The most important programming languages ??in the Internet of Things era

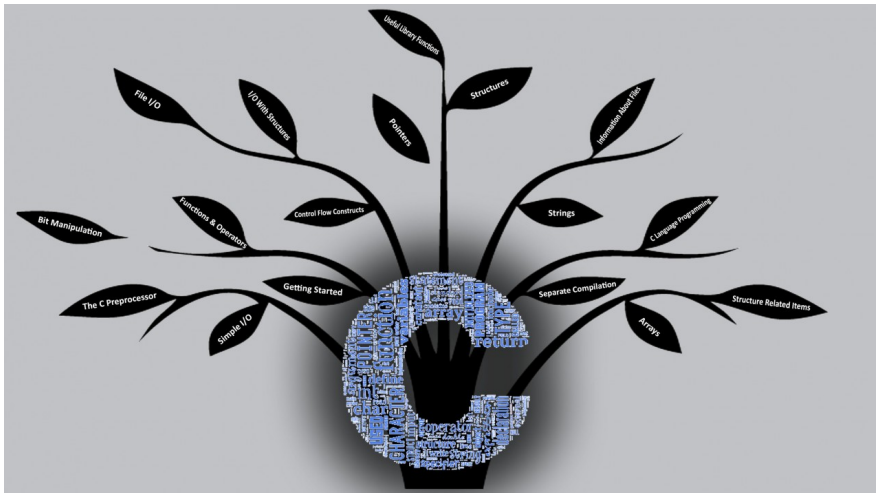
C, C ++, C # or JavaScript are some of the most useful programming languages ??for developing embedded systems.

Internet of Things (IoT) , also known as the ***"Everything of the World"*** , is a completely new platform for engineers and designers. However, one of the few things that can't be changed when we move into this era is programming languages. In order to be successful in developing applications that follow the IoT trend, on the one hand, code employees need to grasp the most core changes, on the other hand, to take advantage of the advantages of the established languages. submit in each project.

Recently, ReadWrite page - the leading communication platform for Internet of Things trend and "connected world" has shared a very interesting article about the most important programming languages ??in this era. If you are looking for a career in information technology, find out now and start learning.

The most popular languages

According to ReadWrite, C, C ++, Java and C # are the most popular choices. C is considered a very useful language for programming embedded systems because it does not require much processing capability. C ++ is a good alternative if the IoT device requires more complex tasks - such as temperature regulators or smart toasters - than warm or thermal probes. Meanwhile, Java is compatible with devices that require Interface and calculator because it is more flexible, lighter (because of high-level language) and is also very popular in teaching.



C is a structured programming language, allowing to create an application consisting of many functional modules (files). Each module contains many functions and these functions can be closed in the module containing them or can be accessed by any other place outside the module containing it. This is a language that is not rigorously tested, meaning that the translation program is incapable and never checks the type, at the same time, you can also assign strings to integer variables . C matches those Professional programmers, able to determine the correctness of the code they have written.

C ++ is an extension of C, providing some additional capabilities to enable object-oriented programming, while possessing more features and complexity than C. C ++ fully supports object-oriented programming, including 4 such pillar features are Encapsulation, Data hiding, Inheritance and Polymorphism. In addition, this programming language also supports classes that allow writers to specify objects that constitute the application. However, the level of object support of C ++ is weak, not clear and easy to make programmers make mistakes in the process of writing code. In general, C ++ is suitable for people who are used to C or those who only know object-oriented programming or who want to combine two structured and object-oriented programming schools.

C # is Microsoft's platform and strategy language. This is a pure and purely object-oriented language, capable of implementing most of the properties of object-oriented models like Java. C # can be translated into machine code to run on .Net platform - providing many powerful, rich and diverse objects to solve many of the problems that often occur in programming. Thus, if you write code in C # language, you spend very little cost, time, the application runs reliably - thanks in part to the object-oriented nature of C #, partly because the application primarily uses objects written by Microsoft (or a third party).

Python is suitable for programmers who require simplicity. In addition, Python can be extended for use in industry or data analysis in the financial sector. For any application that requires large data retrieval capabilities, Python is a very potential candidate and powerful enough for application in embedded platforms.

In addition to the above options, **JavaScript** is also a popular programming language. This is a scripting language based on existing or self-defined development objects and widely used in web applications.

Some other options

If you don't like using C, you can choose from some low-level programming languages.**B #** is one of them. Basically, B # is designed for the development of large systems that are scaled to fit an embedded platform, much like C ++, but omitted some unnecessary features and added properties. Real-time control. If the IoT project is based on an embedded platform that is not too large and complex, B # is the language you can consider.

Assembly is considered the "prince" of low-level and very popular programming languages ??by microcontroller programming experts. In essence, this language is the relationship between programming language and device structure, each assembly is specially designed for a different computer structure, compact code, fast operation and takes up less memory . However, the drawback is that Assembly lacks the facilities to minimize programming risks. If the code doesn't work or the new processor doesn't accept the code, you're too unlucky!

Google also developed a new programming language called **Go** , available on a variety of processors and platforms. This is a language derived from C but improved to overcome some of the previous limitations in harnessing the power of new-generation multi-core processors and hardware. In addition to Go, Google has also tested many other programming languages, including **Weave** for IoT devices. If you get support from developers in and out of Mountain View, it will sooner or later become more prevalent.