Volatile

to order or tell someone to do something, especially in a formal way

Volatile is a type qualifier in 'C' used with variables to instruct the complier not to invoke any optimization on the variable operation.

↳ to request help from someone, especially a god, when you want to improve a situation.

permission or agreement

It tells the complier that the value of the variable may change at any time with our without the programmer's consent. So, the complier turns of optimizing the read-write operations on variables which are declared using volatile keyword.

If we have some of variables which are not using inside of the code block. Optimization understand these things and avoid the waste of memory. But volatile says "in every condition, check this variables. Are they changing or not and define this variables". } important part of volatile.

Syntax of using 'volatile"

CASE 1:

uint8_t volatile my_data
volatile uint8_t my_data

both are identical

"likely to change suddenly and unexpectedly, especially by getting worse"

→ my_data is a volatile variable of type unsigned integer_8

CASE 2:

uint8_t volatile *pStatusReg;

pStatus is a non-volatile pointer, pointing to volatile data of type unsigned integer_8.

CASE 3:

uint8_t *volatile pStatusReg;

pStatusReg is a volatile pointer, pointing to non-volatile data of type unsigned integer_8.

CASE 4:

uint8_t volatile *volatile pStatusReg;

pStatusReg is a volatile pointer, pointing to volatile data of type unsigned integer_8.

} Rarely used

When to use 'Volatile' qualifier?

Use volatile when your code is dealing with below scenarios

1. Memory mapped peripheral registers of the microcontrollers.
2. Multiple task accessing global variables (read/write) in an RTOS multithreaded apllication.
3. When a global variable is used to share data between the main code and an ISR code.