UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

****

TRABAJO FIN DE MÁSTER

ViCON - Sistema de Visión Configurable aplicado a Reconocimiento Facial

MÁSTER EN SISTEMAS ELECTRÓNICOS PARA ENTORNOS INTELIGENTES

CHRISTIAN DIEGO COBOS MARCOS

MÁLAGA, 2025

E.T.S. DE INGENIERÍA DE TELECOMUNICACIÓN, UNIVERSIDAD DE MÁLAGA

**ViCON – Sistema de Visión Configurable aplicado a Reconocimiento Facial**

Autor: Christan Diego Cobos Marcos

Tutor: Martin González García

Departamento: Departamento de Tecnología Electrónica

Titulación: Máster en Sistemas Electrónicos para Entornos Inteligentes

Palabras clave: FPGA, VHDL, FT245, MT9V111, Basys3, Visión por Computador, Qt, OpenCV, Procesado de Imágenes en Tiempo Real.

**Resumen**

Este proyecto presenta el diseño e implementación de un sistema de adquisición y transmisión de imágenes en tiempo real, configurable y basado en FPGA. El sistema emplea una placa Xilinx Basys3 que integra un sensor de imagen MT9V111, una interfaz USB FT245 asíncrona y un módulo de control descrito en VHDL. La arquitectura incluye máquinas de estados finitos (FSM) y memorias FIFO para gestionar las solicitudes de datos desde un PC, así como la captura, sincronización y envío de datos de vídeo hacia el mismo.

En el lado software, se ha desarrollado una aplicación en C++/Qt, que realiza solicitudes de imagen, recibe los fotogramas transmitidos por la FPGA y los procesa mediante OpenCV, habilitando tareas de visualización, detección y reconocimiento facial. El diseño se validó mediante simulación y pruebas en placa, logrando la transmisión y reconstrucción fiable de imágenes en escala de grises a través de la interfaz USB.

Los resultados muestran la viabilidad de la solución como un sistema embebido de bajo coste para aplicaciones de visión por computador en tiempo real, destacando la modularidad de la arquitectura y la posibilidad de extenderla hacia entornos de Edge AI y sistemas inteligentes.

E.T.S. DE INGENIERÍA DE TELECOMUNICACIÓN, UNIVERSIDAD DE MÁLAGA

**ViCON – Configurable Vision System Applied to Face Recognition**

Author: Christian Diego Cobos Marcos

Supervisor: Martin González García

Department: Departamento de Tecnología Electrónica

Degree: Máster en Sistemas Electrónicos para Entornos Inteligentes

Keywords: FPGA, VHDL, FT245, MT9V111, Basys3, Computer Vision, Qt, OpenCV, Real-Time Image Processing.

**Abstract**

This project presents the design and implementation of a configurable, FPGA-based real-time image acquisition and transmission system. The system uses a Xilinx Basys3 board that integrates an MT9V111 image sensor, an asynchronous FT245 USB interface, and a control module described in VHDL. The architecture includes finite state machines (FSM) and FIFO memories to manage data requests from a PC, as well as the capture, synchronisation and sending of video data to it.

On the software side, a C++/Qt application has been developed, which request and image, receives the frames transmitted by the FPGA and processes them using OpenCV, enabling visualisation, detection and facial recognition tasks. The design was validated through simulation and board testing, achieving reliable transmission and reconstruction of grayscale images through the USB interface.

The results show the viability of the solution as a low-cost embedded system for real-time computer vision applications, highlighting the modularity of the architecture and the possibility of extending it to Edge AI environments and intelligent systems.

ÍNDICE

[1. Introducción 5](#_Toc120465433)

[1.1 Objetivos 5](#_Toc120465434)

[1.2 Entorno de usuarios 5](#_Toc120465435)

[1.3 Directivas del proyecto 5](#_Toc120465436)

[1.4 Alternativas y competencia 5](#_Toc120465437)

[1.5 Estructura de la memoria 6](#_Toc120465438)

[2 Requisitos y casos de uso 7](#_Toc120465439)

[2.1 Requisitos del proyecto 7](#_Toc120465440)

[2.1.1 Descripción de los Requisitos Funcionales 8](#_Toc120465441)

[2.1.2 Descripción de los Requisitos No Funcionales 8](#_Toc120465442)

[2.2 Casos de uso 8](#_Toc120465443)

[2.2.1 Descripción textual de los casos de uso 9](#_Toc120465444)

[3 Diseño y desarrollo del sistema 10](#_Toc120465445)

[3.1 Arquitectura Física 10](#_Toc120465446)

[3.1.1 … 10](#_Toc120465447)

[3.1.2 … 10](#_Toc120465448)

[3.1.3 Esquema hardware 10](#_Toc120465449)

[3.2 Arquitectura Lógica 10](#_Toc120465450)

[3.2.1 … 11](#_Toc120465451)

[3.2.2 … 11](#_Toc120465452)

[4 Resultados 12](#_Toc120465453)

[4.1 Verificación de requisitos 12](#_Toc120465454)

[a. Entorno 1 12](#_Toc120465455)

[<Prueba de verificación de requisito/s> 12](#_Toc120465456)

[b. Entorno x 12](#_Toc120465457)

[<Prueba de verificación de requisito/s> 12](#_Toc120465458)

[4.2 Resultados de las pruebas del sistema 13](#_Toc120465459)

[4.3 Plan de validación 13](#_Toc120465460)

[4.3.1 Pruebas del plan de validación 13](#_Toc120465461)

[5 Conclusiones y trabajo futuro 15](#_Toc120465462)

[5.1 Trabajo futuro 15](#_Toc120465463)

[Referencias 16](#_Toc120465464)

# Introducción

<Motivación y contexto del TFM>

## Objetivos

<Descripción de los objetivos del TFM>

## Entorno de usuarios

<¿A quién va dirigido este TFM? Descripción de los posibles usuarios>

## Directivas del proyecto

A continuación, las Tablas 1.1 y 1.2 resumen la descripción tanto del problema abordado en este proyecto como de la solución propuesta.

**Tabla 1.1.** Descripción del problema

|  |  |
| --- | --- |
| El problema de |  |
| Afecta a |  |
| Lo cual tiene como impacto |  |
| Una solución satisfactoria sería |  |

**Tabla 1.2.** Descripción de la solución

|  |  |
| --- | --- |
| Para |  |
| Los cuales |  |
| ASISTE |  |
| Que |  |
| Frente a |  |
| Nuestro producto |  |

## Alternativas y competencia

<Descripción de alternativas/competencia al sistema desarrollado>

## Estructura de la memoria

<Breve descripción de cómo se organiza el contenido de la memoria>

## Requisitos y casos de uso

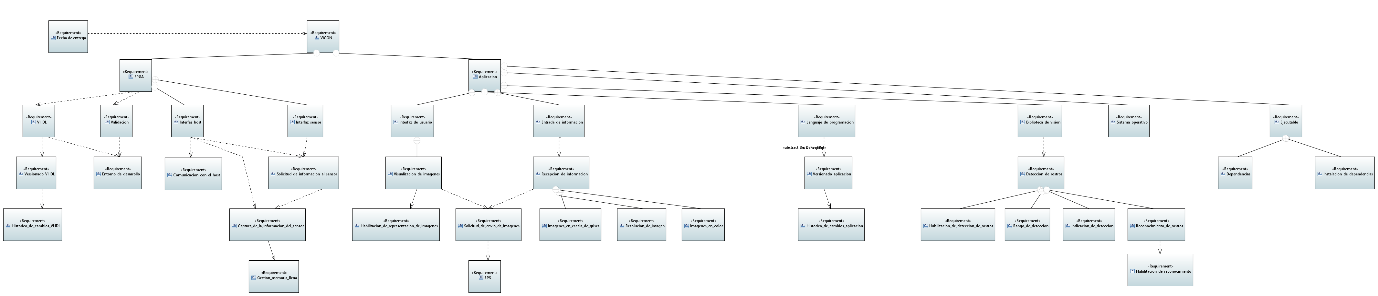
<En este Capítulo se describen los requisitos que guiarán nuestro proyecto, así como los casos de uso para los que se pretende dar solución>

## Requisitos del proyecto

<La Tabla 2.1 resume el conjunto de requisitos capturados para nuestro proyecto. En la columna de prioridad, el valor “F” hace referencia a un requisito fundamental del proyecto y “D” a requisito deseable (requisito que se intentará cumplir). La Figura 2.1 muestra las relaciones entre requisitos de forma gráfica.>

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Nombre | Prioridad | Precedencia |
| SW1 | Aplicación | F |  |
| SW2 | Interfaz de usuario | F | SW1 |
| SW3 | Lenguaje de programación | F | SW1 |
| SW4 | Entrada de información | F | SW1 |
| SW5 | Recepción de la información | F | SW4 |
| SW6 | Visualización de imágenes | F | SW2 |
| SW7 | Solicitud de envío de imágenes | F | SW4, SW5 |
| SW8 | Habilitación de representación de imágenes | F | SW6 |
| SW9 | FPS | F | SW7 |
| SW10 | Imágenes en escala de grises | F | SW5 |
| SW11 | Resolución de imagen | F | SW5 |
| SW12 | Imágenes en color | D | SW5 |
| SW13 | Biblioteca de visión | F | SW1 |
| SW14 | Detección de rostros | F | SW13 |
| SW15 | Habilitación de detección de rostros | F | SW14 |
| SW16 | Rango de detección | F | SW14 |
| SW17 | Indicación de detección | F | SW14 |
| SW18 | Reconocimiento de rostros | O | SW14 |
| SW19 | Habilitación de reconocimiento | O | SW18 |
| SW20 | Sistema Operativo | F | SW1 |
| SW21 | Ejecutable | F | SW1 |
| SW22 | Dependencias | F | SW21 |
| SW23 | Instalación de dependencias | F | SW22 |
| SW24 | Versionado aplicación | F | SW22 |
| SW25 | Histórico de cambios aplicación | F | SW24 |
| HW1 | FPGA | F |  |
| HW2 | VHDL | F | HW1 |
| HW3 | Versionado VHDL | F | HW2 |
| HW4 | Histórico de cambios VHDL | F | HW3 |
| HW5 | Validación | F | HW1 |
| HW6 | Entorno de desarrollo | F | HW3, HW4 |
| HW7 | Interfaz host | F | HW1 |
| HW8 | Interfaz sensor | F | HW1 |
| HW9 | Solicitud de información al sensor | F | HW7, HW8 |
| HW10 | Captura de la información del sensor | F | HW7, HW9 |
| HW11 | Comunicación con el host | F | HW7 |
| HW12 | Gestión de memoria llena | D | HW10 |
| PR1 | Fecha de entrega | F |  |

**Tabla 2.1**. Requisitos del proyecto



**Figura 2.1** Diagrama general de requisitos

### Descripción de los Requisitos Funcionales

* **SW4:** La aplicación recibirá como datos de entrada las imágenes de la cámara.
* **SW5:** La recepción de la información de la FPGA se realizará mediante USB.
* **SW6:** La aplicación deberá ser capaz de mostrar las imágenes recibidas del sensor en tiempo real.
* **SW7:** Las imágenes a mostrar deberán ser solicitadas a la FPGA a demanda, en función de la capacidad que la aplicación tenga para representarlas
* **SW8:** Las representación de imágenes, y por ende su recepción, deberá ser habilitable y deshabilitable desde la aplicación
* **SW9:** La visualización de imágenes deberá mostrar, al menos, 10 fotogramas por segundo cuando esté habilitada.
* **SW10:** Las imágenes recibidas y mostradas deberán ser, al menos, en escala de grises.
* **SW11:** La resolución de las imágenes será VGA (640x480 píxeles).
* **SW12:** La aplicación mostrará las imágenes recibidas en color.
* **SW13:** La aplicación hará uso de la biblioteca OpenCV para implementar los algoritmos relacionados con la detección y reconocimiento de rostros
* **SW14:** La aplicación deberá implementar un algoritmo de detección de rostros sobre las imágenes recibidas.
* **SW15:** La aplicación deberá permitir habilitar y deshabilitar la detección de rostros sobre las imágenes.
* **SW16:** El reconocimiento facial debe ser preciso y fiable en un rango de distancias de 1 a 5 metros entre la cámara (controlada por la FPGA) y los rostros de las personas.
* **SW17:** Se indicarán los rostros detectados sobre la propia imagen mostrada.
* **SW18:** La aplicación podrá determinar si los rostros detectados se corresponden con aquellos previamente almacenados en una base de datos
* **SW19:** La aplicación deberá permitir habilitar y deshabilitar el reconocimiento de rostros.
* **HW7:** El hardware debe ser capaz de transmitir datos a demanda del dispositivo host al que se conecte, tal que sea posible visualizarlos en el segundo en tiempo real.
* **HW8:** La FPGA se comunicará con una cámara mediante una implementación hardware desarrollado a medida, con el fin de gestionar dicha comunicación entre ambos dispositivos de manera eficiente.
* **HW9:** La FPGA solicitará información de imagen a la cámara, aguardando como respuesta dicha información, cuando se haya recibido una petición del host
* **HW10:** La FPGA se encargará de capturar la información recibida desde el sensor para ser transmitida al host.
* **HW11:** La FPGA se comunicará con el host mediante el uso de una implementación hardware de comunicaciones de FTDI para comunicación serie.
* **HW12:** El sistema será capaz de gestionar que la memoria se llene, tal que pueda mantenerse la detección en tiempo real en todo momento.

### Descripción de los Requisitos No Funcionales

* **SW1:** Se proporcionará una aplicación software para ser ejecutada en un dispositivo host.
* **SW2:** La aplicación software proporcionará una interfaz de usuario para interactuar con la aplicación.
* **SW3:** El apartado software estará desarrollado en uno de los siguientes lenguajes: C, C++.
* **SW20:** La aplicación será compatible, al menos, con el sistema operativo Windows 10.
* **SW21:** El ejecutable entregado será autocontenido.
* **SW22:** El ejecutable entregado debe incluir todas las dependencias necesarias para su correcta instalación.
* **SW23:** Si alguna dependencia no puede ser incluida directamente en el ejecutable, este debe ser capaz de detectar esta situación y proceder a instalar automáticamente dichas dependencias sin intervención manual del usuario.
* **SW24:** El código fuente de la aplicación desarrollado durante el proyecto debe ser versionado y almacenado en un repositorio, haciendo uso de un sistema de control de versiones adecuado.
* **SW25:** El repositorio que almacena el proyecto de la aplicación debe permitir el acceso tanto al desarrollador como al cliente, tal que sea posible seguir los cambios y recuperar versiones anteriores del código.
* **HW1:** Se hará uso de un chip FPGA de bajo coste y de una gama actual.
* **HW2:** El diseño hardware realizado sobre la FPGA se hará mediante lenguaje VHDL.
* **HW3:** El código fuente VHDL desarrollado durante el proyecto debe ser versionado y almacenado en un repositorio, haciendo uso de un sistema de control de versiones adecuado.
* **HW4:** El repositorio que almacena el proyecto debe permitir el acceso tanto al desarrollador como al cliente, tal que sea posible seguir los cambios y recuperar versiones anteriores del código.
* **HW5:** La validación del desarrollo en FPGA será realizada mediante scripts TCL, que se encargarán de mostrar que el desarrollo cumple con la funcionalidad deseada.
* **HW6:** El entorno de desarrollo del diseño hardware será Vivado Design Suite.
* **PR1:** El proyecto debe terminar el 31/05/2025.

## Casos de uso

Los actores externos con los que interactúa nuestro sistema se muestran en la Tabla 2.2. Las funcionalidades del sistema son, básicamente, capturar los datos procedentes de los sensores y notificar sus valores, así como notificar situaciones de alerta, y comprobar el estado de la conexión. El diagrama que muestra la relación del sistema (SMC) con los actores descritos en la Tabla 2.2 se muestra en la Figura 2.2.

|  |  |
| --- | --- |
| Nombre | Descripción |
|  |  |
|  |  |
|  |  |
|  |  |

**Tabla 2.2.** Actores del sistema

**Figura 2.2**. Diagrama de casos de uso

### Descripción textual de los casos de uso

<Descripción en texto de los casos de uso. Ver ejemplo en el docx ‘Especificación Historia de Antonio’>

## Diseño y desarrollo del sistema

## Arquitectura Física

<El diagrama general de la arquitectura física se muestra en la Figura 3.1. Descripción en texto justificando esta arquitectura física.>

**Figura 3.1** Arquitectura física del sistema

### …

<Descripción de cada uno de los bloques que forman la arquitectura física: ¿por qué se ha seleccionado? ¿cuál es su misión en el sistema? configuración, parámetros, etc.

### …

### Esquema hardware

<Una vez descrita la arquitectura física y presentados todos los elementos que la componen, así como nombrado su conexionado, describir el montaje hardware final. Si hay una Prueba de Concepto previa (antes de tener el sistema final), se describe aquí también. Añadir fotos del montaje final>

## Arquitectura Lógica

<Descripción de la arquitectura lógica en la Figura 3.x>

**Figura 3.x** Vista general de la arquitectura lógica

### …

<Descripción de cada uno de los bloques que forman la arquitectura lógica: diagrama de flujo, librerías utilizadas, variables, etc. Se puede usar 3.2.1 para describir brevemente el entorno de programación usado y luego 3.2.2 y siguientes para describir los bloques de la arquitectura lógica.>

### …

## Resultados

## Verificación de requisitos

##### Entorno 1

<Descripción del entorno de las pruebas. Puede no ser el definitivo>

### <Prueba de verificación de requisito/s>

|  |  |  |
| --- | --- | --- |
| **Requisito/s** | | <Requisitos que se verifican> |
| **Prueba** | <Descripción de la prueba> | |
| **Salida** | 1. <Qué se observa en la salida> 2. …. | |
| **Errores** | 1. <Posibles problemas> | |

|  |  |  |
| --- | --- | --- |
| **Requisito/s** | | <Requisitos que se verifican> |
| **Prueba** | <Descripción de la prueba> | |
| **Salida** | 1. <Qué se observa en la salida> 2. …. | |
| **Errores** | 1. <Posibles problemas> | |

##### Entorno x

<Descripción del entorno de las pruebas>

### <Prueba de verificación de requisito/s>

|  |  |  |
| --- | --- | --- |
| **Requisito/s** | | <Requisitos que se verifican> |
| **Prueba** | <Descripción de la prueba> | |
| **Salida** | 1. <Qué se observa en la salida> 2. …. | |
| **Errores** | 1. <Posibles problemas> | |

|  |  |  |
| --- | --- | --- |
| **Requisito/s** | | <Requisitos que se verifican> |
| **Prueba** | <Descripción de la prueba> | |
| **Salida** | 1. <Qué se observa en la salida> 2. …. | |
| **Errores** | 1. <Posibles problemas> | |

## Resultados de las pruebas del sistema

<Descripción de las pruebas de verificación del sistema ¿ha habido errores? Si los ha habido ¿se han solucionado? ¿cómo?>

## Plan de validación

|  |  |
| --- | --- |
| **Nombre del entregable** | **Pruebas asociadas para su validación** |
| <Entregable> | <Pruebas que validan el entregable |

### Pruebas del plan de validación

|  |  |
| --- | --- |
| **<Descripción de la prueba>** | **<Id de prueba>** |
| **Descripción:**  <Breve descripción de la prueba> | |
| **Prerrequisitos**  <Condiciones en las que se hace la prueba> | |
| **Pasos:**  <Pasos que se llevan a cabo para ejecutar la prueba> | |
| **Resultado esperado:**  **<Qué se espera obtener>** | |
| **Resultado obtenido:**  **<Descripción del resultado obtenido. Se pueden añadir fotografías que muestren el resultado>** | |

## Conclusiones y trabajo futuro

<Analizar qué se ha desarrollado, si cumple con lo esperado. Describir qué problemas han surgido. Qué falta para que el sistema sea realmente práctico.>

## Trabajo futuro

<Qué se podría hacer ahora para mejorar el sistema propuesto>

# Referencias

<Referencias>