
Manual de Instalación y configuración de la librería ARUCO en Windows

<http://umh1782.edu.umh.es>

Requisitos: - Windows XP, Windows 7, Windows 8 - (32 bits)
- Microsoft Visual Studio C++ 2010
- Librería OpenCV instalada y configurada

Más información en <http://www.uco.es/investiga/grupos/ava/node/26>

1) Instalar la librería ARUCO:

ARUCO es una librería *OpenSource* con licencia BSD desarrollada por el grupo **AVA** de la Universidad de Córdoba para la implementación de aplicaciones de *Realidad Aumentada* mediante la detección de marcadores codificados. Está basada en la librería **OpenCV**, por lo que debemos tener esta librería instalada en nuestro ordenador ([Manual de instalación de OpenCV](#))

Podemos descargar el código fuente de la librería desde la web del proyecto ARUCO (<http://sourceforge.net/projects/aruco/>). Para simplificar la instalación, en el blog de la asignatura (pestaña **material/software**) está disponible una versión ya compilada para Windows (x86-VisualC++2010-OpenCV2.4.8). También dispones de una versión compilada para el compilador *Visual Studio 2013*. Si tienes otra versión de Windows /Compilador/LibreríaOpenCV deberás compilarla de forma manual.

- Descargaremos las librerías ARUCO compilada ([arucolib125_vc10_opencv248.zip](#)) desde el blog de la asignatura (Apartado **Material/Software**).
- Descomprimiremos el contenido del fichero. Está formado por tres carpetas:
 - o **include:** contiene las cabeceras de la librería
 - o **lib:** contiene la librería en formato estático (versión normal y debug)
 - o **utils:** contiene aplicaciones en línea de comandos para testear la librería y gestionar marcadores.
- Copiaremos el contenido en la carpeta '**c:\opencv**'

2) Configurar el proyecto en VC++2010 con la librería ARUCO (1ª vez):

Seguiremos el mismo procedimiento visto en el tutorial para crear un proyecto en Visual C++2010 con la librería OpenCV utilizando **Hojas de Propiedades**. Podemos modificar las hojas de propiedades ya creadas, pero en este manual describiremos el proceso partiendo de cero.

Una vez creado el proyecto, la ventana del programa nos muestra en el panel izquierdo el gestor de configuración del proyecto '**Property Manager**'. Si no estuviera visible esta ventana, la activaríamos con la opción del menú '**View > Property Manager**', o '**View > Other Windows > Property Manager**'.

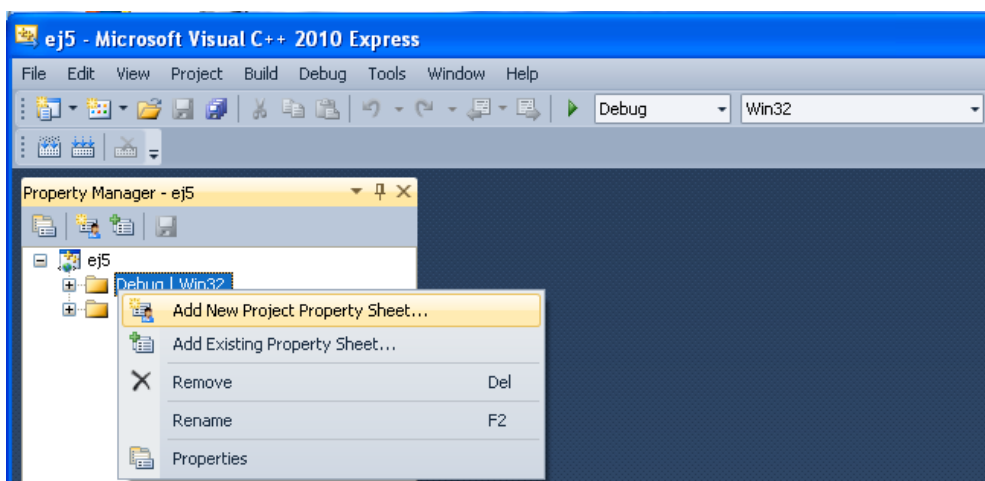
En la ventana del gestor de configuración ('**Property Manager**') nos mostrará la página creada con dos apartados (**Debug/Release**). En Visual C++ tenemos dos modos de compilación, cada uno de ellos con una configuración independiente:

- **Modo Debug:** el compilador añade código extra que nos ayudara a depurar los errores de nuestro programa. Este es el modo que utilizaremos habitualmente
- **Modo Release:** cuando el programa está terminado, la compilación se optimiza eliminando el código de depuración, obteniendo así un ejecutable más pequeño y más rápido.

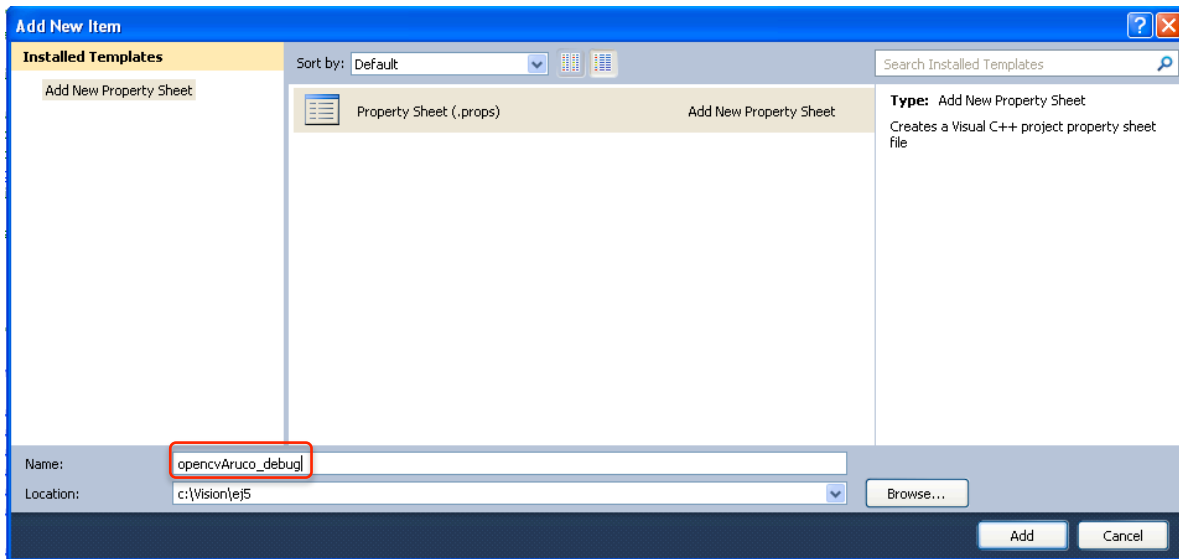
Vamos a describir la configuración de los dos modos:

a) Configuración modo Debug:

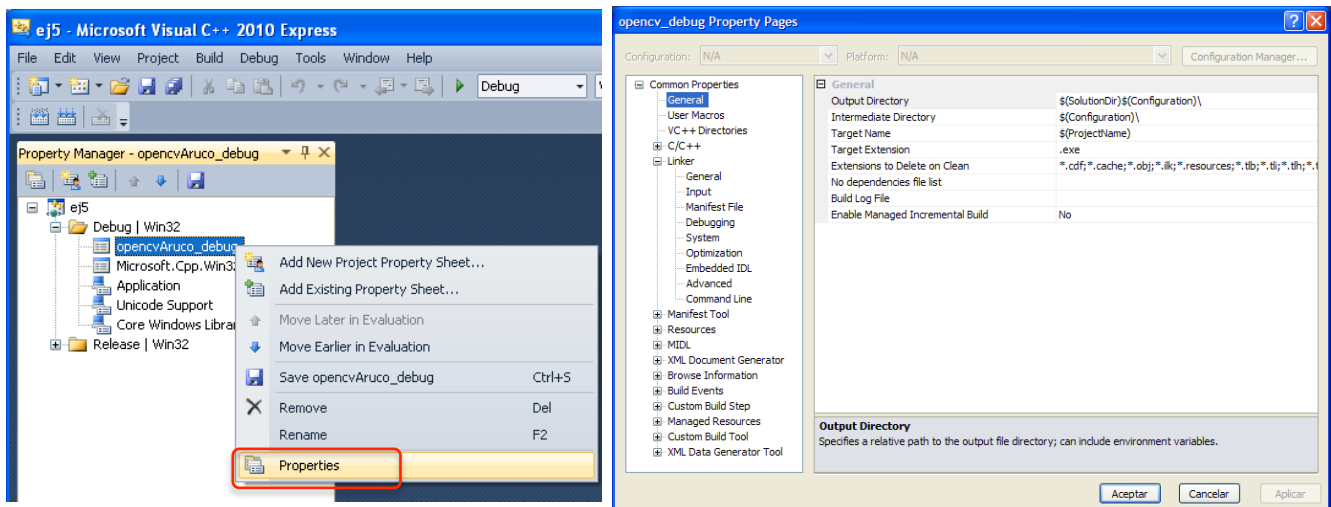
- Pulsaremos el botón derecho del ratón sobre nodo '**Debug**', desplegando el menú contextual en el que seleccionaremos '**Add New Project Property Sheet**'.



- Nos abrirá una nueva ventana donde asignaremos el nombre del fichero de configuración (**opencvAruco_debug**). Pulsando el botón '**Add**' creará el fichero de configuración en la carpeta de nuestro proyecto (**opencvAruco_debug.props**).



- Pulsaremos el botón derecho del ratón sobre la etiqueta '**opencv_debug**' de la hoja de propiedades del modo '**Debug**' que hemos creado. En el menú contextual seleccionaremos la opción '**Properties**'. Esto abrirá la ventana donde realizaremos la configuración del proyecto.



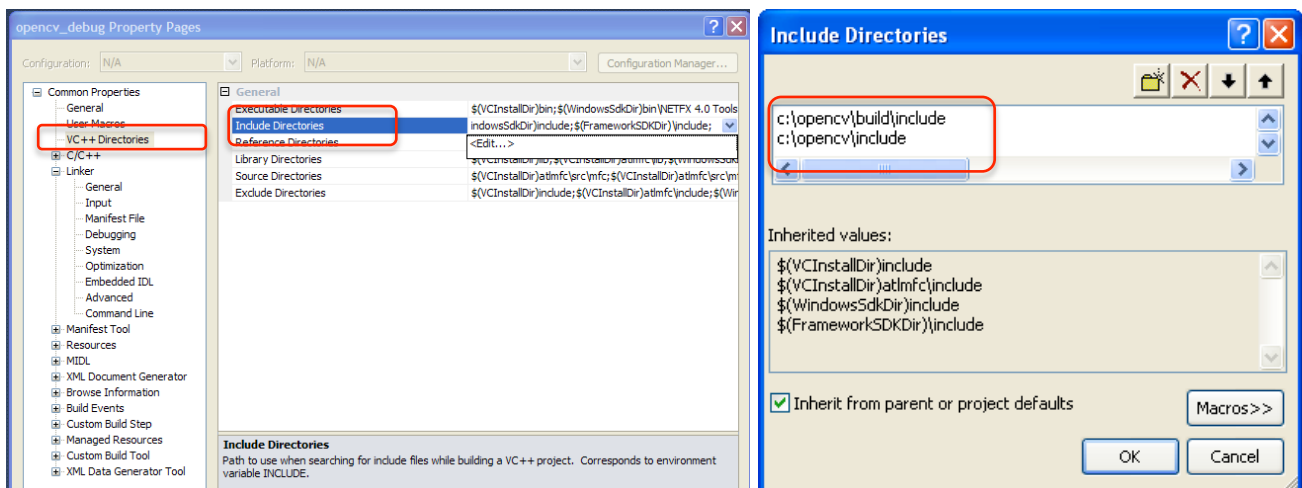
A continuación modificaremos tres apartados del fichero de configuración:

- *VC++ Directories -> Include Directories*
- *VC++ Directories -> Library Directories*
- *Linker -> Input -> Additional Dependencies*

Vamos por partes:

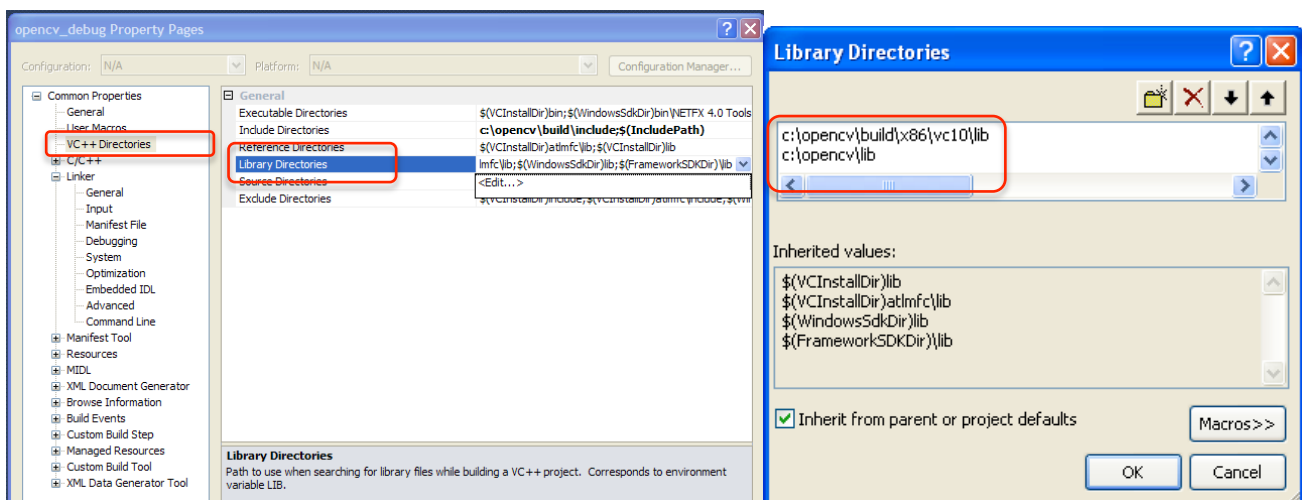
a.1) Directorio de cabeceras:

- Editaremos la opción **VC++ Directories -> Include Directories**, pulsando sobre la flecha azul del lateral derecho nos muestra un menú con la opción **<Edit>** que pulsaremos y abrirá una ventana de edición. Añadiremos los directorios:
 - o **c:\opencv\build\include**
 - o **c:\opencv\include**



a.3) Directorio de librerías:

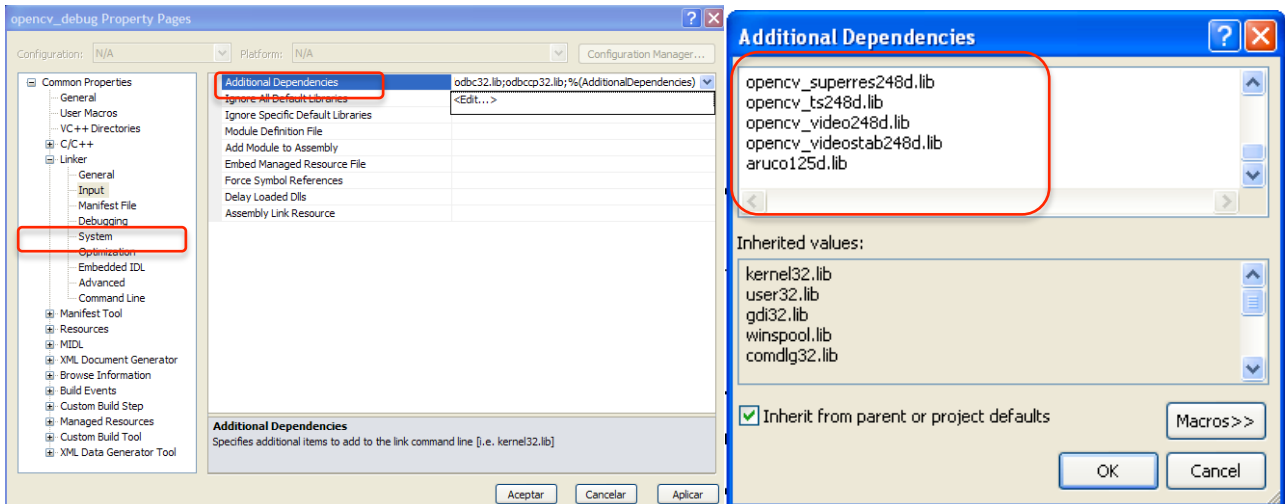
- Editaremos la opción **VC++ Directories -> Library Directories**, pulsando sobre la flecha azul del lateral derecho nos muestra un menú con la opción **<Edit>** que pulsaremos y abrirá una ventana de edición. Añadiremos los directorios específicos de nuestro S.O.:
 - **c:\opencv\build\x86\vc10\lib**
 - **c:\opencv\lib**



a.3) Librerías:

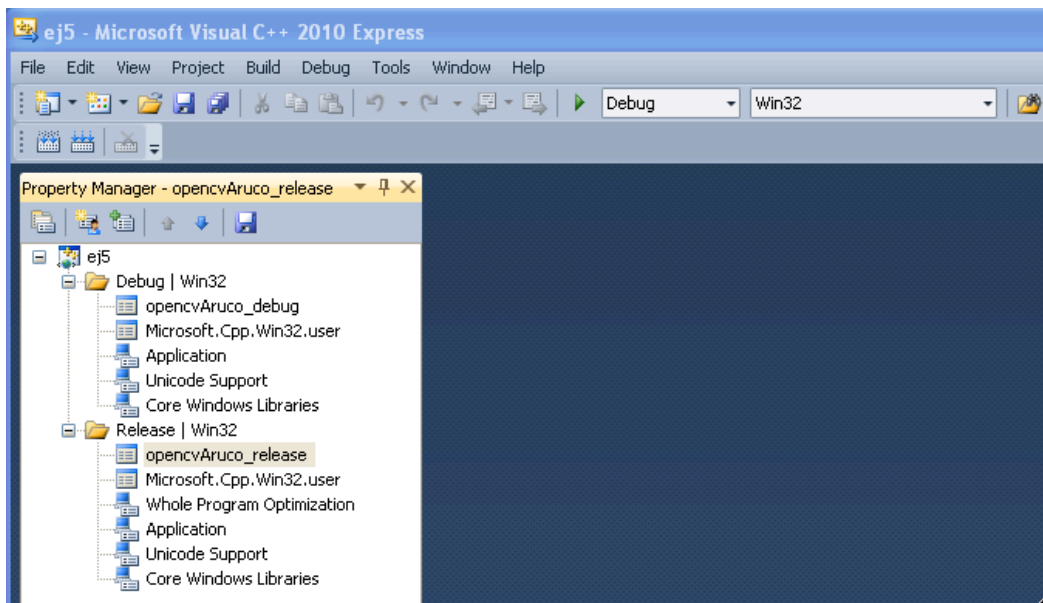
- Editaremos la opción **Linker -> Input -> Additional Dependencies**, pulsando sobre la flecha azul del lateral derecho nos muestra un menú con la opción **<Edit>** que pulsaremos y abrirá una ventana de edición. Añadiremos el listado de librerías de la **opencv** y la librería **ARUCO**:

```
opencv_calib3d248d.lib
opencv_contrib248d.lib
opencv_core248d.lib
opencv_features2d248d.lib
opencv_flann248d.lib
opencv_highgui248d.lib
opencv_imgproc248d.lib
opencv_legacy248d.lib
opencv_ml248d.lib
opencv_nonfree248d.lib
opencv_objdetect248d.lib
opencv_photo248d.lib
opencv_stitching248d.lib
opencv_superres248d.lib
opencv_ts248d.lib
opencv_video248d.lib
opencv_videostab248d.lib
aruco125d.lib
```



b) **Configuración modo Release:**

- Repetiremos los mismos pasos del apartado anterior pero sobre la etiqueta '**Release**'. El fichero de propiedades lo llamaremos '**opencvAruco_release**' y en lugar de las versiones de depuración de las librerías de la **opencv** y **ARUCO**, deberemos incluir las librerías sin información de depuración:



```
opencv_calib3d248.lib
opencv_contrib248.lib
opencv_core248.lib
opencv_features2d248.lib
opencv_flann248.lib
opencv_highgui248.lib
opencv_imgproc248.lib
opencv_legacy248.lib
opencv_ml248.lib
opencv_nonfree248.lib
opencv_objdetect248.lib
opencv_photo248.lib
opencv_stitching248.lib
opencv_superres248.lib
opencv_ts248.lib
opencv_video248.lib
opencv_videostab248.lib
aruco125.lib
```

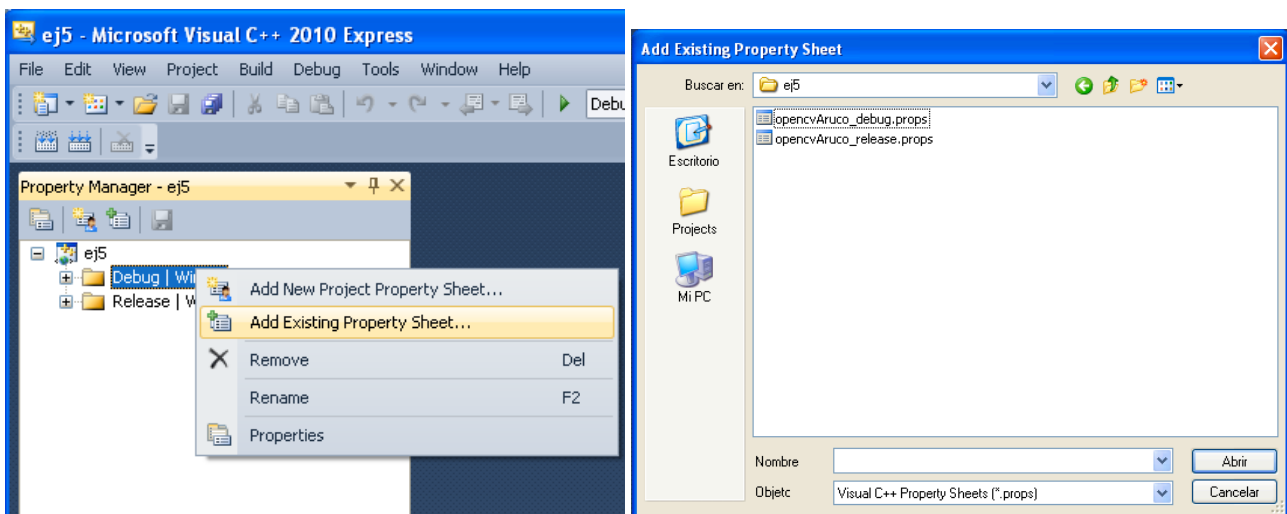
Al terminar este apartado tendremos en la carpeta de nuestro proyecto dos ficheros con toda la configuración que guardaremos para uso futuro:

opencvAruco_debug.props
opencvAruco_release.props

3) Configurar el proyecto a partir de una página de propiedades previa:

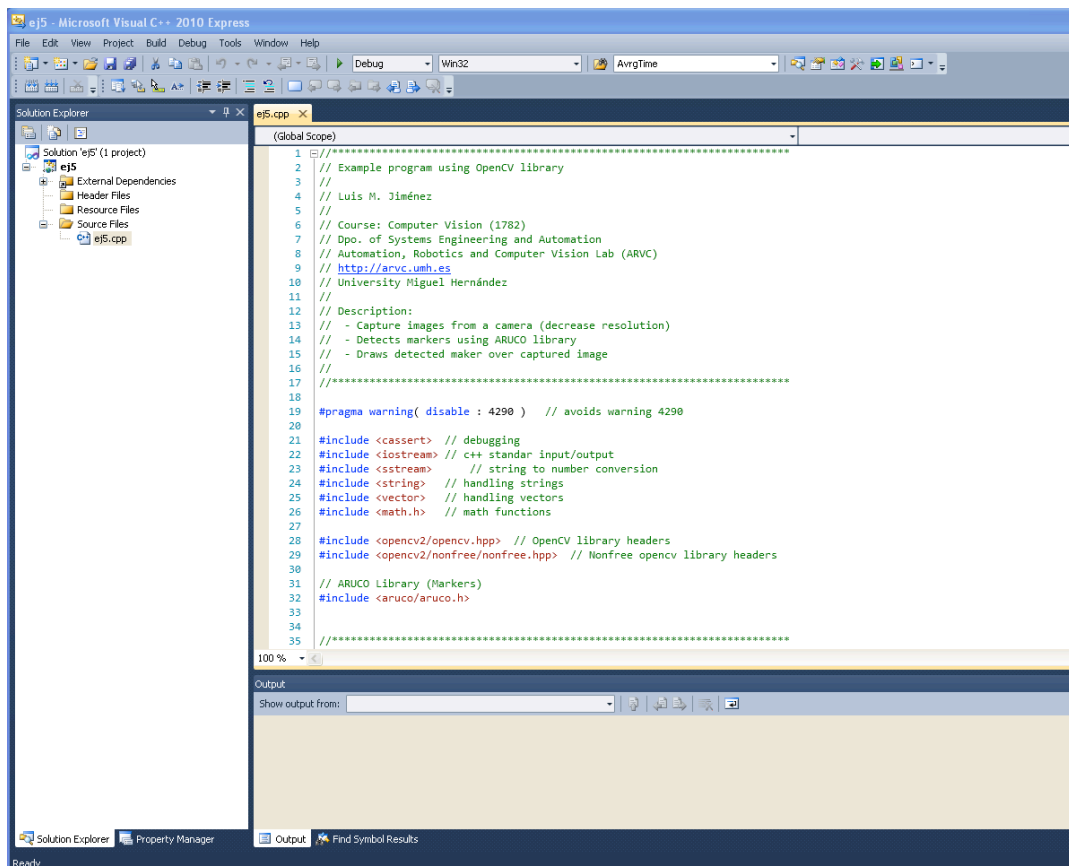
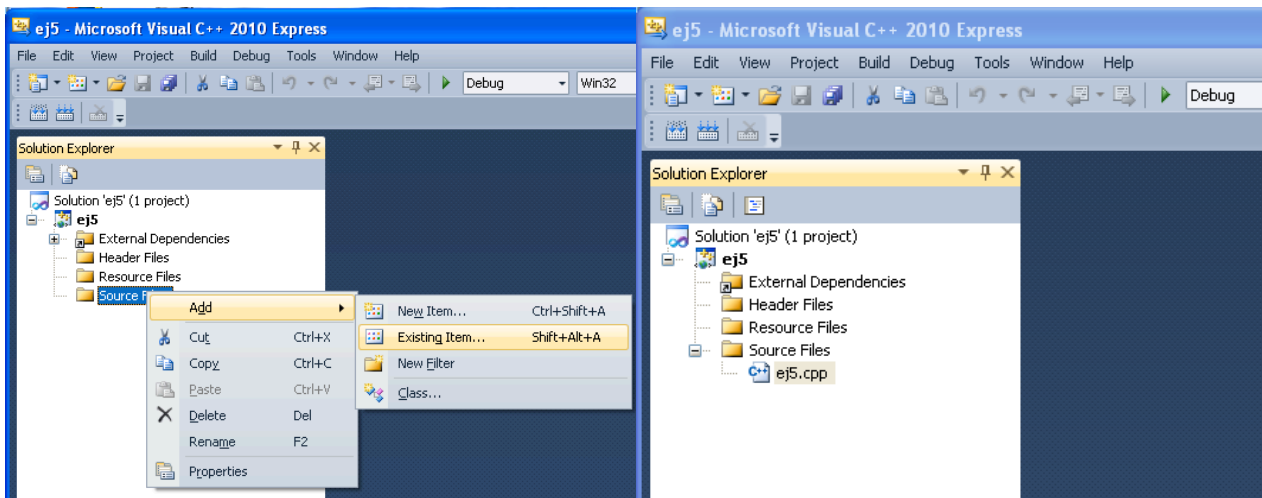
En este caso reutilizaremos los ficheros de configuración creados en el proyecto anterior.

- Copiaremos los ficheros '*opencvAruco_debug.props*' y '*opencvAruco_release.props*', en la carpeta de nuestro proyecto.
- En la ventana '*Property Manager*' de nuestro proyecto, pulsando con el botón derecho del ratón sobre las pestañas '*Debug*' y '*Release*' añadiremos las hojas de propiedades correspondientes con la opción del menú '*Add Existing Property Sheet*'



4) Añadir el código fuente de nuestro programa:

- Descargaremos el fichero de ejemplo '*ej5.cpp*' del blog de la asignatura. Lo copiaremos en la carpeta creada para nuestro proyecto.
- Activaremos la vista del '*Solution Explorer*'. Menú '*View-> SolutionExplorer*'
- Sobre el nodo de la aplicación (*ej5*) desplegaremos las subcarpetas y pulsaremos el botón derecho del ratón sobre la carpeta '*Source Files*'. Seleccionaremos la opción '*Add/Existing Files*' y cargaremos el fichero de código *ej5.cpp*.



A continuación compilaremos el programa (Menú '**Build/Build Solution**' o **F7**) y lo ejecutaremos (Menú **Debug/Start Debugging**' o **F5**).

