

# Creación y uso de bibliotecas en proyectos realizados con el IDE Keil uVision

<b>Apellidos, nombre</b>	Capella Hernández, Juan Vicente ( <a href="mailto:jcapella@disca.upv.es">jcapella@disca.upv.es</a> )
<b>Departamento</b>	Depto. de Informática de Sistemas y Computadores (DISCA)
<b>Centro</b>	Universitat Politècnica de València

# 1 Resumen de las ideas clave

En este artículo se introduce al lector en la creación y uso de bibliotecas en el entorno de desarrollo Keil uVision. Todo ello de forma guiada, paso a paso, para que pueda seguirlo y aplicarlo a la vez. La reutilización de software que propician las bibliotecas es un aspecto de gran importancia para un desarrollador, debido a que permite crear proyectos reutilizando “piezas” y por tanto nos permite ser más eficientes en nuestros desarrollos.

## 2 Introducción

Una biblioteca no es más que un módulo que contiene una serie de funciones, en este caso en lenguaje C [1]. Normalmente dichas funciones guardarán algún tipo de relación, por ejemplo serán funciones relacionadas con cierto tipo de cálculos, relacionadas con el manejo de un dispositivo, etc. El principal objetivo que se persigue con el uso de bibliotecas es organizar el software que se va desarrollando de forma que posteriormente permita su reutilización de forma sencilla. Además, no hemos de perder de vista que no solo podemos utilizar en futuros proyectos nuestras propias bibliotecas que hayamos ido desarrollando a lo largo del tiempo, sino también bibliotecas de terceros (en C existe gran cantidad de bibliotecas disponibles, siendo una de sus principales ventajas), pudiendo nosotros también colaborar y compartir nuestros desarrollos. Todo ello nos permitirá ser más eficientes en nuestro trabajo.

En una biblioteca en C, el fichero que contendrá la implementación de las funciones que la componen tendrá como extensión .c y el fichero que contendrá la declaración de las mismas tendrá como extensión .h (llamado fichero de cabecera) [1].

En los siguientes apartados veremos cómo podemos crear bibliotecas e ir almacenándolas de forma ordenada para facilitar su reutilización posterior, y cómo realizar ésta. Todo ello utilizando el IDE Keil uVision [2] que es un entorno integrado de desarrollo de sistemas empotrados [3].

## 3 Objetivos

Una vez que el alumno se lea con detenimiento este documento, será capaz de:

- Entender la filosofía de bibliotecas.
- Desarrollar bibliotecas utilizando el IDE Keil uVision.
- Incorporar una biblioteca en un proyecto propio.

## 4 Desarrollo

A continuación se desarrollarán cada uno de los aspectos indicados en la introducción y objetivos, realizando las explicaciones de la forma más práctica y guiada posible.

### 4.1 Creación de una estructura de directorios para los proyectos y bibliotecas

Aunque el lector puede adaptar esta estructura a sus necesidades, creemos que es adecuada para la mayoría de los casos, y un buen punto de partida para organizar y archivar el código que vayamos desarrollando, así como código de terceros que podamos obtener. La propuesta consiste, como puede apreciarse en la figura 1, en crear un directorio que hemos llamado "proyectos" donde iremos creando un subdirectorio para cada proyecto que generemos. Dentro de este directorio terceros hemos creado un subdirectorio llamado "terceros", en el cual iremos creando un subdirectorio para cada biblioteca que creemos u obtengamos de un tercero (de ahí el nombre).

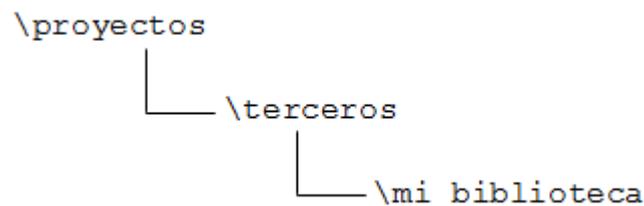


Imagen 1. Estructura de directorios

### 4.2 Creación de una biblioteca

Como se ha indicado en la introducción, una biblioteca para proyectos en lenguaje C básicamente se compone de un fichero de cabecera (con extensión .h) y un fichero de código (con extensión .c) [1]. Por tanto para crear nuestra biblioteca seguiremos los siguientes pasos:

- Crear un fichero en blanco (como se muestra en la figura 2), y guardarlo con el nombre que queramos y extensión .h dentro de la carpeta que acabamos de crear en el paso anterior, como se indica en la figura 3. En este fichero escribiremos las declaraciones de las funciones que vamos a incluir en esta biblioteca (solo los prototipos o cabeceras, no el código que las implementa), así como la definición de tipos (typedef), constantes (#define), etc. También se realizarán los #include necesarios.

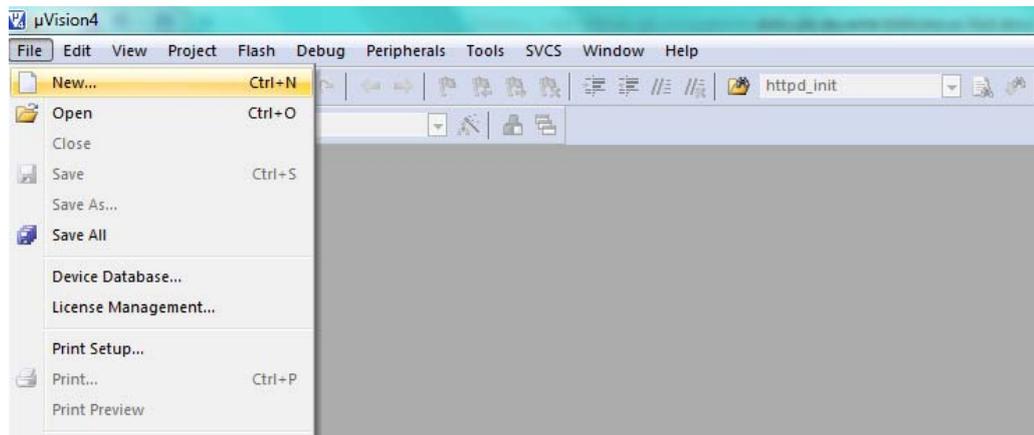


Imagen 2. Creación ficheros componentes de la biblioteca

- Crear un fichero en blanco (menú "File > New.."), y guardarlo con el nombre que queramos y extensión .c dentro de la carpeta que hemos creado para nuestra biblioteca previamente. En este fichero implementaremos (programaremos) las funciones declaradas en el fichero de cabecera .h, el cual incluiremos mediante un #include..

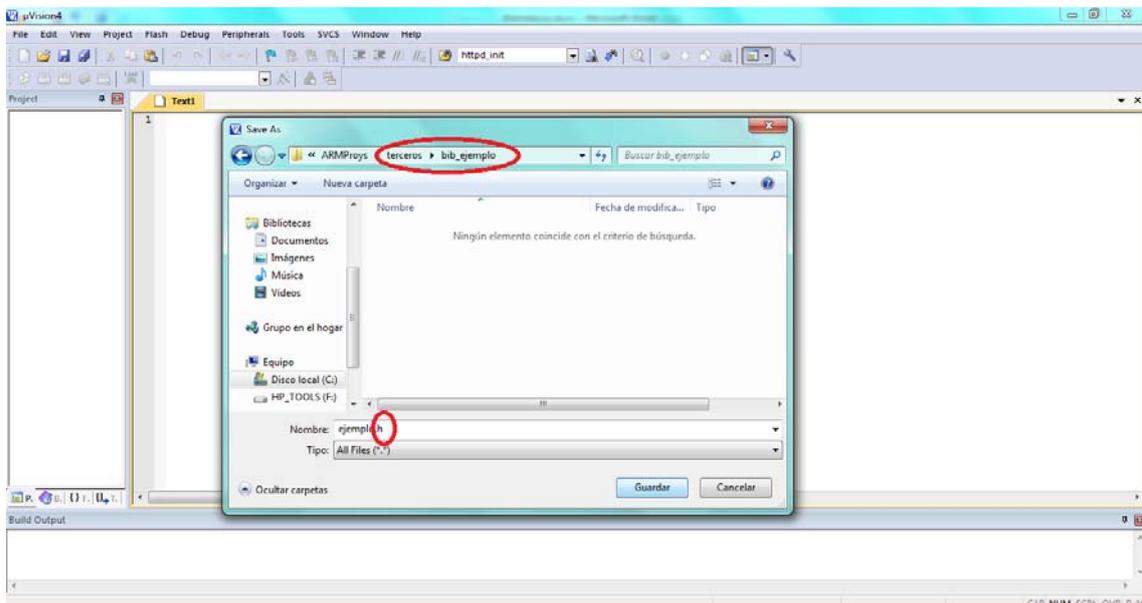


Imagen 3. Guardando los ficheros, con indicación extensión correspondiente en cada fichero (.h / .c)

### 4.3 Uso de bibliotecas

Para usar una biblioteca seguiremos los siguientes pasos:

- Nos cercioraremos que la hayamos ubicado en nuestra estructura de directorios (carpeta terceros).
- Añadiremos el módulo .c a nuestro proyecto. Para ello, una vez abierto con el Keil el proyecto desde el que vamos a utilizar la biblioteca, localizaremos en la ventana vertical de la izquierda titulada "Project" la carpeta lógica donde añadiremos el módulo en nuestro proyecto, y sobre dicha carpeta presionaremos el botón derecho del ratón, y elegiremos la opción "Add Files to Group..." (figura 4), tras lo cual nos aparecerá un cuadro de diálogo que nos permitirá navegar hasta el directorio terceros, donde podremos localizar el subdirectorio correspondiente a la biblioteca que queremos incorporar. Una vez seleccionado el módulo .c clicaremos en el botón "Add" y posteriormente en el botón "Close".

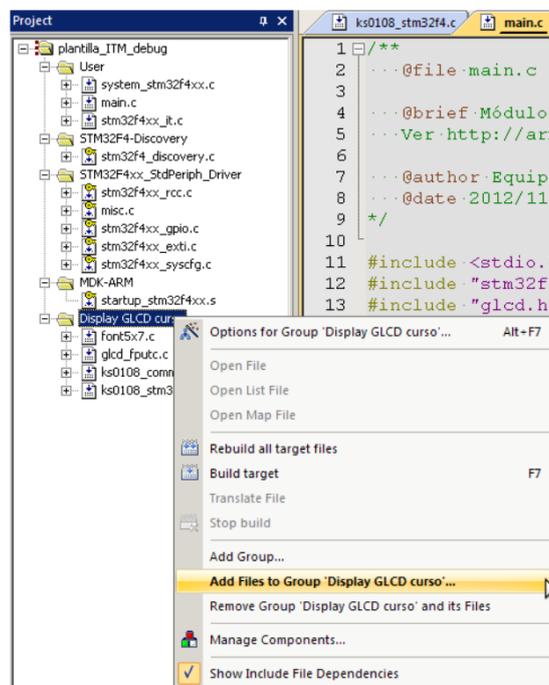


Imagen 4. Añadir módulos a un proyecto Keil

- Ahora tendremos que añadir la ruta de la carpeta de nuestra biblioteca en la lista de rutas (Include Paths) del entorno Keil. Para ello seleccionaremos la opción "Options for Target" del menú "Project", pestaña "C/C++", y pulsaremos en el botón de los tres puntos, a la derecha de la caja "Include Paths" (figura 5).

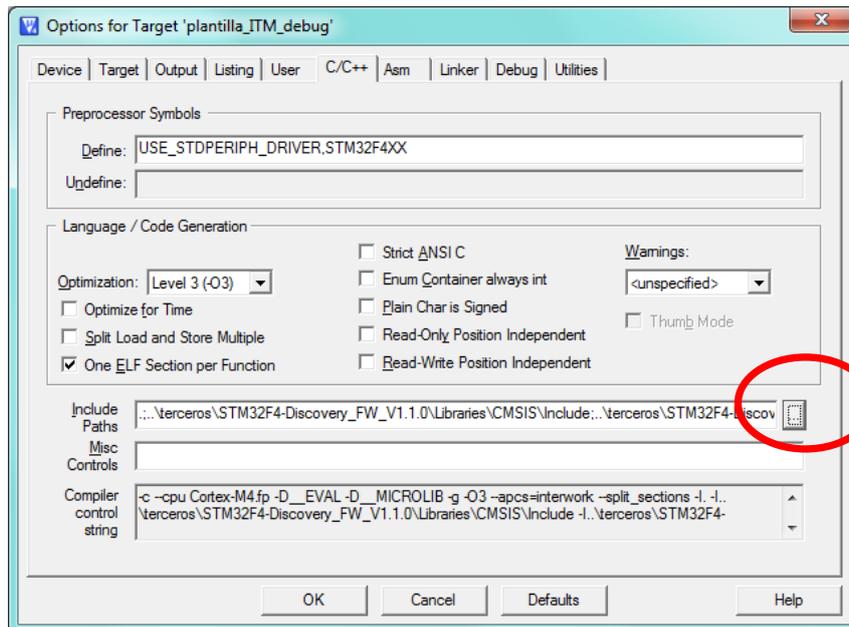


Imagen 5a. Añadir rutas al listado "Include Paths" del keil

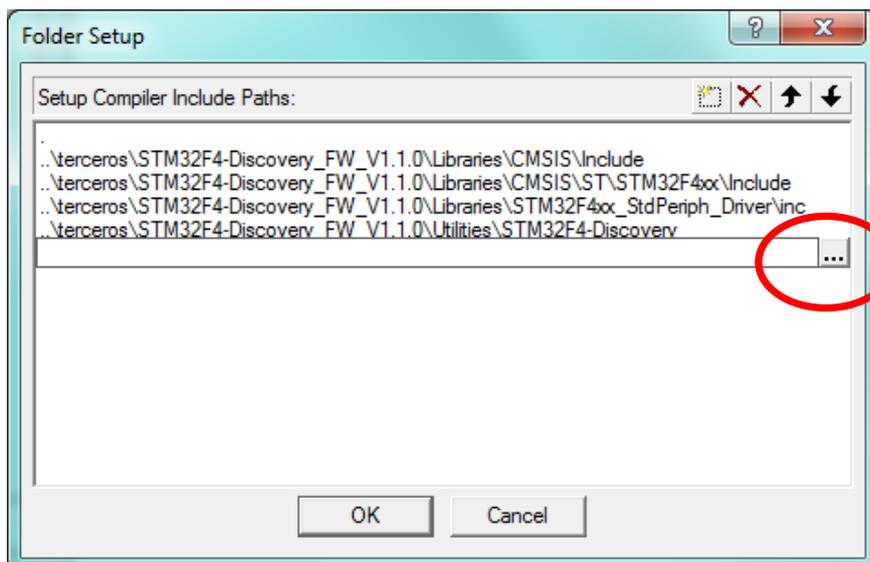


Imagen 5b. Añadir rutas al listado "Include Paths" del keil

- De esta forma ya podremos incorporar un `#include` a la biblioteca creada y que vamos a utilizar desde un proyecto nuevo (sin necesidad de incluir toda la ruta o path) como se muestra en la figura 6, e invocar desde el mismo las funciones que necesitemos de la biblioteca.

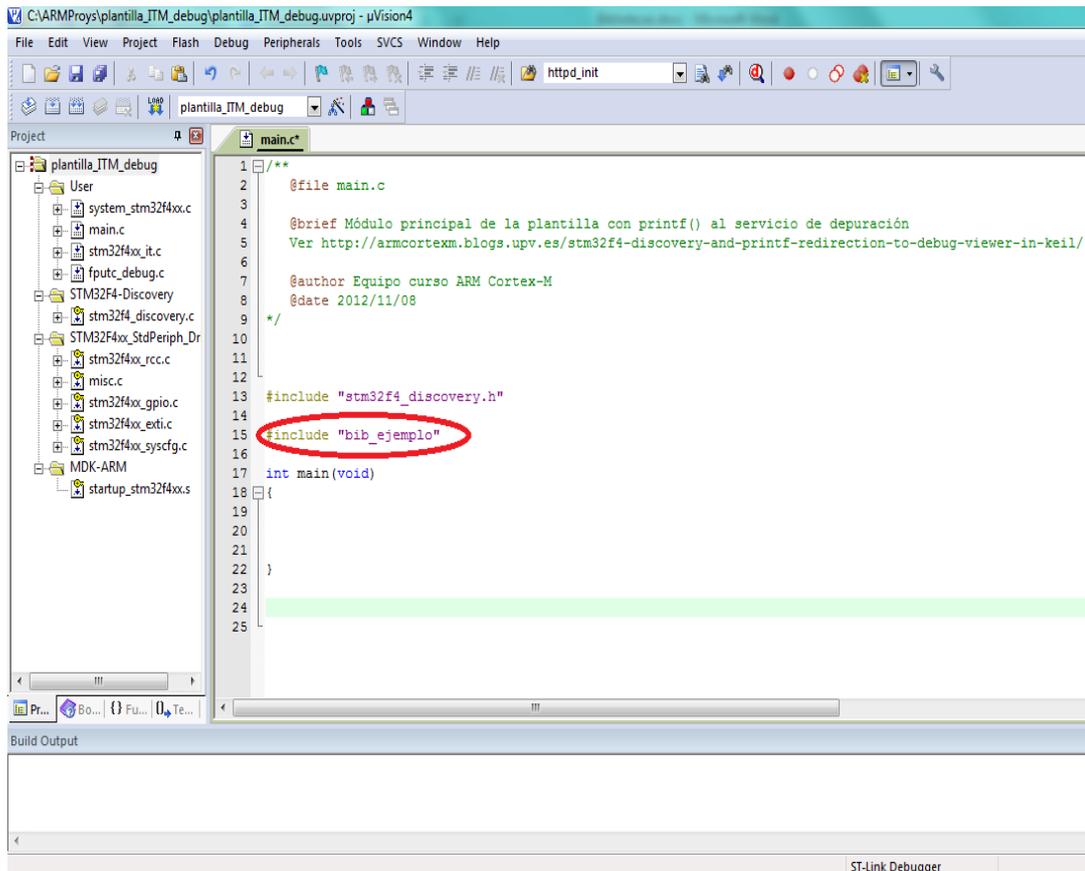


Imagen 6. Directiva #include para utilización de la biblioteca "bib\_ejemplo.h" desde otro proyecto

## 5 Cierre

A lo largo de este objeto de aprendizaje hemos tratado los fundamentos de la creación y utilización de bibliotecas en C con el entorno de desarrollo Keil uVision, lo cual te permitirá organizar mejor el código que vayas desarrollando facilitándote su posterior reutilización, pudiendo también utilizar bibliotecas de terceros.

Para comprobar qué realmente has aprendido cómo crear y utilizar una biblioteca es el momento de que te pongas manos a la obra e intentes crear una, por ejemplo que contenga funciones de cálculo de áreas de figuras geométricas, y posteriormente la utilices desde un proyecto Keil, invocando las funciones que has implementado en la biblioteca.

¡¡ÁNIMO!!

## 6 Bibliografía

[1] Galeano Gustavo: Programacion de sistemas embebidos en C, Alfaomega grupo editor, 2010.

[2] <http://www.keil.com/arm/mdk.asp>

[3] Jonathan W. Valvano: Introduction to the Arm Cortex-M microcontrollers. Fifth edition, 2015.