

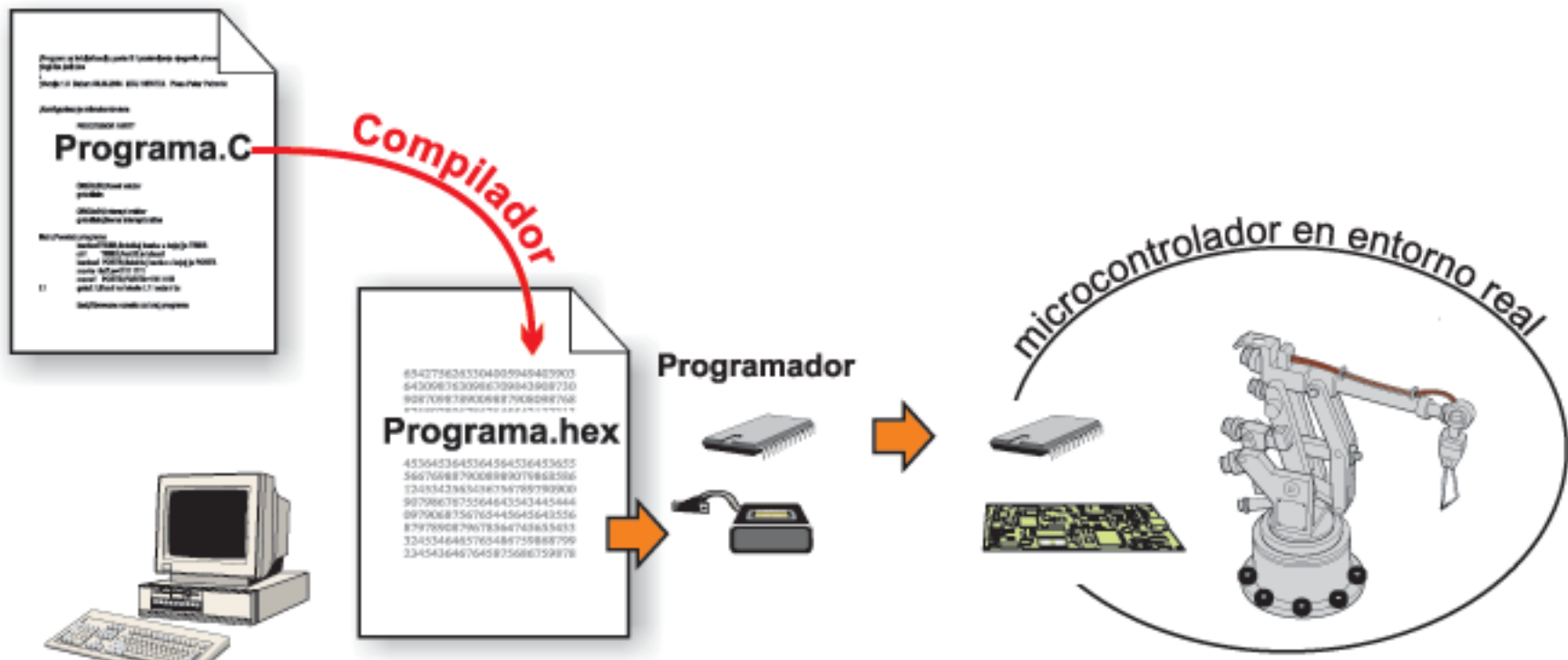
MikroElektronika books

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

Table of Contents

2.2 Características Principales del Mikroc

A continuación vamos a presentar a los elementos principales del lenguaje mikroC desarrollado por Mikroelektronika. Este lenguaje es muy similar al C estándar, no obstante en determinados aspectos difiere del ANSI estándar en algunas características. Algunas de estas diferencias se refieren a las mejoras, destinadas a facilitar la programación de los microcontroladores PIC, mientras que las demás son la consecuencia de la limitación de la arquitectura del hardware de los PIC. Aquí vamos a presentar características específicas del lenguaje mikroC en la programación de los microcontroladores PIC. El término C se utilizará para referirse a las características comunes de los lenguajes C y mikroC.



Este libro describe una aplicación muy concreta del lenguaje de programación C utilizado en el compilador mikroC PRO for PIC. En este caso, el compilador se utiliza para la programación de los microcontroladores PIC.

FASES DE COMPILACIÓN

El proceso de compilación consiste en varios pasos y se ejecuta automáticamente por el compilador. Por con, un conocimiento básico del funcionamiento puede ser útil para entender el concepto del lenguaje mikroC. El archivo fuente contiene el código en mikroC que usted escribe para programar el microcontrolador. El preprocesador se utiliza automáticamente por el compilador al iniciarse el proceso de la compilación. El compilador busca las directivas del preprocesador (que siempre empiezan por '#') dentro del código y modifica el código fuente de acuerdo con las directivas. En esta fase se llevan a cabo inclusión de archivos, definición de constantes y macros etc, lo que facilita el proceso. Más tarde vamos a describir estas directivas en detalle. El **analizador sintáctico** (parser) elimina toda la información inútil del código (comentarios, espacios en blanco). Luego, el**compilador** traduce el código a un archivo binario denominado archivo .mcl. El **enlazador** (linker) recupera toda la información requerida para ejecutar el programa de los archivos externos y la agrupa en un solo archivo (.dbg). Además, un proyecto puede contener más de un archivo fuente y el programador puede utilizar funciones predefinidas y agrupadas dentro de los archivos denominados librerías. Por último, el **generador .hex** produce un archivo .hex. Es el archivo que se va a cargar en el microcontrolador.



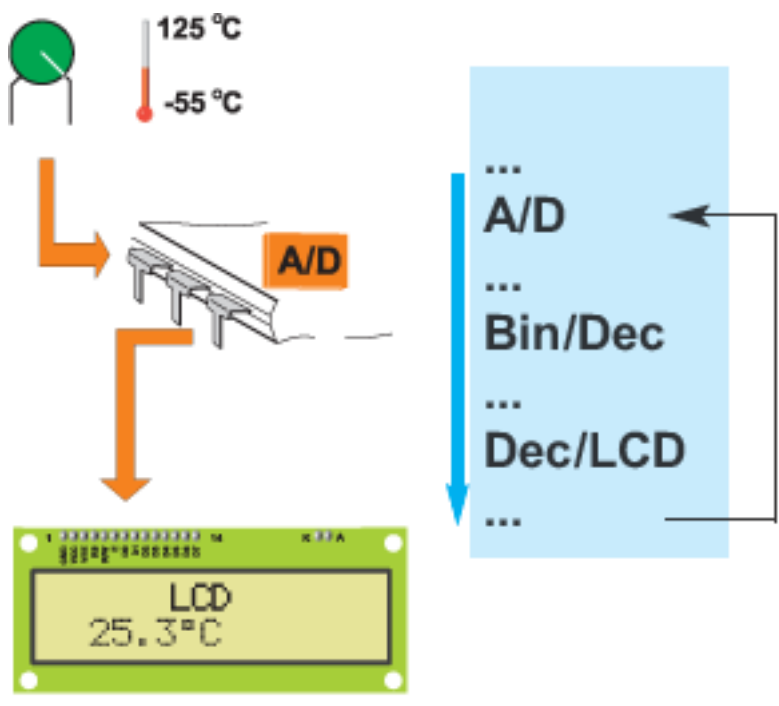
El proceso entero de la compilación que incluye todos los pasos anteriormente descritos se le denomina “building”.

ESTRUCTURA DE PROGRAMA

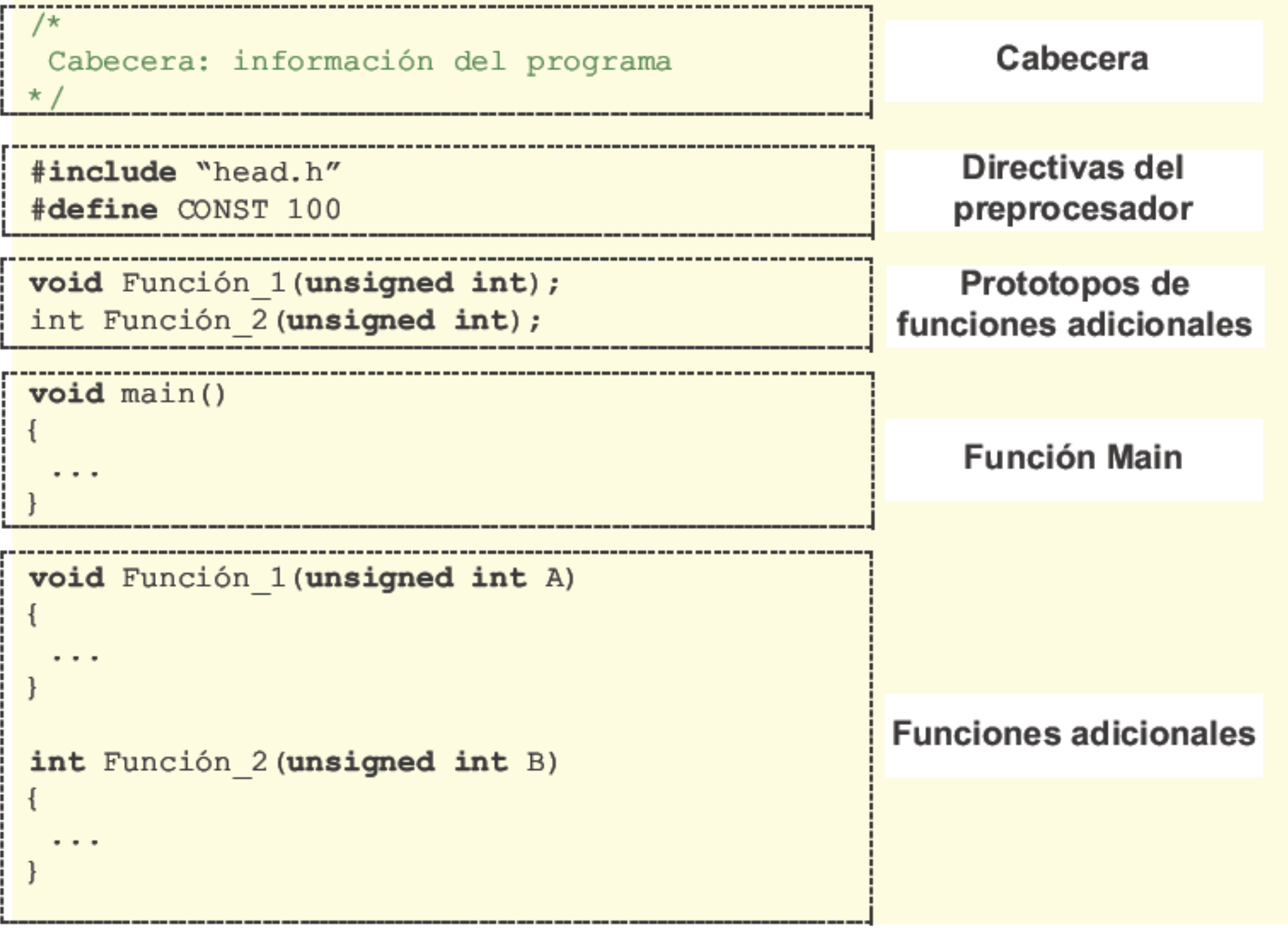
La idea principal de escribir un programa en C es de “romper” un problema mayor en varios trozos más pequeños. Supongamos que es necesario escribir un programa para el microcontrolador para medir la temperatura y visualizar los resultados en un LCD. El proceso de medición se realiza por un sensor que convierte temperatura en voltaje. El microcontrolador utiliza el convertidor A/D para convertir este voltaje (valor analógico) en un número (valor digital) que luego se envía al LCD por medio de varios conductores. En consecuencia, el programa se divide en cuatro partes, de las que cada una corresponde a una acción específica:

1. Activar y configurar el convertidor A/D incorporado;
2. Medir el valor analógico;
3. Calcular temperatura; y
4. Enviar los datos en el formato apropiado al LCD;

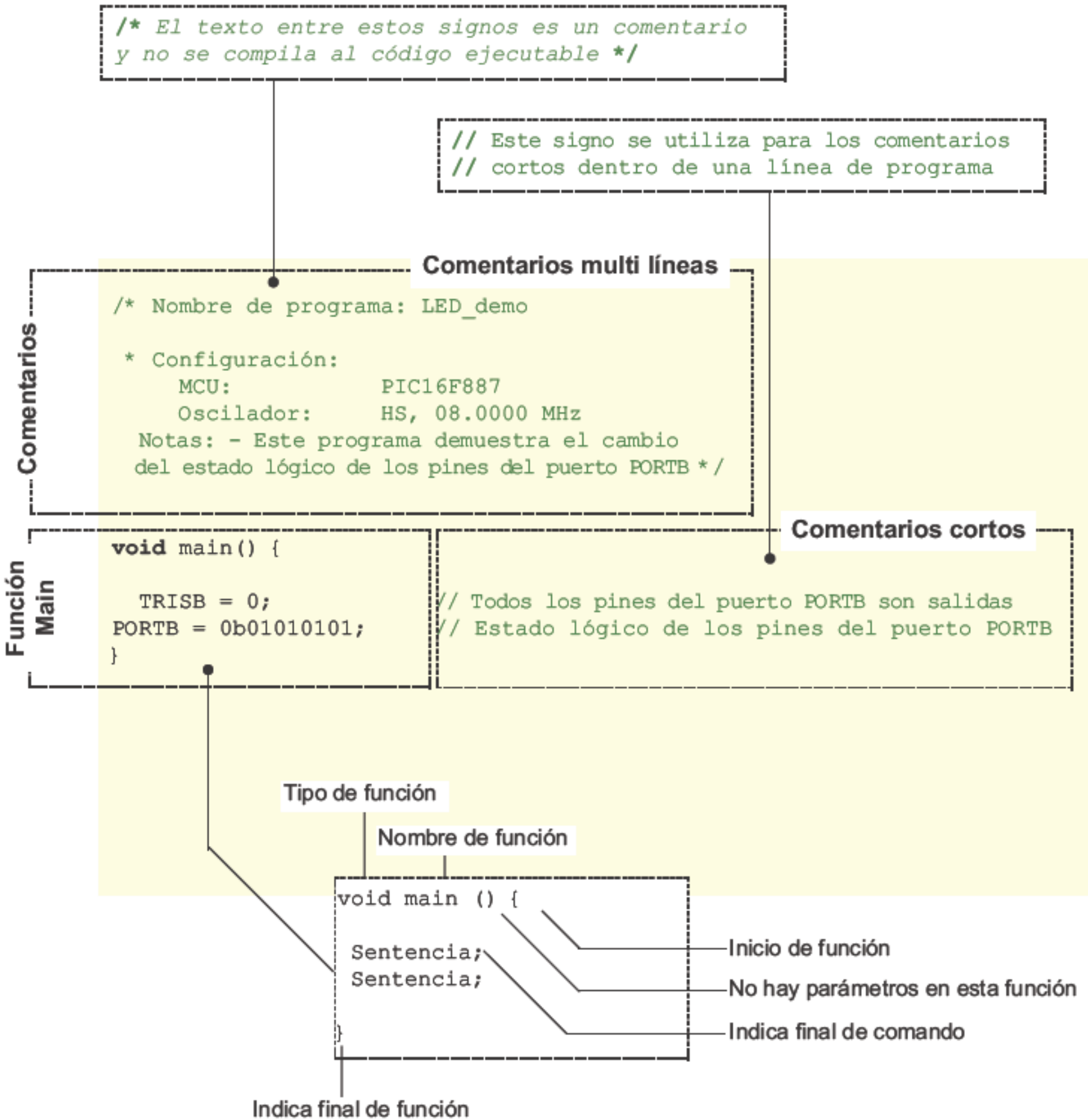
Los lenguajes de programación de alto nivel como es C le permiten solucionar este problema con facilidad al escribir cuatro funciones que se ejecutarán cíclicamente sin parar. La idea general es de dividir el problema en varios trozos, de los que cada uno se puede escribir como una sola función. Todos los programas escritos en mikroC contienen por lo menos una función llamada main() que encierra entre llaves {} las sentencias a ser ejecutadas. Esto es la primera función a ser ejecutada al iniciarse la ejecución de programa. Las otras funciones se pueden llamar dentro de la función main. En otras palabras, podemos decir que la función main() es obligatoria, mientras que las demás son opcionales. Si todavía no ha escrito un programa en C, es probable que todo le resulte confuso. No se preocupe, acéptelo tal como es por el momento y más tarde entenderá la sintaxis.



¡Y ahora, su primer programa 'real'! La figura muestra la estructura de programa, señalando las partes en las que consiste.



La manera de escribir el código en C es muy importante. Por ejemplo, C difiere entre minúsculas y mayúsculas, así que la función main() no se puede escribir MAIN() o Main(). Además, note que dos líneas del código dentro de la función terminan con un punto y coma. En C todas las sentencias deben terminar con un punto y coma ';', así el compilador puede aislarlas y traducirlas a código máquina.



COMENTARIOS

Los comentarios son las partes del programa utilizados para aclarar las instrucciones de programa o para proporcionar más información al respecto. El compilador no hace caso a los comentarios y no los compila al código ejecutable. Dicho de manera sencilla, el compilador es capaz de reconocer los caracteres especiales utilizados para designar dónde los comentarios comienzan y terminan y no hace nada de caso al texto entre ellos durante la compilación. Hay dos tipos de tales caracteres. Unos designan los comentarios largos que ocupan varias líneas de programa marcados por la secuencia especial /*...*/, mientras que otros designan los comentarios cortos que caben en una sola línea //. Aunque los comentarios no pueden afectar a la ejecución de programa, son tan importantes como cualquier otra parte de programa. Aquí está el porqué... Con frecuencia es necesario mejorar, modificar, actualizar, simplificar un programa... No es posible interpretar incluso los programas simples sin utilizar los comentarios.

SUBSCRIBE TO



JOIN US

Careers

Make a Click

Internship

COMPANY

About us

Contact

Leadership

PressKit

Distributors

Timeline

Terms

TOOLCHAINS

PICdsPIC

PIC32ARM

AVRFT90X

8051PSOC

CEC

RESOURCES

mikroBUS™mikroSDK

Click CloudPremium TS

Hexiwear™Libstock™

eBooksOutlet

Legacy