1. Don't be greedy for single chip

There are many kinds of single-chip manufacturers, such as PIC singlechip, there are as many as 500 models!Obviously it is impossible to learn all kinds of SCM, and it is not necessary.In fact, it's enough to have one or two single-chip microphones (STC, AVR, STM32, etc.).We have done more than 100 single-chip projects over the years, mainly using three single chip microcomputers.

For beginners, it is recommended to choose a single chip microcomputer (of course, the basic 51 MCU is must-learn), and learn it and use it.Even if other kinds of singlechip need to be used in the future, due to the solid foundation, it can be learned quickly because the single-chip function is very similar after all.As to which MCU, it is best to consult experienced teachers or engineers.

2. It is better to learn than to learn more

Some learning board called "free" with several G ~ dozens of G learning materials, not to mention how these massive amounts of data, rather go through these information is enough to spend months and last year's time.After reading, only to understand a general idea,you have forgotten all the time .Rather than just calm down, to understand each point, understand every detail.I learned C only using the source code of a real product and a C grammar book.Do a thorough study of every line of programs, and you can't understand the grammar of the book (now it's

more convenient to have the Internet).Don't miss every knowledge point.After learning the source code, I'll write the utility in a minute.The advantage of this learning method is that it is not only easy to understand, but also to use it in practical application scenarios.

3. According to the need to learn

There are many kinds of single-chip microcomputers, and the range of applications is wide, and the peripheral devices are numerous. It is impossible to learn all the time.Except if you have plenty of time.Learning by demand in practice is an efficient way to learn,because you remember when you learn.And a year of engineers and three years of engineers learning a functional module, obviously three years of engineers to learn quickly.(unity of knowledge and practice!)

4. Ability is more important than knowledge

In the process of work or learning, it is a reflection of the ability of the engineer to solve the problems quickly.This ability has to do with individual talent, and more importantly, nurture.As long as you are diligent in practice, thinking, and accumulating, and not letting go of every question or problem, your ability will continue to improve.

5. Experience is more important than knowledge

Experience is not something you can learn by looking at dozens of gigabytes of data.There are two ways to gain experience: to accumulate and worship the master.Accumulation is necessary, but also hard and

slow.It's a great way to be a master, but it depends on the fate.

6. Do are as important as books

See more information, do not go to work, can only be a little knowledge, the time has long not need to return to the teacher.

Wang yangming, an ancient Chinese philosopher, thought that the "unity of knowledge and practice" was perfect for guiding the study of electronic engineers.Practicing on the learning board, doing experiments, no pressure, just dabble, not really doing the work.The real thing is to undertake development projects and make actual products.The real thing is to undertake development projects and make actual products.It's time to see the real chapter, and you'll feel stressed and motivated, which is the best way to learn.

7. Cultivate good habits

A lot of beginner programmers use ABC to define variables when they come up,and don't like commenting and writing documents,and Save the day (and I used to),and think it's efficient,rather they have planted many hidden dangers.The new driver likes to drive fast compete with other cars,and think it's too bad for the old driver to drive too much slowly,it's all about rookie.Even minor accidents can take a few days.In peacetime, you can earn a few seconds, which one is it.

Good habits can save lives.I have developed a habit of not using my hands to contact the chips and any metal when taking the circuit

board.One time I was studying and thinking about a circuit board for more than an hour.Later, I realized that the power source had been forgotten so the board is charged,and it's all 220V!It's a good habit to save me.

8. Understand reuse

Reuse is something that this project does and other items can be referenced.Reusable knowledge includes software, circuits, architectures, and so on.The benefit of reuse is to improve development efficiency and improve product quality.This requires the engineer to consciously increase the general design in the development process, and pay attention to the accumulation of knowledge.

9. Learn the basic knowledge of operating system

There is no proper operating system for single-chip microcomputers, so you don't have to operate the system. You can write it by yourself.Even so, learning the basics of a little operating system,for example, real-time concepts, multichannel systems, multi-tasking, atomic operations, reentrant functions, etc., are very helpful for software design.

And, of course, if you have the right operating system, I suggest you try to use it.The benefit is to reduce the workload, reduce errors, and facilitate migration.Recently, we introduced the real-time operating system of TreeOS 1.0, which is used for single-chip microcomputer, and beginners can also quickly grasp it, so please try it.

WeiboID：@TreeOS Mrs Lin