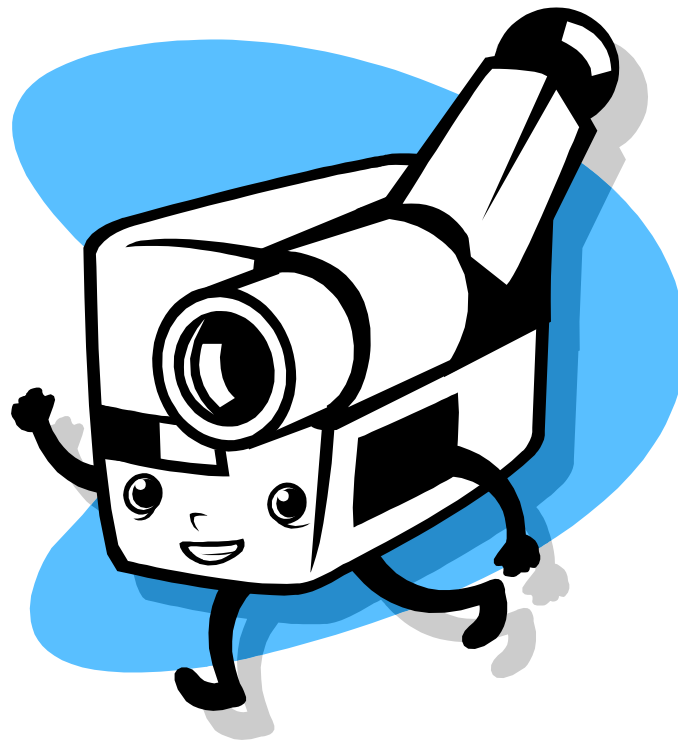


RESUM PROJECTE FINAL DE CARRERA



TÍTOL: Adquisició de senyal de vídeo multicàmera per a una smartroom

AUTOR: Jordi Cenzano Ferret

DIRECTOR: Josep R. Casas Pla

DATA: 21-04-2005

1 Introducció

El que se'ns demana en aquest projecte és dissenyar i implementar un sistema real (no és cap simulació) d'adquisició multicàmera. El sistema ha de mantenir el millor possible la sincronia entre les diferents càmeres, ha de ser capaç de capturar en la màxima resolució que es pugui aconseguir dins el pressupost donat.

A més, aquest sistema ha de poder controlar una càmera activa (càmera mòbil motoritzada) automàticament, podent memoritzar i recuperar posicions.

El control del sistema ha de ser el més centralitzat i automatitzat possible, mantenint una bona escalabilitat.

En la Figura 1-1 veiem el diagrama de blocs inicial del sistema.

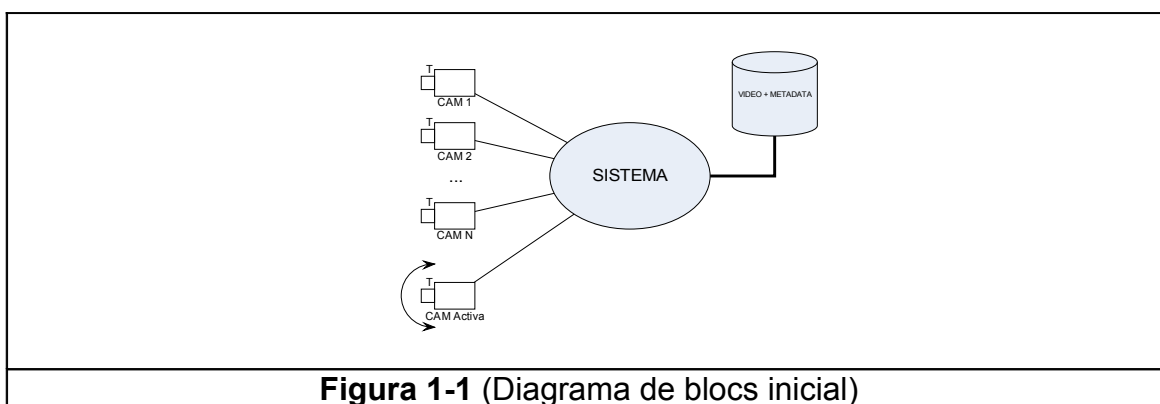


Figura 1-1 (Diagrama de blocs inicial)

2 Selecció de l'equipament hardware necessari

El primer pas va ser fer un estudi dels pros i contres que ens oferien les diverses opcions hardware que varem trobar en el mercat: tipus de càmeres, òptiques, opcions de sincronització, targetes d'adquisició, timestamping, etc...

Després de valorar diverses arquitectures del sistema, varem optar per utilitzar un multicàmera sincronitzat al frame amb 1 PC per càmera que s'ocupés d'adquirir el senyal de vídeo i fer-hi el timestamping corresponent. Tot el sistema estaria controlat per un servidor que operaria, ordenaria i mouria la gran quantitat de dades que el sistema genera.

Pel control de la càmera activa vam decidir utilitzar un hardware de control industrial i un hardware propi que varem dissenyar i desenvolupar.

A la Figura 2-2 veiem un diagrama de blocs hardware del sistema.

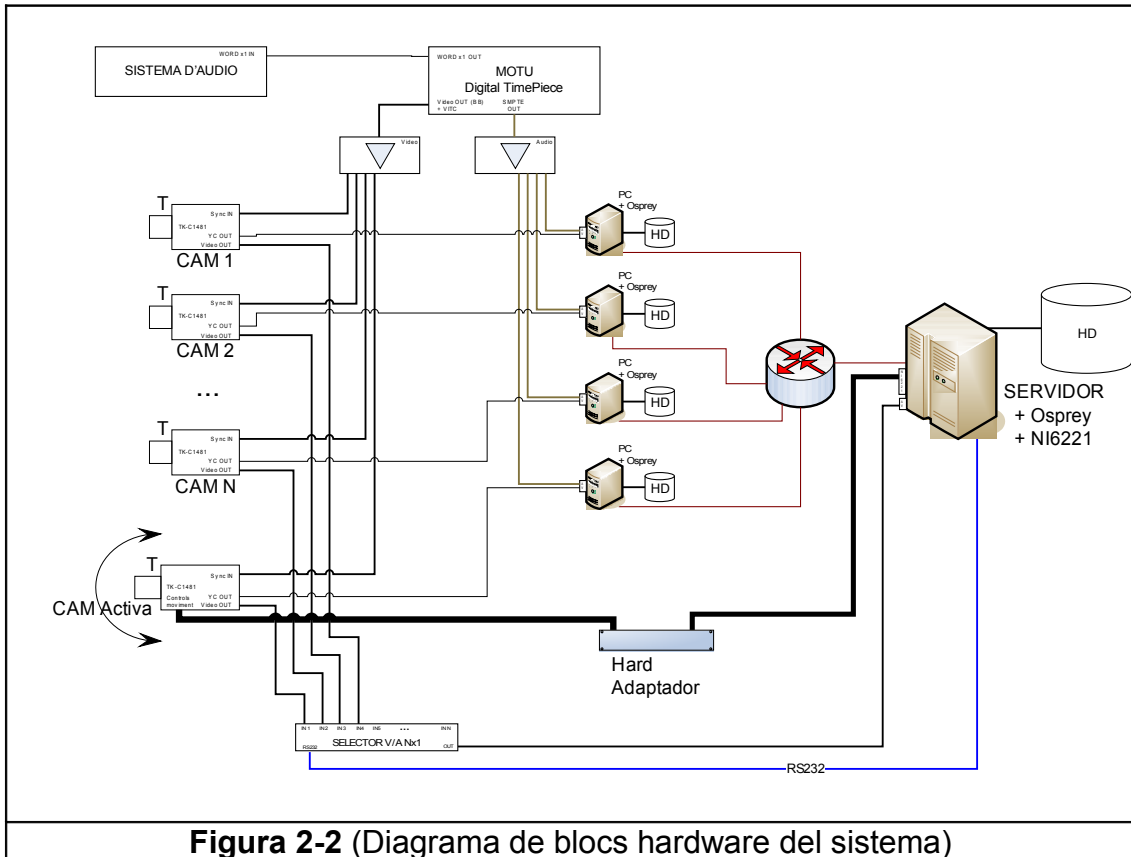


Figura 2-2 (Diagrama de blocs hardware del sistema)

El disseny hardware que varem implementar manté perfectament el requeriment inicial d'escalabilitat.

3 Disseny del software de control

El disseny del hardware ens proporciona el marc per treballar en el disseny del software que ha de controlar el sistema.

El software que varem dissenyar e implementar permet controlar les gravacions (engegar, parar, i ajustar diferents paràmetres) de tots els sistemes de captura. Incorpora també un sistema de gestió de les dades adquirides, a més facilita el calibratge de totes les càmeres des d'una consola central (la del servidor) a través d'un assistent gràfic.

En la Figura 3-3 veiem el diagrama de blocs del software del sistema.

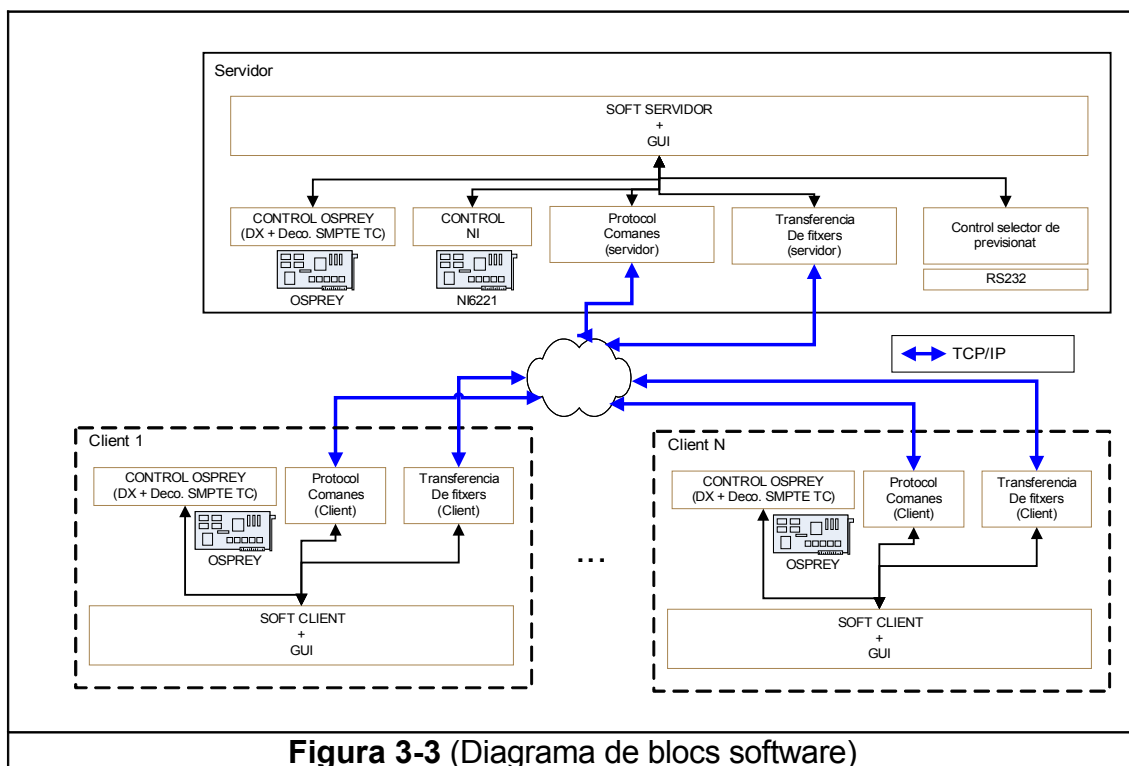


Figura 3-3 (Diagrama de blocs software)

3.1 Software client

El software client es una aplicació gràfica Win32 que està basada en els següents mòduls:

GUI (Graphical User Interface): l'hem programat amb C++ i es basa en la API de win32, utilitza MFC (Microsoft Foundation Classes).

Transferència de fitxers: és el bloc que controla la transferència de fitxers entre tots els PCs d'adquisició i el servidor central. Esta desenvolupat en C++ i està basat en sockets.

Protocol de comanes: s'encarrega de controlar i ajustar tot el sistema. És un protocol que vam desenvolupar específicament per aquest projecte, permet controlar N clients. Està fet amb C++ i basat en sockets.

Control Osprey: és el bloc més complex del sistema, controla la targeta d'adquisició (Osprey), descodifica el senyal de codi de temps (SMPTE LTC) que llegeix la Osprey per un canal d'àudio, etiqueta cada frame de vídeo amb el timestamp que li correspon. També permet codificar el vídeo en diferents formats: BMP, RAW (YUV i RGB), i MPEG2.

Està programat en C++, utilitza la tecnologia Directshow (de Microsoft) per accedir als dispositius multimèdia del sistema.

3.2 Software servidor

El bloc servidor també és una aplicació gràfica Win32 i està basada en les següents parts:

GUI (Graphical User Interface): aquesta part està programada amb Visual Basic .NET i és la que engloba tots els blocs.

Transferència de fitxers: igual que en el client (aquí implementa la part servidora del protocol TCP/IP).

Protocol de comanes: igual que en el client (aquí implementa la part servidora del protocol TCP/IP).

Control Osprey: igual que en el client (en aquest cas sols s'utilitza per visionar qualsevol senyal del sistema).

4 Conclusions

Creiem que hem pogut dissenyar i implementar un sistema que compleix la gran majoria dels objectius inicials, tot el sistema es pot controlar totalment des de la consola d'un sol PC (servidor), permet la captura síncrona de N càmeres, fa una gestió simple i eficient de les dades adquirides i té una molt bona escalabilitat.

Després de fer-hi moltes proves de precisió, eficiència, etc... la part del control de la càmera activa ha estat la que ha donat millors resultats, no sols permet moure la càmera, sinó que també guarda i recupera posicions amb menys de 1º d'incertesa.

La gran virtut d'aquest projecte crec que ha estat que, no només s'ha tractat d'un desenvolupament teòric, sinó que, a mesura que anàvem avançant, veiem com els nostres dissenys es traduïen en equipament real amb el que podíem experimentar i el qual realimentava en part els dissenys originals.

Una altre punt fort del projecte ha estat, segons el meu punt de vista, la gran diversitat de disciplines que ha tocat: vídeo, sincronització (LTC), programació (C++, VB .net, *sockets*), control de targeteria (Osprey 210, NI 6221), electrònica (adaptació de càmera activa).