了，呵呵。希望您能从中获取大量的STM32基础知识。  / m; ^7 z6 a% B  l0 V  3.2 30天上手STM32计划  （1）这里所谓的“上手”，指的是能理解并掌握一些常用的STM32外设，真正想掌握一款处理器，30天根本说明不了什么问题。只能说，你已经入门了。但是，这对我们初学者来说已经足够了。  （2）这里所谓的“30天”，根据每个人的时间安排不同而不同。  8 Q8 R/ [6 B$ A: o3 P- ~  如果您每天有充足的时间学习，那么可以规定自己尽快地能独立地进行简单的STM32开发。如果您每天只有业余时间来学习STM32，建议根据您的具体情况安排时间。毕竟计划时间如果安排太紧，反而收不到良好的效果，只能进入恶性循环，这是我们要避免的。  : ]0 z, n\* N4 P\* A$ Q) k( {& y. ?  但是建议，不管您有无充足的时间，都必须给自己做一个计划！  $ g) X# b6 ]7 N+ I, \_( W  我这里列出一个思路，仅供参考：  - |  L/ i( N: l3 o  步骤一，安装完STM32学习的软件，比如J-Link、KeilforARM（MDK）、ISP（如果需要从串口下载的话）。这些软件安装的详细步骤，可以参考我们推出的相应教程进行。  5 o  a" d0 k2 W+ B1 [  步骤二，挑选部分例程的HEX，比如LED灯的例程HEX文件，下载到STM32开发板中，观察LED灯的闪烁情况。这部分的操作，可以参考我们推出的相应的教程进行。  8 X& F  n( s) m; u' \_0 ~" G( T% j/ f  实际上，以上两个步骤，是为了熟悉要使用的工具软件而已。属于找感觉的阶段。其实我们还没开始STM32的学习呢！  $ U2 r2 g\* |4 y0 k  步骤三，准备几个常用的文档，比如《STM32的用户手册》，《STM32固件库使用手册》等文档。用于平时查阅。这些文档，在光盘中的芯片手册目录中均可找到。  : R  k( y8 F8 l9 s# {2 u' W  步骤四，开始查看例程的编写，看看例程是如何写的，自己可否修改下例程，达到自己想要的效果呢？芯达STM32开发板的光盘中为大家提供了丰富的例程代码，可以参考。您一定可以修改出更精彩的例程！  步骤五，Ucos-II的移植，是否需要试一下？    x7 Y\* A% c# |4 F% @9 x  恭喜你，至此，你已经可以自如进行独立的开发了。最后一步，给自己一个目标（项目），把它实现出来！  " ]# |. b6 O; S/ }8 w2 R' k\* D3 ~  再次强调，以上只是一个学习STM32的思路，供大家参考。下面列出了前面关键的步骤，希望大家能尽快入门。后续的每个模块的编程，请参考我们的芯达STM32入门系列教程。  3.2.1第1步：熟悉调试软件  5 }+ D\* F! e% t7 g- q' {  对初学者来说，我们至少需要安装两个软件：J-Link驱动软件、MDK（就是原来的Keil）软件。这两个软件，用《乡村爱情》里的一句话“必须的！”  如何验证自己已经熟悉调试软件的操作了呢？很简单， STM32开发板光盘里附带了很多HEX格式的文件，可以选择一些HEX文件，来观察运行结果。  7 `/ Q: c' y: n' G\* L. G  比如LED灯的例程HEX文件，下载到芯达STM32开发板中，观察LED灯的闪烁情况。灯在闪烁，就说明你刚才的操作已经把HEX文件烧写到闪存中了。  $ ~' z, V; }. u( @; H( E  该步骤要达到的目标：熟悉调试软件，如烧写HEX出现问题，可简单判别问题所在，并独立解决。  3.2.2第2步：GPIO编程  这是第一次接触固件库的编程，一定要硬着头皮去了解固件库。建议大家尽量去用固件库。而不是避开固件库自己写代码——这样只能在学习中才会发生。实际的项目中，代码成百上千个，如何一个一个自己写？调用固件库中的函数来完成，才是王者之道。  GPIO本身的编程实际上很简单：  1、设置GPIO口的引脚为输入或者输出模式。我们在进行点灯代码的时候，一般设置为推挽输出模式。  + I- D) }6 c$ o\* v  2、操作寄存器，往寄存器里置1或者清零操作——这个步骤，固件库已经提供了专门的GPIO\_SetBits函数和GPIO\_ResetBits函数，我们只要去调用即可实现对IO口的置1和清零。  % w% X9 j& ~- W7 Z6 D  3、实现多种花样的LED闪动，使得自己熟悉GPIO的编程过程。但是在调用固件库的时候，我们还需要有GPIO结构声明，结构成员初始化，系统时钟配置等等注意事项。该步骤要达到的目标：熟悉调试软件，如烧写HEX出现问题，可简单判别问题所在，并独立解决。    U/ ~" Y0 ~, n  3.2.3开始全新的STM32深入研究  经过以上调试软件的熟悉和GPIO口的编程调试后，相信您已经对STM32有一定的了解。至少知道如何利用STM32的固件库去写一个代码。OK，下面我们将开始全新的STM32深入研究。在这个阶段，将要接触到串口编程、TFT液晶屏驱动编程、定时器编程、串行外设接口SPI编程、存储器编程、SD卡与文件系统移植、USB读写、UCOS移植等，有精力还可以研究其他外设。 |