IEEE RAS
ROBOTICS & AUTOMATION SOCIETY
Zagazig student chapter

TASK 8

Embedded Systems Track

DATE :

6/5/2025

PRESENTED BY :

MOHAMED KADRY

Table Of Content

1. What is a macro in C, and how is it defined?
2. What is the difference between macros and functions?
3. What do #ifdef, #ifndef, and #endif do?
4. What does malloc() do, and what type does it return?
5. What is the difference between malloc, calloc, and realloc?
6. Why must we always call free() after dynamic allocation?
7. What is a header guard, and what problem does it solve?
8. What is the typical format of a header guard?
9. How does the preprocessor handle nested includes?

1. Macros in C

Definition:

Preprocessor directives (#define) that perform text substitution before compilation.

Key Properties:

- No type checking (raw text replacement)
- No scope rules (global substitution)
- Can be simple constants or function-like
- Faster than functions (no call overhead) but riskier

Example:

```
#define PI 3.14                    // Constant macro
#define MAX(a,b) ((a) > (b) ? (a) : (b))   // Function-like macro
```

2. Conditional Compilation

Directives:

- #ifdef MACRO: Includes code if MACRO is defined.
- #ifndef MACRO: Includes code if MACRO is not defined.
- #endif: Closes the conditional block.

Use Case:

Platform-specific code, debug modes, or feature toggles.

Example:

```
#ifdef DEBUG
printf("Debug mode active!");
#endif
```

3. Memory Allocation Functions

Function	Behavior	ReturnType	Initialization
malloc()	Allocates raw memory	void*	Uninitialized
calloc()	Allocates + zeroes memory	void*	Zero-filled
realloc()	Resizes existing allocation	void*	Preserves/extends data

Critical Rule:

Always pair with free() to prevent memory leaks.

Example:

```
int* arr = malloc(10 * sizeof(int));  // Allocate
free(arr);                            // Release
```

4. Header Guards

Purpose: Prevent duplicate inclusions of header files.

Mechanism:

```
#ifndef MY_HEADER_H  // If not defined...
#define MY_HEADER_H  // Define it now
// Header contents
#endif               // End guard
```

5. Preprocessor & Nested Includes

Behavior:

1. Processes #include directives recursively.
2. Header guards ensure each file is included once per translation unit.
3. Follows last-in-first-out (LIFO) order.

Example Flow:

```
main.c → #include "utils.h" → #include "math.h"
```