

8th Task in Embedded Systems

◆ What is a macro in C, and how is it defined?

A **macro** is a preprocessor directive used to define constants or code snippets before compilation. It is defined using **#define**.

```
#define PI 3.14  
  
#define SQUARE(x) ((x)*(x))
```

◆ What is the difference between macros and functions?

Macros:

- Faster because there is no function call overhead.
- Do not support type checking.
- Can lead to errors that are hard to debug due to lack of type safety.

Functions:

- Actual compiled blocks of code with parameters and return types.
- Safer with type checking.
- Easier to debug and maintain, but have some overhead due to function calls.

◆ What do **#ifdef**, **#ifndef**, and **#endif** do?

These are conditional compilation directives:

- **#ifdef** MACRO → compile code if MACRO is defined.
- **#ifndef** MACRO → compile code if NOT defined.
- **#endif** → marks the end of the conditional block.

◆ What does **malloc()** do, and what type does it return?

malloc(size) dynamically allocates **raw memory** of given size (in bytes) and returns a pointer of type void.

It must be **typecast** to appropriate type:

```
int *ptr = (int *)malloc(n * sizeof(int));
```

◆ What is the difference between malloc, calloc, and realloc?

Function	Use	Initialization	Use Case
malloc	Allocate memory	No (garbage values)	Fast allocation
calloc	Allocate & zero-initialize	Yes (to 0)	Safe default
realloc	Resize previously allocated memory	Keeps old data	Dynamic growth

◆ Why must we always call free() after dynamic allocation?

To **prevent memory leaks**. Memory from malloc, calloc, or realloc stays allocated until manually released using **free()**.

◆ What is a header guard, and what problem does it solve?

A header guard prevents a header file from being included multiple times, which causes redefinition errors.

◆ What is the typical format of a header guard?

```
#ifndef HEADER_NAME_H
#define HEADER_NAME_H
// Header content
#endif // HEADER_NAME_H
```

◆ How does the preprocessor handle nested includes?

The preprocessor expands #include directives recursively. Header guards ensure **each file is only included once**, avoiding duplicate definitions even if included from multiple files.