C

Qualifiers

Types

Const

Volatile

Static

Extern

Register

Const

→ The **const** qualifier makes the variable unchangeable.

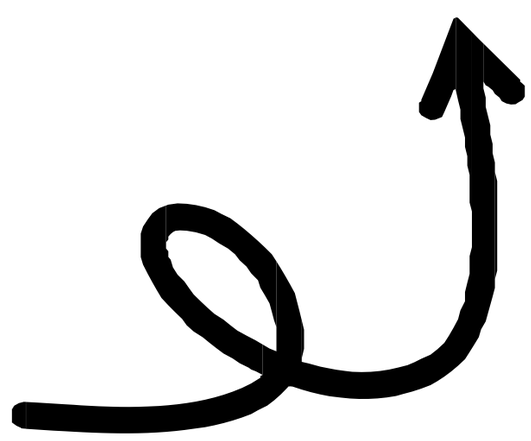
→ It especially puts the data into the **read-only** section (.rodata) of the **FLASH** memory.

Example

```
//constant variable
//value cannot be changed
const int number;
```

Constant variable

```
//constant pointer
//address of pValue cannot be changed
int* const pValue;
```



Constant pointer, means address where it points cannot be changed.

Example

```
//constant function parameters with
//constant return value
const int add(const int number)
{
    //write code here...

} //eo add::
```

A function declaration with constant return type and constant parameter value.

Volatile

→ The **volatile** qualifier warns the compiler of a frequent change of value of variable.

→ It prevents incorrect **optimization** of the compiler.

→ Highly used in **Embedded systems**.

Example

```c
typedef struct
{
    volatile uint32_t SR;
    volatile uint32_t DR;
    volatile uint32_t BRR;
    volatile uint32_t CR1;
    volatile uint32_t CR2;
    volatile uint32_t CR3;
    volatile uint32_t GTPR;
}USART_RegDef_t;
```

USART register structure with volatile SFR* definition to tell the compiler not to optimize as it may contain important data.

**SFR - Special Function Register*

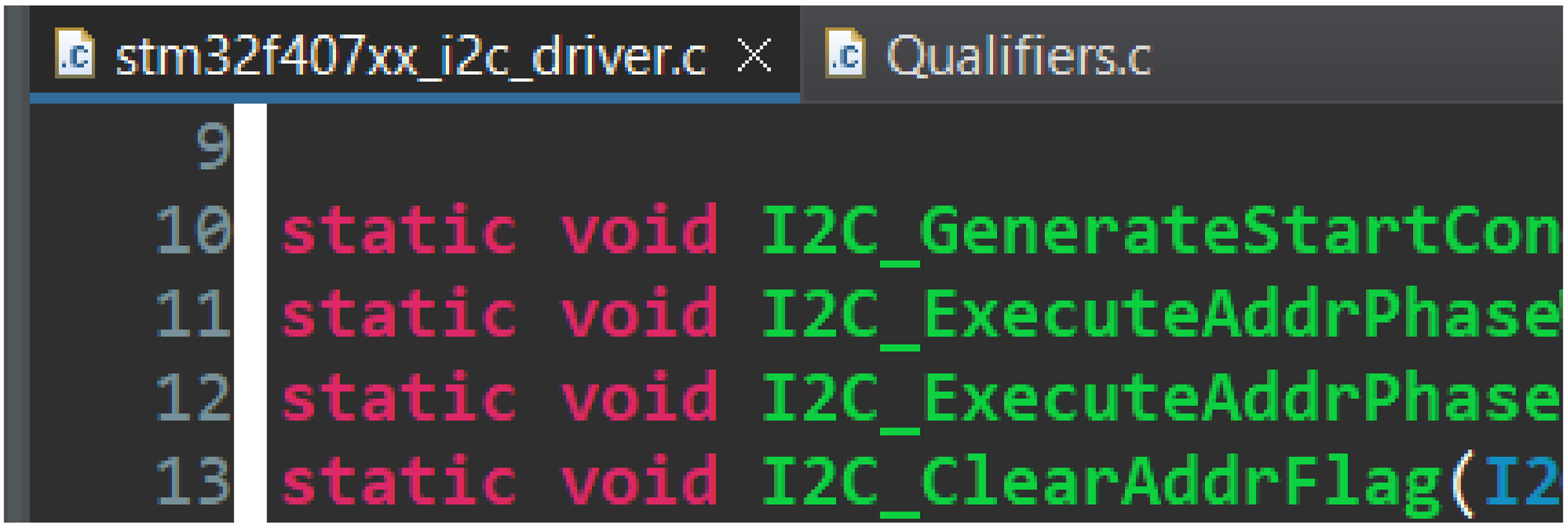
Static

→ The **static** qualifier mainly deals with the **scope** of **variables & functions**.

→ A **function** defined with **static** is only visible within the file.

→ A **variable** defined with **static** retain its value during the function calls.

Example



Because of ‘static’ definition I2C related functions can only be visible to i2c_driver.c current file.

Extern

→ The **extern** qualifier is a type of opposite to **static**.

→ It makes functions and variables visible across the different files.

→ Functions in C are **implicitly extern**, means they can be used across the files without extern.

Register

→ The **register** qualifier is used to store variables in **CPU** register instead of RAM of FLASH for → faster access.

However, not used nowadays as optimization has become automatic.

Applications

- It modifies the behavior of variables with specific specific characteristics.
- The specific qualifiers are used for special memory allocation.
- They are best used in Embedded C to optimize for the code and memory.

THANK YOU!

Follow for more content!!!