ANEXO INSTALACIÓN

INSTALACIÓN DEL ENTORNO (LINUX EMPOTRADO Y LIBRERÍAS)

Un entorno de desarrollo en cualquier plataforma de trabajo constituye un elemento principal para la implementación de cualquier proyecto. En este trabajo fin de máster se requiere un entorno de trabajo en el sistema operativo Linux que sea compatible con la arquitectura de la OdroidXU4. Dado que el SO Linux es un entorno abierto, existen numerosas versiones de este, pero no todas son soportadas por esta placa de desarrollo.

Se ha seleccionado la versión del Sistema Operativo "Ubuntu Mate Odroid XU4" de la página oficial de Ubuntu. La imagen se encuentra en el siguiente enlace:

https://odroid.in/ubuntu_16.04lts/

Dentro del conjunto de imágenes disponibles, se ha seleccionado la *"ubuntu-16.04-mate-odroid-xu3-20161011.img.xz"*. La imagen del sistema se ha instalado en una tarjeta micro SD con capacidad de 32 GB, ya que la capacidad del SO es de 1GB. A pesar de que la OdroidXU4 posee una memoria interna MMC, es necesario disponer de suficiente memoria para la instalación de los distintos programas que se van a utilizar. En la siguiente imagen se muestra la tarjeta de memoria utilizada:

Esta memoria posee una velocidad de lectura de hasta 95 MB/s, con una tecnología de memoria flash SDHC. Estas características la hacen adecuada para el propósito deseado. Una vez montado el sistema operativo, es necesario instalar una serie de librerías y dependencias.

Tras arrancar el SO, la imagen de escritorio es:

El siguiente paso consiste en actualizar el sistema, dependencias y librerías que serán necesarias para compilar los distintos códigos del proyecto. Los pasos se describen a continuación:

$ sudo apt-get upgrade

$ sudo apt-get update

$ sudo apt-get install kate

$ sudo apt-get install build-essential

$ sudo apt-get install linux-headers-generic

$ sudo apt-get install linux-headers-$(uname –r)

$ sudo apt-get install dkms

$ sudo reboot

$ sudo apt-get install linux-headers-$(uname -r) dkms

$ sudo apt-get install module-assistant

$ sudo m-a prepare

A continuación, instalamos las herramientas de compilación, así como las herramientas para la interfaz y poder crear ventanas:

$ sudo apt-get install build-essential cmake

$ sudo apt-get install qt5-default libvtk6-dev

Las librerías necesarias para la entrada y salida de imagen de vídeo son las siguientes:

$ sudo apt-get install zlib1g-dev libjpeg-dev libwebp-dev libpng12-dev libtiff5-dev libjasper-dev libopenexr-dev libgdal-dev libdc1394-22-dev libavcodec-dev libavformat-dev libswscale-dev libtheora-dev libvorbis-dev libxvidcore-dev libx264-dev yasm libopencore-amrnb-dev libopencore-amrwb-dev libv4l-dev

Después se instalan las librerías para la programación multihebra y álgebra lineal:

$ sudo apt-get install libtbb-dev libeigen3-dev

$ sudo reboot

Por último, se instala:

$ sudo apt-get -f install

$ sudo apt-get install libboost-all-dev

$ sudo apt-get install cmake libblkid-dev e2fslibs-dev libboost-all-dev libaudit-dev

$ sudo apt-get update

$ sudo apt-get install limbedtls-dev
$ sudo apt-get install python-software-properties
$ sudo add-apt-repository ppa:grnet/synnefo
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install lubuntu-restricted-extras
$ sudo apt-get install flashplugin-installer

```
$ sudo apt-get install apt-rdepends
$ sudo apt-get install libbz2-dev
$ sudo apt-get install apt-file
$ apt-file update
$ apt-file search bzlib.h
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo reboot
```

Una vez finalizado este paso, ya estaría instalado el entorno de trabajo con todas las dependencias necesarias para editar y compilar los códigos, así como representar los resultados de manera correcta.

INSTALACIÓN DE OPENCV

En esta sección se describen la documentación completa necesaria para instalar las dependencias y el código fuente del programa. El primer paso consiste en instalar el conjunto de dependencias y la actualización de los paquetes de instalación.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo reboot
$ sudo apt-get install build-essential cmake pkg-config
$ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
$ sudo apt-get install libxvidcore-dev libx264-dev
$ sudo apt-get install libgtk2.0-dev
$ sudo apt-get install libatlas-base-dev gfortran
$ sudo apt-get install python2.7-dev python3-dev
$ sudo reboot
```

A continuación, se deben instalar las dependencias necesarias para Python.

```
$ cd ~
$ mkdir pydown
$ cd pydown
$ wget https://bootstrap.pypa.io/get-pip.py
$ sudo python get-pip.py
$ sudo pip install numpy
```

Hay que tener en cuenta que la instalación de "numpy" puede tardar bastante tiempo. Posteriormente se descarga el código fuente de OpenCV. Este documento instala la versión 3.3.1 de OpenCV. La ubicación del software lo puede encontrar en:

https://github.com/opencv/opencv/archive/3.3.1.zip

https://github.com/Itseez/opencv_contrib/archive/3.3.1.zip

No obstante, las órdenes de comando que se muestran a continuación ya tienen en cuenta la ubicación de esta versión del software.

```
$ cd ~
$ mkdir OpenCV_src
$ cd OpenCV_src
$ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.3.1.zip
$ unzip opencv.zip
$ rm opencv.zip
$ wget -O opencv_contrib.zip
https://github.com/Itseez/opencv_contrib/archive/3.3.1.zip
$ unzip opencv_contrib.zip
$ rm opencv_contrib.zip
```

Nótese que la carpeta se ha creado en el directorio raíz. Tener bien ubicadas las carpetas de instalación es importante para saber dónde se encuentran los archivos generados.

Como última orden se ha borrado el archivo comprimido de descarga, ya que el espacio de almacenamiento en la tarjeta es limitado.

La instalación del programa se realiza en la carpeta "build" mediante el comando "make". Esta instrucción puede durar 2 o 3 horas, dependiendo de la plataforma en la que se encuentre instalando.

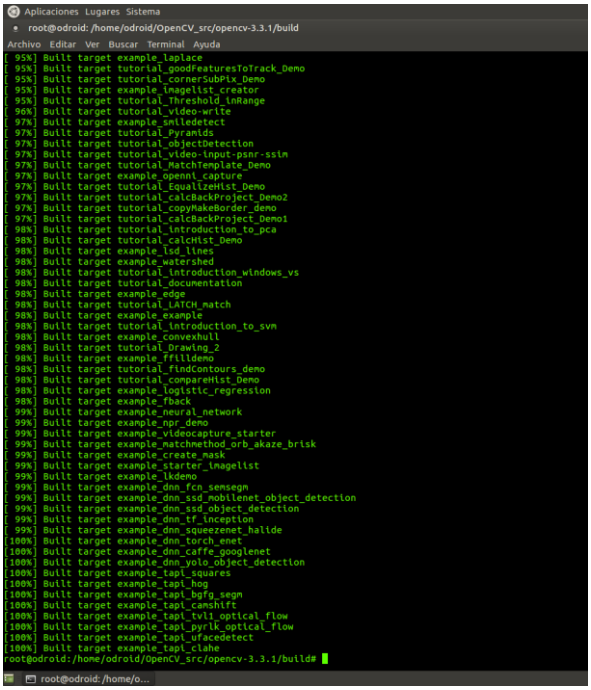
```
$ cd ~
$ cd OpenCV_src
$ cd opencv-3.3.1
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D INSTALL_PYTHON_EXAMPLES=OFF \
-D OPENCV_EXTRA_MODULES_PATH=~/OpenCV_src/opencv_contrib-
3.3.1/modules \
-D BUILD_EXAMPLES=ON ..
```

$ make

$ sudo make install

$ sudo ldconfig

Para comprobar que se ha instalado correctamente, el porcentaje de finalización de 'build-tarjet' debe ser del 100%, tal y como muestra la siguiente imagen:



Una vez instalado el programa, sólo queda establecer los enlaces entre las librerías, de tal forma que el programa se pueda compilar desde cualquier directorio. Para ello hay que seguir los siguientes pasos:

sudo nano /etc/ld.so.conf.d/opencv.conf

Una vez abierto el editor de texto, escribimos:

/usr/local/lib

A continuación, configuramos la librería con la orden:

$ sudo ldconfig -v

La creación de los enlaces se realiza con el comando:

$ export LD_LIBRARY_PATH=~/usr/local/lib/:$LD_LIBRARY_PATH

$ sudo ldconfig

$ export LD_LIBRARY_PATH=~/opencv-2.4.5/build/:$LD_LIBRARY_PATH

$ sudo ldconfig

Se edita el archivo bash.bashrc y se añade el comando de debajo .

$ sudo nano /etc/bash.bashrc

$ export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig

Por ultimo, se instala el paquete linopencv-dev:

$ sudo apt-get install libopencv-dev

Con esta última instrucción ya estaría completamente instalado el OpenCV. A la hora de compilar cualquier Proyecto, es necesario añadir el siguiente commando para incluir las librerías necesarias:

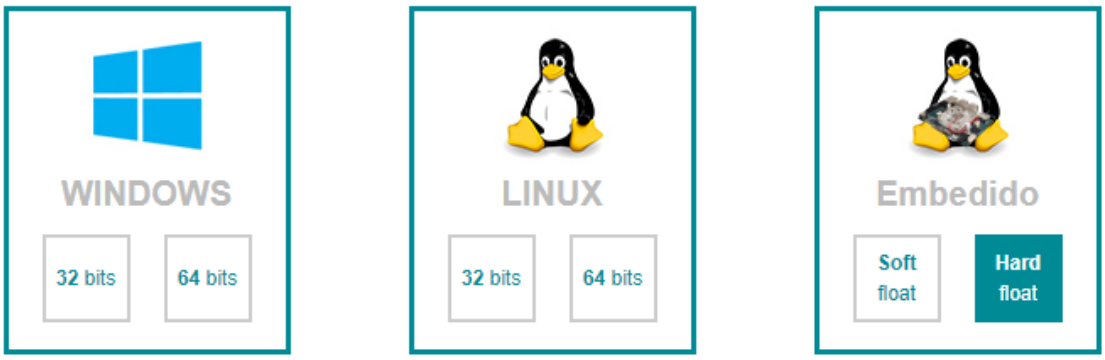
```
-ggdb `pkg-config --cflags --libs opencv`
```

INSTALACIÓN DE UEYE XS 2

Los archivos necesarios para la instalación de los drivers de la cámara UEYE XS 2 se han obtenido en la página oficial:

https://es.ids-imaging.com/store/xs.html

Este software se encuentra disponible en las plataformas de Windows, Linux y Linux embebido, siendo esta última la plataforma seleccionada para realizar el proyecto. Para ello debemos registrarnos en la página con un correo y contraseña para acceder a toda la documentación.



En esta última opción se pueden encontrar 2 versiones: soft float o hard float. Para saber cuál es soportada por el dispositivo, se deben seguir los siguientes pasos:

- Información de la CPU: para ello se ejecuta la siguiente instrucción mostrada a continuación→ cat /proc/cpuinfo

La información proporcionada por el terminal es:

```
root@odroid:/home/odroid# cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 3 (v7l)
BogoMIPS        : 84.00
Features        : swp half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 3

processor       : 1
model name      : ARMv7 Processor rev 3 (v7l)
BogoMIPS        : 84.00
Features        : swp half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 3

processor       : 2
model name      : ARMv7 Processor rev 3 (v7l)
BogoMIPS        : 84.00
Features        : swp half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 3

processor       : 3
model name      : ARMv7 Processor rev 3 (v7l)
BogoMIPS        : 84.00
Features        : swp half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 3

processor       : 4
model name      : ARMv7 Processor rev 3 (v7l)
BogoMIPS        : 120.00
Features        : swp half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
CPU implementer : 0x41
```

- Librerías a usar: ldd /bin/ls

  Una vez ejecutado este comando, el resultado es:

```
root@odroid:/home/odroid# ldd /bin/ls
        libselinux.so.1 => /lib/arm-linux-gnueabihf/libselinux.so.1 (0xb6eb2000)
        libc.so.6 => /lib/arm-linux-gnueabihf/libc.so.6 (0xb6dc6000)
        /lib/ld-linux-armhf.so.3 (0xb6f14000)
        libpcre.so.3 => /lib/arm-linux-gnueabihf/libpcre.so.3 (0xb6d69000)
        libdl.so.2 => /lib/arm-linux-gnueabihf/libdl.so.2 (0xb6d56000)
        libpthread.so.0 => /lib/arm-linux-gnueabihf/libpthread.so.0 (0xb6d32000)
```

Si esta información aparece tal y como se muestran las imágenes anteriores, el soporte adecuado para la instalación de la cámara es el hard float.

El archivo de instalación (.rar) se guarda en una carpeta y se ejecuta el comando:

tar xvf uEyeSDK-4.90.00-ARM_LINUX_IDS_GNUEABI_HF.tgz -C /

Este comando descomprime el archivo en el directorio raíz. De esta forma se encuentra fácilmente localizable. En los sistemas Debian o Ubuntu, se descargan los siguientes paquetes, siendo usuario root:

```
sudo apt-get install libcap-dev
```

```
apt-get install libqt4-qt3support libqtgui4 libqt4-network libqtcore4
libqt4-xml libqt4-sql libjpeg62 libpng12-0
```

A continuación, se ejecuta el script de instalación con la orden:

```
/usr/local/share/ueye/bin/ueyesdk-setup.sh
```

```
ldconfig
```

Este comando debe ejecutarse sin la cámara conectada. Con el comando ldconfig se crean los vínculos y caché necesarios a las bibliotecas compartidas que se encuentren en los distintos directorios de la línea de órdenes. Una vez instalada la cámara, se ejecutarán los siguientes comandos dependiendo si se va a usar mediante USB o IP:

```
USB daemon: /etc/init.d/ueyeusbdrc start
ETH daemon: /etc/init.d/ueyeethdrc start
```

En este caso, el tipo de instalación ha sido realizado mediante USB.

Con este último paso ya se habría instalado la UEYE XS 2 en la OdroidXU4. Ahora sólo falta probar los distintos códigos ejecutables que vienen por defecto en el directorio instalado. Cambiando al directorio:

```
cd /usr/local/share/ueye/bin/
```

Uno de los archivos es el llamado ‘./idscameramanager’, que permite al usuario configurar los parámetros principales de la cámara tal y como se muestra a continuación:

Uno de los parámetros principales es el 'ID' de la cámara, que servirá como identificador a la hora de utilizar el código para su programación. Una vez configurada, basta con ejecutar el comando './ueyedemo', que muestra la interfaz para capturar imágenes y realizar vídeos con la cámara.

Por último, cabe destacar que no se ha utilizado la cámara mediante la interfaz gráfica, sino que se han programado las distintas funciones de configuración a partir de las librerías proporcionadas. Debido a que existen multitud de parámetros de configuración de la cámara, la interfaz gráfica permite obtener un archivo .ini de configuración. De esta forma, el usuario puede tener una noción de qué parámetros son los más importantes a la hora de utilizar las funciones necesarias. Estas se encuentran explicados en el apartado de adquisición de imágenes.

Por último, para compilar los códigos de configuración de la cámara es necesario incluir la librería tal y como se indica en el siguiente ejemplo:

```
g++ -o archivo_ejecutable archivo.cpp -lueye_api
```

INSTALACIÓN DE OPCUA

Este anexo consiste en la instalación del servidor OPCUA en el entorno Linux previamente instalado (ver ANEXO A). Los pasos detallados que se describen a continuación se han obtenido tras descargar los paquetes y librerías necesarias en el siguiente enlace:

https://github.com/open62541/open62541.git

Primero es necesario instalar las dependencias necesarias para instalar la librería. Situándose en el directorio raíz se escribe el comando:

```
$ sudo apt-get install git build-essential gcc pkg-config
cmake python python-six
```

Se instalan las dependencias necesarias para la interfaz gráfica del ccmake, así como para el cifrado, múltiples hilos y la ejecución de test unitarios:

```
sudo apt-get install cmake-curses-gui
sudo apt-get install libmbedtls-dev
sudo apt-get install liburcu-dev
sudo apt-get install check
```

Para la generación y estilo de la documentación:

```
sudo apt-get install python-sphinx graphviz
sudo apt-get install python-sphinx-rtd-theme
```

A continuación, se clona dentro del directorio raíz el archivo procedente del enlace anteriormente mencionado.

```
git clone https://github.com/open62541/open62541.git
```

La compilación de la librería se realiza utilizando el procedimiento habitual seguido en los anexos anteriores, donde se empleaba el cmake:

```
$ cd open62541

$ mkdir build

$ cd build

$ cmake ..

$ ccmake ..
```

Durante el proceso de instalación, se abre una ventana emergente donde hay que activar la opción de UA_ENABLE_AMALGAMATION, que permite obtener toda la librería en dos únicos ficheros: open62541.c y open62541.h.

Una vez haya realizado este paso, es recomendable guardar esta configuración y se compila la librería con el comando:

```
$make
```

Una vez finalizado el proceso de compilación, se puede comenzar a programar entrando en el directorio donde se encuentran los dos archivos 'open62541'.

Para compilar los proyectos de OPCUA que se realicen, es necesario ejecutar la siguiente línea de comandos:

```
$ gcc -std=c99 open62541.c archivo.cpp -o archivo_objeto -lpthread
```

- El archivo open62541.c contiene la librería open62541.h
- Archivo.cpp: nombre del archivo que contiene todo el código que el usuario haya programado.
- Archivo_objeto: nombre del archivo ejecutable
- -std=c99: el 'C99' se trata de una versión anterior del estándar del lenguaje de programación C, y ayuda a las implementaciones a hacer un mejor uso del hardware en ordenador.

CÓDIGOS

-CÁMERA:

```
#include <string>
#include <iostream>
#include <thread>
#include <unistd.h>
#include "uEye.h"
#include "ueye_deprecated.h"
#include <stdio.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>
#include <time.h>
#include <ctime>
#define RED     "\x1b[31m"                          //Impresion del texto en color Rojo
#define GREEN   "\x1b[32m"                          //Impresion del texto en color Verde
#define YELLOW  "\x1b[33m"                          //Impresion del texto en color Amarillo
#define BLUE    "\x1b[34m"                          //Impresion del texto en color Azul
#define MAGENTA "\x1b[35m"                          //Impresion del texto en color Magenta
#define CYAN    "\x1b[36m"                          //Impresion del texto en color Cian
#define RESET   "\x1b[0m"                           //Impresion del texto en color Normal
#define tresh 100
#define max_tresh 255
#define PATH "/usr/local/share/ueye/bin"            //Define el Path donde se guardara la imagen
#define IMAGEN "/Resultado.jpg"                     //Nombre de la imagen de la que tomara el OpenCV
using namespace std;

class Camera
{
public:

    //void initCamera(HIDS* hCam_internal);                                 //Configura las variables y parametros necesarios para el uso de funciones
    void adquisition(HIDS* hCam_internal, int* memID_int);                      //Funcion que realiza la deteccion de las monedas detectadas
    //void savephoto(HIDS* hCam_internal);                                  //Guarda la imagen en la ruta predefinida. Despues la tratara el OpenCV
    void initializeCameraInterface(HIDS* hCam_internal);
};
```

```
using namespace std;
void Camera::initializeCameraInterface(HIDS* hCam_internal){
    int nMemoryId;//nuevo
    int colorMode;//NUEVO
    INT nRet = is_InitCamera (hCam_internal, NULL);
    if (nRet == IS_SUCCESS){
        cout << "Camera initialized!" << endl;
    }
    UINT nPixelClockDefault =21;
    nRet = is_PixelClock(*hCam_internal, IS_PIXELCLOCK_CMD_SET, (void*)&nPixelClockDefault, sizeof(nPixelClockDefault));

    if (nRet == IS_SUCCESS){
        cout << "Camera pixel clock succesfully set!" << endl;
    }else if(nRet == IS_NOT_SUPPORTED){
        cout << "Camera pixel clock setting is not supported!" << endl;
    }

    colorMode =IS_COLORMODE_BAYER;

    nRet = is_SetColorMode(*hCam_internal,colorMode);

    if (nRet == IS_SUCCESS){
        cout << "Camera color mode succesfully set!" << endl;
    }
}
```

```
void Camera::adquisition(HIDS* hCam_internal, int* memID_int){
    unsigned t,t1; //definicion de tiempos
       // ------------------------------------------------------------------------------------------------------------------
      cout<<"INICIO..."<<endl;
       sleep(1);
      t=clock();
      //t=getcurrent
      HIDS hCam = 0;
      int memID = 0;
      char* pMem=NULL;
     initializeCameraInterface(&hCam);//LLAMADA A LA CONFIGURACION DE LA C?MARA
      //is_AllocImageMem(hCam, 2500, 2500, 32, &pMem, &memID);//1980x1980
      is_AllocImageMem(hCam, 1280, 720 ,16, &pMem, &memID);//1980x1080
      INT nRet = is_SetImageMem(hCam, pMem, memID);
      cout<<"SE VA A INICIAR EL PROGRAMA..."<<endl;
           INT displayMode = IS_SET_DM_DIB;
           nRet=is_SetDisplayMode (hCam, displayMode);
           printf("Status displayMode %d\n",nRet);
           nRet = is_FreezeVideo(hCam, IS_WAIT);
           cout<<"Status is_FreezeVideo"<<nRet<<endl;//El encendido de la camara ha tenido exito
           IMAGE_FILE_PARAMS ImageFileParams;
           //INT is_SaveImage(hCam,"/usr/local/share/ueye/bin/imagen.jpeg");
           ImageFileParams.pwchFileName =L"./imagen.png";
           ImageFileParams.pnImageID = NULL;
           ImageFileParams.ppcImageMem = NULL;
           ImageFileParams.nQuality = 98;
           ImageFileParams.nFileType = IS_IMG_PNG;
           nRet = is_ImageFile(hCam, IS_IMAGE_FILE_CMD_SAVE, (void*) &ImageFileParams, sizeof(ImageFileParams));
           cout<<"Status is_ImageFile"<<nRet<<endl;
           ImageFileParams.pwchFileName =L"./imagen.bmp";
           ImageFileParams.pnImageID = NULL;
           ImageFileParams.ppcImageMem = NULL;
           ImageFileParams.nQuality = 98;//Se establece una calidad del 98%
           ImageFileParams.nFileType = IS_IMG_BMP;
           nRet = is_ImageFile(hCam, IS_IMAGE_FILE_CMD_SAVE, (void*) &ImageFileParams, sizeof(ImageFileParams));
           cout<<"Status is_ImageFile"<<nRet<<endl;
           ImageFileParams.pwchFileName =L"./imagen.jpg";
           ImageFileParams.pnImageID = NULL;
           ImageFileParams.ppcImageMem = NULL;
           ImageFileParams.nQuality = 98;
           ImageFileParams.nFileType = IS_IMG_JPG;
           nRet = is_ImageFile(hCam, IS_IMAGE_FILE_CMD_SAVE, (void*) &ImageFileParams, sizeof(ImageFileParams));
           cout<<"Status is_ImageFile"<<nRet<<endl;
           t1=clock();
```

-OPENCV:

```
    //---------------------------------------------------------------------------
    //                          DECLARACION DE LA CONSTANTES
    //---------------------------------------------------------------------------
    RNG rng;
    ofstream archivo;                          //Archivo donde se guardaran los datos
    Mat src,src_gray,gray,gauss,original,canny,salida,image_recortada;
    Mat canny_output,fondo,result,diff,dst;
    /*
    int cont_m1;                                        //Contador de monedas de 1 centimo
    int cont_m2;                                        //Contador de monedas de 2 centimos
    int cont_m3;                                        //Contador de monedas de 5 centimos
    int cont_m4;                                        //Contador de monedas de 10 centimos
    int cont_m5;                                        //Contador de monedas de 20 centimos
    int cont_m6;                                        //Contador de monedas de 50 centimos
    int cont_m7;                                        //Contador de monedas de 1 euro
    int cont_m8;                                        //Contador de monedas de 2 euros
    */

public:
    //OpenCV(); //SE QUEDA SIEMPRE COMENTADO
    void savedFiles();                                  //Configura las variables y parametros necesarios para el uso de funciones
    void detection(int *cont_m1,int *cont_m2,int *cont_m3,int *cont_m4,int *cont_m5,int *cont_m6,int *cont_m7,int *cont_m8);


};
```

```
    for(int i = 0; i < circles.size(); i++ )
    {
          sprintf(buff,"%s" "%d","Moneda",i+1);
          Point center(cvRound(circles[i][0]), cvRound(circles[i][1]));//Estos valores son los que se pasaran al servidor
      Point a(-30,40);
      float radius = cvRound(circles[i][2]); //Estos valores son los que se pasaran al servidor
      float diametro=radius*2*0.151;
      vect_diam[i]=diametro;//se me va rellenando el vector
      circle( image_recortada, center, 3, Scalar(0,255,0), -1, 8, 0 );// Centro del Circulo
      circle( image_recortada, center, radius, Scalar(0,0,255), 2.5, 8,0);//Linea del circulo
      putText(image_recortada, buff,center-a,FONT_HERSHEY_SIMPLEX,0.5,CV_RGB(0,0,255),2 );//Seria interesante que cada circulo

      if(vect_diam[i]>=15 && vect_diam[i]<=16.5){//comparobacion de monedas de 1 centimo
                (*cont_m1)++;                                       //Moneda de 1 centimos
                cout << "MONEDA " <<i+1<< "es de 1 CENTIMO: "<< "Diametro : \n" << diametro << endl;
            }
          else if(vect_diam[i]>16.5 && vect_diam[i]<=19.33){
                (*cont_m2)++;                                       //Moneda de 2 centimos
                cout << "MONEDA " <<i+1<< "es de 2 CENTIMOS: "<< "Diametro : \n" << diametro << endl;
          }
          else if(vect_diam[i]>19.33 && vect_diam[i]<=21.6){
                (*cont_m3)++;                                       //Moneda de 5 centimos

                cout << "MONEDA " <<i+1<< "es de 5 CENTIMOS: "<< "Diametro : \n" << diametro << endl;
          }

          //}
          else if(vect_diam[i]>21.6 && vect_diam[i]<=22.75){
                (*cont_m5)++;                                       //Moneda de 20 centimos
                archivo<<vect_diam[i]<<endl;
               cout << "MONEDA " <<i+1<< "es de 20 CENTIMOS: "<< "Diametro : \n" << diametro << endl;
          }



/*----------ALGORITMO DE DETECCION DE MONEDAS-------------*/
void OpenCV::detection(int *cont_m1,int *cont_m2,int *cont_m3,int *cont_m4,int *cont_m5,int *cont_m6,int *cont_m7,int *cont_m8){
   *cont_m1=*cont_m2=*cont_m3=*cont_m4=*cont_m5=*cont_m6=*cont_m7=*cont_m8=0;
    archivo.open("fichero.txt");
    //archivo.open("Datos.txt",ios::out); //Abrimos el archivo. Si no existe lo crea
    //archivo.clear();                    //Hacemos el borrado del archivo
    vector <Vec3f> circles;
    vector<vector<Point>>contornos;
    vector<Vec4i> hierarchy;
     original=imread("p34.png",CV_LOAD_IMAGE_COLOR);
    original = imread("/usr/local/share/ueye/bin/tfm/imagen.jpg",CV_LOAD_IMAGE_COLOR);
    //Abre la imagen que se encuentra en la ruta de la variable buffer
    //Por lo tanto, a partir de ese punto se define una zona de 100 pixeles de ancho por 150 de alto.
    Rect myrect(250,10,900,700);//Definimos un rectangulo para la region de interes:Establece una región que comenzara en el punto de la coordenada
    Mat image_recortada=original(myrect);

    namedWindow( "Imagen Recortada", CV_WINDOW_AUTOSIZE );
    imshow("Imagen Recortada", image_recortada);
    cvtColor(image_recortada,gray, COLOR_BGR2GRAY);
    GaussianBlur(gray, gauss,Size(3,3),0,0);                          //Se aplica un suavizado Gaussiano
    imshow("suavizado", gauss);

    threshold(gray, diff, 45, 100,0 );
    imshow("Resta", diff);
    Canny(gauss,canny_output, 25,50);
    namedWindow( "Canny", CV_WINDOW_AUTOSIZE );
    imshow("Canny", canny_output);
    HoughCircles( gauss, circles, CV_HOUGH_GRADIENT, 1, 30, 100, 30, 50, 100);//Se usa la imagen gauss
    cout << "Numero de Monedas Detectadas : " << circles.size()<<endl;//Mostramos el número de monedas por consola


    //Nos dice el radio y las coordenadas del centro
    char buff[100];
    static int tipo;
    findContours(gray, contornos, hierarchy,  RETR_EXTERNAL,CV_CHAIN_APPROX_SIMPLE);         //Buscamos los contornos
    cout<<"Numero de contornos: "<<contornos.size()<<endl;
    namedWindow( "Contours", CV_WINDOW_AUTOSIZE );
    float vect_diam[circles.size()];
    float vect_centro[circles.size()];


      namedWindow("Dibujo",CV_WINDOW_AUTOSIZE);
      imshow("Dibujo", image_recortada );
      threshold(gray, dst, 200, 100,THRESH_BINARY );

     findContours(canny_output, contornos, hierarchy, CV_RETR_TREE,CV_CHAIN_APPROX_SIMPLE , Point(0,0));
     Mat drawing = Mat::zeros(canny_output.size(), CV_8UC3);
      for( int i = 0; i< contornos.size(); i++ )
        {
          Scalar color = Scalar( rng.uniform(0, 255), rng.uniform(0,255), rng.uniform(0,255) );
          drawContours( drawing, contornos, i, color, 2, 8, hierarchy, 0, Point() );
        }
    imshow( "Contours", drawing );
    archivo.close();                                                  //Se cierra el archivo por seguridad
    //Se guardan los resultados en archivos
    imwrite ( "/usr/local/share/ueye/bin/Resultado.jpg",image_recortada );
    imwrite ( "/usr/local/share/ueye/bin/gray_image.jpg",gray );
```

-CLIENTE OPC-UA:

```
class Cliente_Opcua{
//-----------------------------------------------------------------------------
//                      VARIABLES
//-----------------------------------------------------------------------------

public:
         UA_ClientConfig config;
         UA_Client *client;
         UA_Variant value, value1,value2,value3,value4,value5,value6,value7;

public:
//-----------------------------------------------------------------------------
//                      M É T O D O S   P Ú B L I C O S
//-----------------------------------------------------------------------------

     ///Client_Opcua ();
    //! @brief Constructor
    //! @param <Poner los parametros necesarios>
    //! @return <none>
    //   Cliente_Opcua();
     Cliente_Opcua ();
    //! @brief conexión servidor Opcua
    //! @param <Poner los parametros necesarios>
    //! @return <none>
    UA_StatusCode cliente_Conection();
    //! @brief Lee los valores de los tgas del servidor
    //! @param <Poner los parametros necesarios>
    //! @return <none>
        void read_Tags();
    //! @brief Libera de memoria los contenedores de los tags
    //! @param <Poner los parametros necesarios>
    //! @return <none>
    void libera_Contenedores();
    //! @brief Desconecta el cliente
    //! @param <Poner los parametros necesarios>
    //! @return <none>
    void desconecta_Cliente();
```

```
    void Cliente_Opcua::read_Tags(){
    UA_StatusCode status_client;

/*********************Lectura Tags del Servidor Según Nodos Del Servidor***********************/
/*********************E Impresión en Pantalla del Cliente**************************************/
/**********************************************************************************************/
cout<<"                 Cliente Opcua:Sistema Vision Artificial"<<endl;
cout<<"                 ============================================="<<endl;
cout<<"                 TFM: David Castillo Saez               "<<endl;
cout<<"Tags del Servidor"<<endl;
cout<<"================ "<<endl;
     //Inicializa el contenedor y lee el tag
    UA_Variant_init(&value);
    /*Lee el tag del nodo del servidor "monedas 1 centimo y lo muestra"*/
    status_client = UA_Client_readValueAttribute(client, UA_NODEID_NUMERIC(1,1), &value);
     if(status_client == UA_STATUSCODE_GOOD &&
    UA_Variant_hasScalarType(&value, &UA_TYPES[UA_TYPES_INT32])){
    cout<<"Monedas 1 centimo: "<< *(UA_Int32*)value.data<<endl;

   }
     UA_Variant_init(&value1);
    status_client = UA_Client_readValueAttribute(client, UA_NODEID_NUMERIC(1, 2), &value1);
cout<<status_client<<endl;
    if(status_client == UA_STATUSCODE_GOOD &&
    UA_Variant_hasScalarType(&value1, &UA_TYPES[UA_TYPES_INT32])) {
    cout<<"Monedas 2 centimos: "<<*(UA_Int32*) value1.data<<endl;
    }
    UA_Variant_init(&value2);
    status_client = UA_Client_readValueAttribute(client, UA_NODEID_NUMERIC(1,5), &value2);
    if(status_client == UA_STATUSCODE_GOOD &&
    UA_Variant_hasScalarType(&value2, &UA_TYPES[UA_TYPES_INT32])) {
    cout<<"Monedas 5 centimos: "<<*(UA_Int32*) value2.data<<endl;
    }


  if(status_client == UA_STATUSCODE_GOOD &&
  UA_Variant_hasScalarType(&value6, &UA_TYPES[UA_TYPES_INT32])) {
  cout<<"Monedas 1 euro: "<<*(UA_Int32*) value6.data<<endl;
  }

  UA_Variant_init(&value7);
  status_client = UA_Client_readValueAttribute(client, UA_NODEID_NUMERIC(1, 10007), &value7);

  if(status_client == UA_STATUSCODE_GOOD &&
  UA_Variant_hasScalarType(&value7, &UA_TYPES[UA_TYPES_INT32])) {
  cout<<"Monedas 2 euros: "<<*(UA_Int32*) value7.data<<endl;
  }


  usleep(1000000);
  }

/ Libera los contenedores
 void Cliente_Opcua::libera_Contenedores(){
  UA_Variant_deleteMembers(&value);
  UA_Variant_deleteMembers(&value1);
  UA_Variant_deleteMembers(&value2);
  UA_Variant_deleteMembers(&value3);
  UA_Variant_deleteMembers(&value4);
  UA_Variant_deleteMembers(&value5);
  UA_Variant_deleteMembers(&value6);
  UA_Variant_deleteMembers(&value7); }
 //desconecta el cliente
  void Cliente_Opcua::desconecta_Cliente(){
  UA_Client_delete(client);}
```

```
//Variables globales
UA_Boolean running = true;
UA_Logger logger = UA_Log_Stdout;
//Finalizacion del cliente con Ctrl-C
static void stopHandler(int sign) {
    UA_LOG_INFO(logger, UA_LOGCATEGORY_CLIENT, "Received Ctrl-C");
    running = 0;
}


 int main(){
//Variables
signal(SIGINT, stopHandler);
//Declara una clase Cliente_Opcua
Cliente_Opcua cliente1;
//Bucle infinito de lectura de tagas y reconexión si se interrumpe
while(running){
system("clear");//Limpia Pantalla
estado=cliente1.cliente_Conection();//Conecta el cliente
if(estado != UA_STATUSCODE_GOOD) {//Si Falla Conexión Parpadea y Avisa
            UA_LOG_ERROR(logger, UA_LOGCATEGORY_CLIENT, "Not connected. Retrying to connect in 1 second");
            cliente1.desconecta_Cliente();
            usleep(100000);

        }
 else{//si la conexion es buena lee los tags
     UA_LOG_ERROR(logger, UA_LOGCATEGORY_CLIENT, "Conectado con éxito al servidor");
            usleep(100000);//Intevalo de Actualización ns
            cliente1.read_Tags();
            cliente1.libera_Contenedores();
            cliente1.desconecta_Cliente();

}
};
cliente1.libera_Contenedores();
cliente1.desconecta_Cliente();
return 0;}
```

-SERVIDOR OPC-UA:

```
class Server_Opcua
{
public:
   // Server_Opcua(); //NO DESCOMENTAR ESTO
    //! @brief Construye la Server.
    //! @param <none>
    //! @return <none>

    //-------------------DEFINICIÓN DE VARIABLES PUBLICAS Y COMPARTIDAS-----


   // static UA_Server *server;
    //static UA_Boolean running;
    //static mutex *mutex_programa;
    bool conexion;                  //1 si el servidor esta conectado, 0 si no esta conectado.
    bool server_connected;          //1 si el servidor se encuentra conectado, 0 si no se encuentra conectado.
    bool client_connected;          //1 si el cliente se encuentra conectado, 0 si no se encuentra conectado.
    bool photo_adquisition;         //1 si se esta realizando la captura de una imagen, 0 si no.
    unsigned t0,t1;                 //Definicion de los tiempos del programa (inicio y fin de proceso)
    //UA_Boolean running;
    UA_VariableAttributes attr,attr1,attr2,attr3,attr4,attr5,attr6,attr7,attr8;
    UA_Variant value1,value2,value3,value4,value5,value6,value7,value8;



    //! @brief Destruye la Server.
    //! @param <none>
    //! @return <none>

//-----------------------------------------------------------------------
//                    M É T O D O S   P Ú B L I C O S
//-----------------------------------------------------------------------

     //virtual ~Server_Opcua();
    //! @brief Inicializa la aplicación del servidor
    //! @param <Poner los parametros necesarios>
    //! @return <none>
      void initConfiguration();

    //! @brief Finaliza la aplicación del servidor
    //! @param <none>
    //! @return <none>
      void closeServer(void);

    //! @brief Realiza la conexion del servidor
    //! @param <Se indicara tanto el puerto como la direccion IP>
    //! @return <none>
      bool doConexion();

    //! @brief Añade los distintos Tags al servidor OPCUA
    //! @param <Habra que especificar el puerto y la IP de la conexion>
    //! @return <none>
      void AddTags(UA_Int32 *cont_m1,UA_Int32 *cont_m2,UA_Int32 *cont_m3,UA_Int32 *cont_m4,UA_Int32 *cont_m5,UA_Int32 *cont_m6,UA_Int32 *cont_m7,UA_Int

    //! @brief Asigna los Tags a las variables de OpenCV
    //! @param <none>
    //! @return <none>
      void TagsAssigment(UA_Int32 *cont_m1,UA_Int32 *cont_m2,UA_Int32 *cont_m3,UA_Int32 *cont_m4,UA_Int32 *cont_m5,UA_Int32 *cont_m6,UA_Int32 *cont_m7,

    //! @brief Actualiza lost tags despues del conteo de monedas
     //! @param <none>
     //! @return <none>
      void TagsUpdate(int cont_m1,int cont_m2,int cont_m3,int cont_m4,int cont_m5,int cont_m6,int cont_m7,int cont_m8);
```

```
//------------------------------------------------------------------------------
//                           DEFINICION DE CONSTANTES
//------------------------------------------------------------------------------
using namespace std;
UA_ServerConfig *config=UA_ServerConfig_new_default();
UA_Server *server=UA_Server_new(config);
UA_Boolean running=true;
void Server_Opcua::closeServer(){

    UA_Server_delete(server);
    UA_ServerConfig_delete(config);//Despues de conectar el servidor
}
bool Server_Opcua::doConexion(){

    UA_StatusCode status = UA_Server_run(server, &running);

}


void Server_Opcua::initConfiguration(){

        //Comprobamos si se ha realizado la conexion con exito
        if(conexion==1){
            server_connected=1;
              cout<<"GPIOs cargados con exito"<<endl;
              cout<<"Camara inicializada con exito"<<endl;
        }
        else{
            cout<<"No se ha podido realizar la configuracion de los dispositivos"<<endl;
            sleep(0.3);
            cout<<"Reintentando conexion..."<<endl;
            doConexion();//Llamada a realizar la conexion del servidor
            }
        }
```

```
void Server_Opcua::AddTags(UA_Int32 *cont_m1,UA_Int32 *cont_m2,UA_Int32 *cont_m3,UA_Int32 *cont_m4,UA_Int32


    /* Añade tag monedas de 1 centimo a un nodo del servidor*/
    UA_VariableAttributes attr1 = UA_VariableAttributes_default;
    attr1.displayName = UA_LOCALIZEDTEXT("en-US", "monedas 1 centimo");
    /*al ser el nivel de acceso de lectura y escritura, el cliente puede escribir y modificar el valor*/
    attr1.accessLevel=UA_ACCESSLEVELMASK_READ|UA_ACCESSLEVELMASK_WRITE;
    UA_Int32 monedas_1 = 0;
    UA_Variant_setScalar(&attr1.value, &monedas_1, &UA_TYPES[UA_TYPES_INT32]);
    UA_NodeId newNodeId1 = UA_NODEID_STRING(1, "monedas 1 centimo");
    UA_NodeId parentNodeId1 = UA_NODEID_NUMERIC(0, UA_NS0ID_OBJECTSFOLDER);
    UA_NodeId parentReferenceNodeId1 = UA_NODEID_NUMERIC(0, UA_NS0ID_ORGANIZES);
    UA_NodeId variableType1 = UA_NODEID_NULL; /* take the default variable type */
    UA_QualifiedName browseName1 = UA_QUALIFIEDNAME(1, "monedas 1 centimos");
    UA_Server_addVariableNode(server, newNodeId1, parentNodeId1, parentReferenceNodeId1,
    browseName1, variableType1, attr1, NULL, NULL);
    /* Añade tag monedas de 2 centimos a un nodo del servidor*/

    UA_VariableAttributes attr2 = UA_VariableAttributes_default;
    attr2.displayName = UA_LOCALIZEDTEXT("en-US", "monedas 2 centimos");
    attr2.accessLevel=UA_ACCESSLEVELMASK_READ|UA_ACCESSLEVELMASK_WRITE;
    /* inicializa el tag*/
    UA_Int32 monedas_2 = 0;
    UA_Variant_setScalar(&attr2.value, &monedas_2, &UA_TYPES[UA_TYPES_INT32]);
    UA_NodeId newNodeId2 = UA_NODEID_STRING(1, "monedas 2 centimos");
    UA_NodeId parentNodeId2 = UA_NODEID_NUMERIC(0, UA_NS0ID_OBJECTSFOLDER);
    UA_NodeId parentReferenceNodeId2 = UA_NODEID_NUMERIC(0, UA_NS0ID_ORGANIZES);
    UA_NodeId variableType2 = UA_NODEID_NULL;
    UA_QualifiedName browseName2 = UA_QUALIFIEDNAME(1, "monedas 2 centimos");
    UA_Server_addVariableNode(server, newNodeId2, parentNodeId2, parentReferenceNodeId2,
    browseName2, variableType2, attr2, NULL, NULL);

    /* Añade tag monedas de 5 centimos a un nodo del servidor*/
    UA_VariableAttributes attr3 = UA_VariableAttributes_default;
    attr3.displayName = UA_LOCALIZEDTEXT("en-US", "monedas 5 centimos");
    attr3.accessLevel=UA_ACCESSLEVELMASK_READ|UA_ACCESSLEVELMASK_WRITE;
```

```
//Declaramos contenedores
//UA_Variant value1;
UA_Variant_init(&value1);
UA_Int32  c1=(UA_Int32)(*cont_m1);

//UA_Variant value2;
UA_Variant_init(&value2);
UA_Int32 c2=(UA_Int32)(*cont_m2);

//UA_Variant value3;
UA_Variant_init(&value3);
UA_Int32 c3=(UA_Int32)(*cont_m3);

//UA_Variant value4;
UA_Variant_init(&value4);
UA_Int32 c4=(UA_Int32)(*cont_m4);

//UA_Variant value5;
UA_Variant_init(&value5);
UA_Int32  c5=(UA_Int32)(*cont_m5);

//UA_Variant value6;
UA_Variant_init(&value6);
UA_Int32 c6=(UA_Int32)(*cont_m6);

//UA_Variant value7;
UA_Variant_init(&value7);
UA_Int32 c7=(UA_Int32)(*cont_m7);

//UA_Variant value8;
UA_Variant_init(&value8);
UA_Int32  c8=(UA_Int32)(*cont_m8);
```