

C Versus EMBEDDED C: Unlock 10 Key Differences

C and Embedded C are both programming languages that are used to create software. However, there are few key differences between the two.

C is a general-purpose programming language that is used to create a wide variety of software, such as games, operating systems, compilers, databases, server applications, network programs, device drivers, etc.

Embedded C is a subset of C that is designed specifically for embedded systems. Embedded systems are mostly small, low-power devices which are built for specific purposes, and have limited memory and processing power. Embedded C takes advantage of the features of these devices to create software that is efficient and reliable. Embedded C is a more specialized language that is designed for systems with limited memory and processing power.

The important differences between C and Embedded C are described in the following section:

1. **Memory usage:** C programs consume more memory as they are not designed for systems with limited memory. They typically use a lot of memory, especially if they are using large data structures or complex algorithms whereas Embedded C programs are designed for systems with limited memory and must be careful to use memory efficiently.
2. **Processing power:** C programs are written for systems with large processing power. They can be slow if they are using complex algorithms or performing calculations. Embedded C programs are designed for systems with limited processing power which should be used efficiently.
3. **Compilers:** C compilers are available for a wide variety of platforms or operating systems which means that C programs can be compiled and run on a variety of computers. Embedded C compilers are only available for specific platforms which means that embedded C programs can only be compiled and run on the specific platform for which that compiler was created.
4. **Standard Libraries:** C has a large standard library that provides a wide range of functions. This library can be used to perform a variety of tasks, such as input/output, string manipulation, and mathematical operations. Embedded C has a smaller standard library that provides only the most essential functions.
5. **Scalability:** C programs are easily scalable which means that new lines of code can be added without any limitations of size of the program whereas Embedded C programs are not scalable due to limited memory and processing power of the hardware for which the Embedded C code is written.

6. **Hardware Dependency:** C programs are not dependent on the hardware or the operating system on which the compiler runs, whereas Embedded C programs are completely dependent on the hardware for which the code is written.
7. **Task Execution:** C programs support multiple task executions at a time whereas Embedded C programs do not support multiple task executions at a time.
8. **System Support:** C supports both software and hardware systems whereas Embedded C supports only hardware systems.
9. **Time Requirement:** C programs consume more time for development and deployment whereas Embedded C programs require less time for development and deployment.
10. **Programming style:** C programs are typically written in a free-format style. This means that the programmer has a lot of freedom in how they format the code. Embedded C programs are typically written in a structured style. This means that the code is organized in a way that makes it easier to read and understand.

Here is a table that summarizes the key differences between C and Embedded C:

Feature	C	Embedded C
Memory usage	C is not designed for systems with limited memory.	Embedded C is designed for systems with limited memory.
Processing power	C is not designed for systems with limited processing power.	Embedded C is designed for systems with limited processing power.
Compilers	C compilers are typically available for a wide variety of platforms.	Embedded C compilers are typically only available for specific platforms.

Libraries	C has a large standard library that provides a wide range of functions.	Embedded C has a smaller standard library that provides only the most essential functions.
Scalability	C programs can be scaled efficiently	Embedded C programs cannot be scaled easily
Hardware Dependency	C does not depend on the hardware being used	In Embedded C when changes are made in hardware appropriate changes need to be made in software and vice versa
Task Execution	C supports multiple task executions at a time	Embedded C does not support multiple task executions at a time.
System Support	C supports both software and hardware systems	Embedded C supports only hardware systems
Programming style	C programs are written in a free-format style.	Embedded C programs are written in a structured style.
Time Requirement	C programs consume more time for development and deployment	Embedded C programs consume less time for development and deployment

To summarize, C is used to program complex systems and software applications which makes its code complex, large and difficult to maintain, whereas embedded C is used to develop firmware and low level systems and provides greater optimization, performance, speed, and reliability.