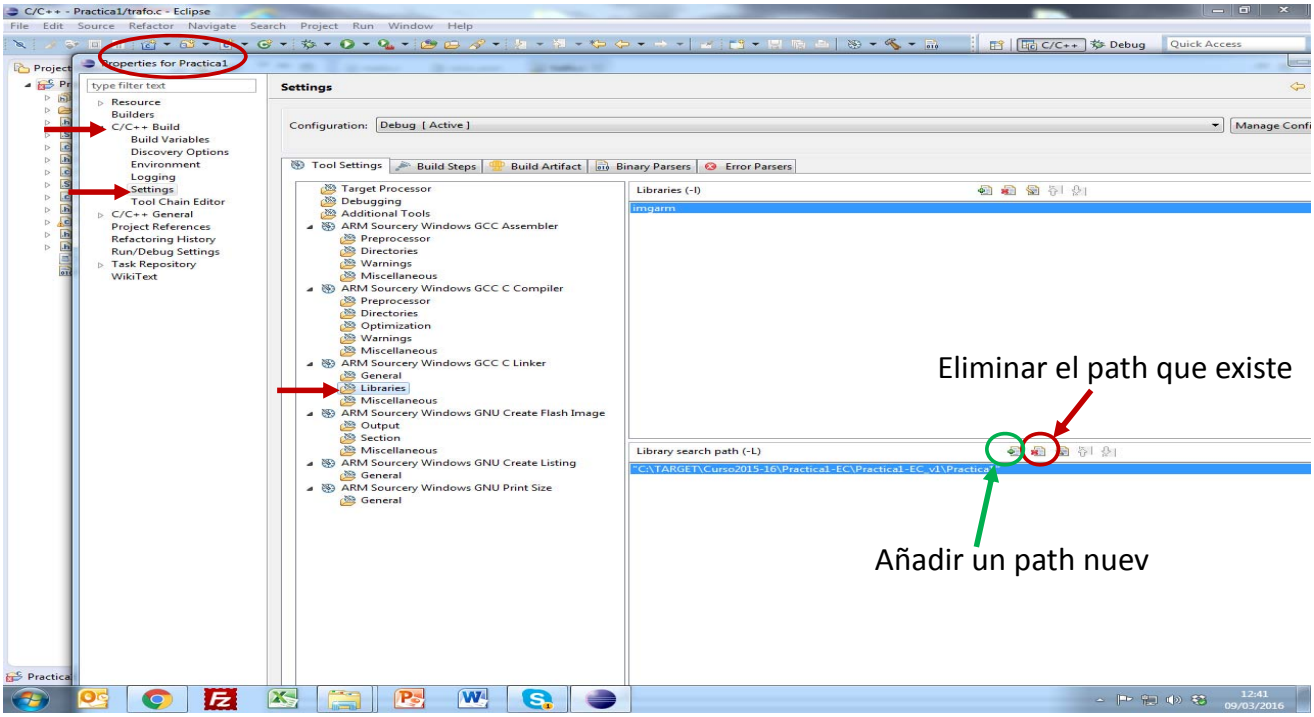
Cosas importantes sobre la práctica 1

Tenéis que indicar el path de la biblioteca libimgarm.a

- Ratón encima del proyecto: Botón derecho seleccionar propiedades



Sobre las funciones que hay que hacer en ensamblador

- En la función apply-gaussian() hay que tener en cuenta que se llama a otra función que tiene 5 parámetros:

  int gaussian(unsigned char* im1, int width, int height, int i, int j)

  El quinto parámetro hay que pasarlo por la pila

- En la función rgb2gray() hay que multiplicar por coeficientes

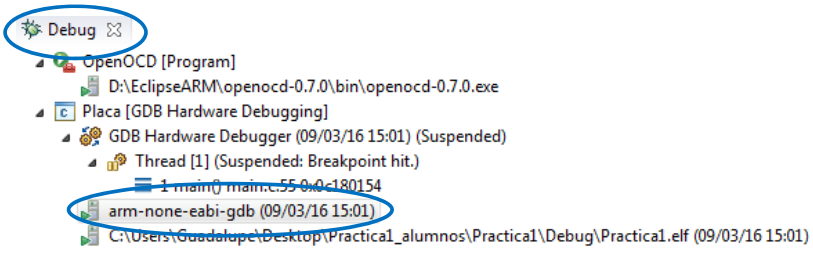
  dest = (3483*orig.R + 11718*orig.G + 1183*orig.B) /16384

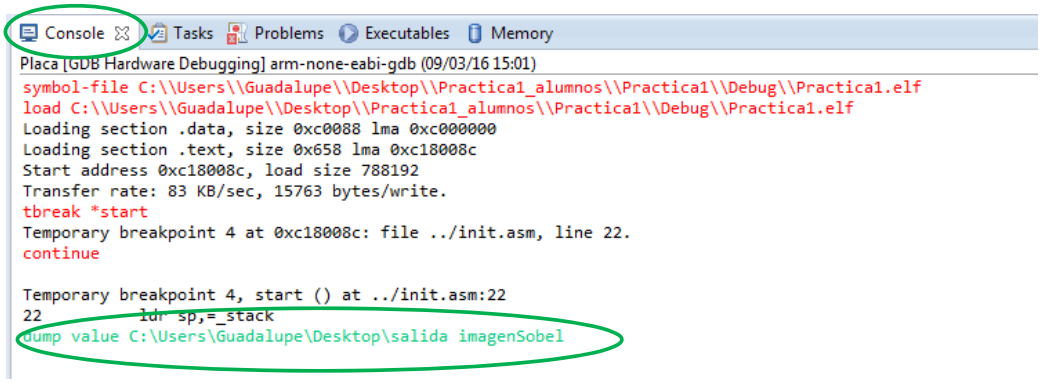
  Estos coeficientes tienen valores muy grandes que no permite la inst mov, luego para guardar en un registro ese valor hay que usar ldr:

  ~~mov, R0, # 3483~~ ldr R0, = 3483

Para Visualizar las imágenes que se van generando

1. **Pasar los valores de la memoria de la placa a un fichero en el PC**
   - Poner un break point en la instrucción return del main
   - En la ventana Debug seleccionar: **arm-none-eabi-gdb**
   - En la ventana de la consola escribir: **dump value nombreFicheroSalida NombreMatrizImagen**



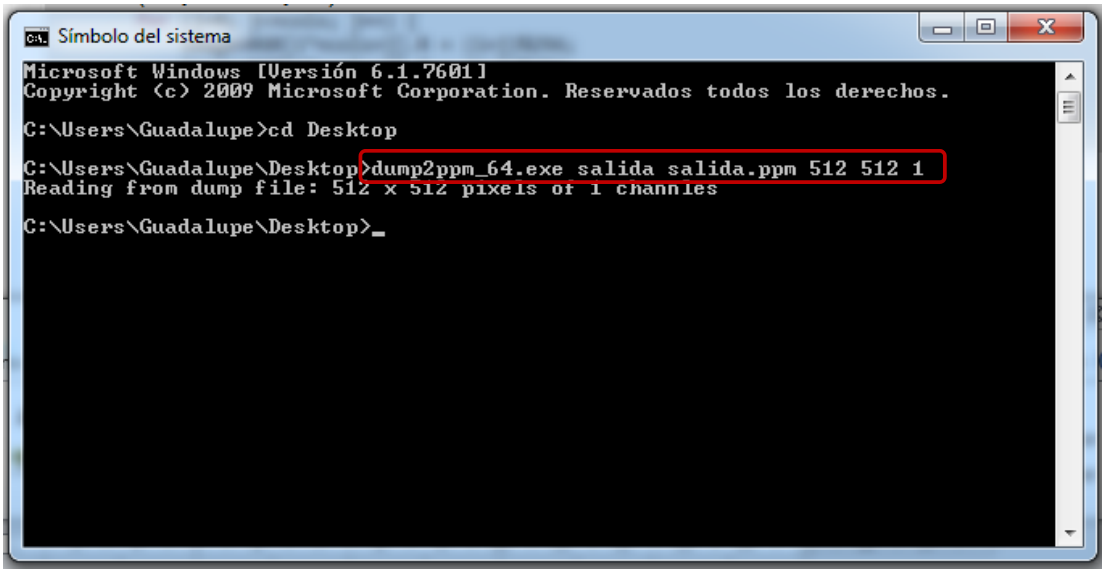


Para Visualizar las imágenes que se van generando

2. Convertir el fichero de salida de la práctica a extensión ppm

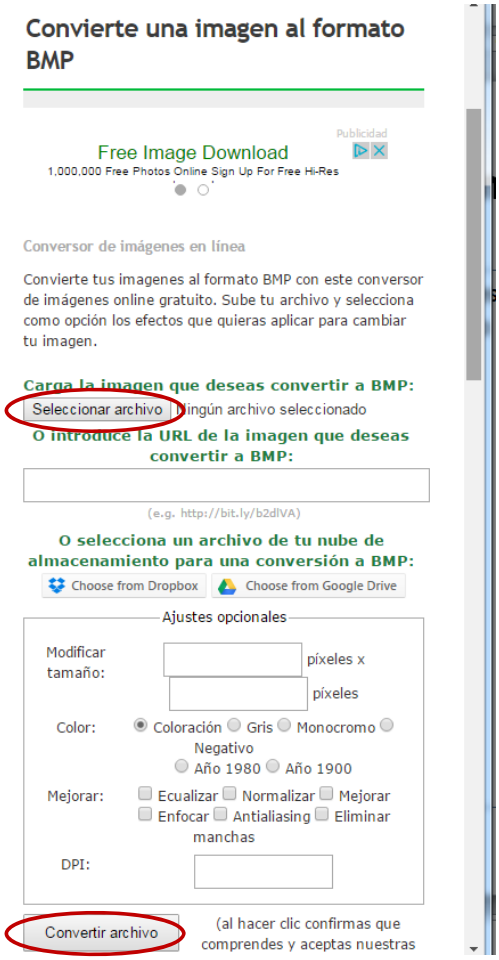
- Abrir la consola del sistema
- Colocarse en la carpeta donde está el fichero de salida de la práctica y el dump que se proporciona con la práctica (de 32 o 64 bits dependiendo del ordenador)
- Desde esa carpeta escribir en la consola

dump2ppm._64.exe nombreFicheroSalida nombrefichero.ppm 512 512 1



Para abrir un fichero con extensión ppm

- Si en el ordenador no se dispone de ninguna herramienta para visualizar en ese formato, **existen herramientas online para convertir una imagen PPM en BMP**, formato propietario de Windows.
- **Por ejemplo**



Para Visualizar las imágenes que se van generando

- Esta es la imagen final que genera la práctica

Para saber las direcciones de memoria de las variables y de las funciones

- En esta práctica al tener variables que ocupan mucho (Lena es una imagen de 512x512) no es fácil saberlo mirando la ventana de desensamblado
- **Para saber la dirección donde empiezan las funciones**
  - Poniendo el ratón encima de la función te dice en que dirección de memoria está

Para saber las direcciones de memoria de las variables y de las funciones

- **Para saber la dirección de las variables:**
  - Poner un break point en las funciones
  - Cuando se para en una función dar al icono de entrar en la función  una vez
  - Una vez dentro de la función hay dos maneras de saber cuales son las direcciones de los vectores que contienen las imágenes
    - En los registro se puede ver los valores de los parámetros de la función
      - En las tres funciones que tenemos los primeros parámetros son las direcciones de los vectores que contienen las imágenes
    - Poniendo el ratón encima de las variables de la función se puede ver su dirección y su contenido