

# MikroElektronika books

Book: [Microcontroladores PIC – Programación en C con ejemplos](#)

☰  
Table of Contents

## 2.9 Características Principales del Preprocesador

El preprocesador es un programa que procesa el código antes de que pase por el compilador. Funciona bajo el control de las líneas de comando del preprocesador denominadas directivas. Las directivas del preprocesador se colocan en el código fuente, normalmente en el principio del archivo. Antes de pasar por el compilador, el código fuente se examina por el preprocesador que detecta y ejecuta todas las directivas del preprocesador. Las directivas del preprocesador siguen a una regla de la sintaxis especial, empiezan por un símbolo ‘#’ y no requieren ningún punto y coma al final (;).

### DIRECTIVAS DEL PREPROCESADOR

La siguiente tabla contiene un conjunto de directivas del preprocesador frecuentemente utilizadas: Las directivas del preprocesador se

⏪

DIRECTIVAS	FUNCIONES
#include	Define una sustitución de macro
#undef	Quita una definición de nombre de macro
#define	Especifica un archivo a ser incluido
#ifdef	Prueba para definición de macro
#endif	Especificar el final de #if
#ifndef	Prueba si una macro no está definida
#if	Prueba las condiciones de compilar
#else	Especifica alternativas cuando la prueba de #if falla
#elif	Especifica alternativas cuando más de dos condiciones se necesitan

⏩

pueden dividir en tres categorías:

- Definiciones de macro
- Inclusiones de archivos
- Control de compilación

Ahora, vamos a presentar sólo las directivas del preprocesador utilizadas con más frecuencia. Sin embargo, no es necesario saber todas ellas para programar microcontroladores. Sólo tenga en cuenta que el preprocesador es una herramienta muy poderosa para los programadores avanzados en C, especialmente para el control de compilación.

### DIRECTIVAS DEL PREPROCESADOR PARA DEFINIR MACROS

Por medio de los macros es posible definir las constantes y ejecutar funciones básicas. Una sustitución de macro es un proceso en el que un identificador del programa se sustituye por una cadena predefinida. El preprocesador sustituye cada ocurrencia del identificador en el código fuente por una cadena. Después de la sustitución, el código será compilado normalmente. Esto significa que el código sustituido debe respetar la sintaxis del mikroC. La acción se realiza por medio de la directiva '#define'.

```
#define PI 3.14159 // Sustitución simple, PI será sustituido por
                  // el valor 3.14159 en todas las partes del programa
```

También puede utilizar los parámetros para realizar substituciones más complejas:

```
#define VOLUMEN (D,H) (((D/2)*(D/2)*PI))*H // Macro con parámetros
```

Entonces, en el código, la siguiente sentencia:

```
Tanque_1 = VOLUMEN (Diámetro,altura);
```

será sustituida por:

```
Tanque_1 = (((Diámetro/2)*(Diámetro/2)*PI)*altura;
```

Por medio de la directiva #undef es posible quitar una definición de nombre de macro. Así se especifica que la substitución que se ha definido anteriormente ya no va ocurrir en el siguiente código. Esto es útil cuando usted quiere restringir la definición sólo a una parte particular del programa.

```
#undef TANQUE // Quitar la definición del macro VOLUMEN
```

### INCLUSIÓN DE ARCHIVOS

La directiva de preprocesador #include copia un archivo específico en el código fuente. El código incluido debe observar la sintaxis de C para ser compilado correctamente. Hay dos formas de escribir estas directivas. En el primer ejemplo, sólo el nombre de archivo se especifica, así que el preprocesador lo buscará dentro del archivo include. En el segundo ejemplo, se especifica la ruta entera, así que el archivo estará directamente incluido (este método es más rápido).

```
#include <nombre_de_archivo>           // Se especifica sólo el nombre del archivo
#include "C:\Ruta\nombre_de_archivo.h" // Se especifica la localidad
                                         // exacta del archivo
```

SUBSCRIBE TO



newsletter



github



linkedin



facebook



youtube



instagram



twitter

JOIN US

Careers

Make a Click

Internship

#### COMPANY

About us    Contact  
Leadership    PressKit  
Distributors    Timeline  
Terms

#### TOOLCHAINS

PIC    dsPIC  
PIC32    ARM  
AVR    FT90X  
8051    PSOC  
CEC

#### RESOURCES

mikroBUS™    mikroSDK  
Click Cloud    Premium TS  
Hexiwear™    Libstock™  
eBooks    Outlet  
Legacy