



MÁSTER EN INGENIERÍA ELECTRÓNICA



VNIVERSITAT
DE VALÈNCIA

TRABAJO FIN DE GRADO

DESARROLLO DE UN SISTEMA DE CAPTACIÓN DE
IMÁGENES Y DATOS PARA DISPOSITIVOS IOT

AUTOR: RUBÉN GARRIDO ALONSO

TUTORES: VICENT GIRBÉS Y VALERO LAPARRA

JULIO 2023



VNIVERSITAT
DE VALÈNCIA



Escola Tècnica Superior
d'Enginyeria **ETSE-UV**

TRABAJO FIN DE GRADO

DESARROLLO DE UN SISTEMA DE CAPTACIÓN DE IMÁGENES Y DATOS PARA DISPOSITIVOS IOT

AUTOR: RUBÉN GARRIDO ALONSO

TUTORES: VICENT GIRBÉS Y VALERO LAPARRA

Declaración de autoría:

Yo, Rubén Garrido Alonso, declaro la autoría del Trabajo Fin de Grado titulado “Desarrollo de un sistema de captación de imágenes y datos para dispositivos IoT” y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual. El material no original que figura en este trabajo ha sido atribuido a sus legítimos autores.

Valencia, May 23, 2023

Fdo: Rubén Garrido Alonso

Resumen:

Este es el resumen del TFM. Debe ser corto (máximo media página) y cubrir los aspectos principales del TFM.

Abstract:

This is the abstract of the TFM. It must be short and cover the main aspects of the TFM.

Resum:

Aquest és el resum del TFM. Ha de ser curt (màxim mitja pàgina) i cobrir els aspectes principals del TFM.

Agradecimientos:

En primer lugar quiero agradecer a todos aquellos que me han apoyado durante todos estos años.

En segundo lugar...

Contents

1	Introducción	17
1.1	Introducción	17
1.2	Objetivos	17
2	Estado del arte	19
2.1	Tecnologías utilizadas	19
2.1.1	Entorno Linux	19
2.2	Estudio previo	20
2.2.1	Microcontrolador ESP32	20
2.2.2	Servidor HTTP Python	21
2.2.3	Módulos y librerías adicionales	22
3	Arquitectura del microcontrolador	23
3.1	Descripción arquitectura ESP32	23
3.2	Protocolos utilizados	23
3.3	Implementación de la captura de las fotos	23
3.4	Funcionamiento como cliente HTTP	23
3.5	Funcionamiento como servidor HTTP	23
3.6	Implementación de la captura de las fotos	23
4	Servidor HTTP	25
5	Resultados y discusión	27
6	Conclusiones	29
6.1	Revisión de costes	29
6.2	Conclusiones	29
6.3	Trabajo futuro	29
A	Apéndice	31
A.1	Ejemplos del lenguaje de marcado Latex	31

Bibliografía

32

Chapter 1

Introducción

1.1 Introducción

Cada vez es mayor el peso que la tecnología esta teniendo en el día a día de la sociedad. La interacción con dispositivos tecnológicos está convirtiéndose poco a poco en algo cotidiano. Son cada vez mas las tareas que pueden ser suplidas por dispositivos electrónicos que ayudan a que tareas del día a día, incluso labores más peligrosas puedan desarrollarse de una forma segura y eficiente.

Uno de los sectores en auge en esta última década es el sector de los sistemas embarcados y la capacidad de procesamiento de estos mismos, es cada vez mayor la capacidad de dotar de inteligencia a dispositivos cada vez con dimensiones mas reducidas pero con una capacidad de rendimiento que crece de forma exponencial.

El proyecto que se describirá a lo largo de la memoria se encarga de dotar a determinados dispositivos de la capacidad de comunicación con un servidor central, utilizado para procesar una determinada informacion que cada uno de los dispositivos decida enviar, generalmente, sistemas de visión. Para poder conseguir esto, se ha tenido que poner a punto microcontrolador capaz de poder capturar imágenes a través de una cámara. Posteriormente, en un servidor central, dicha información será atendida y posteriormente procesada para extraer la información necesaria.

1.2 Objetivos

El objetivo principal de este proyecto es establecer una comunicación bidireccional efectiva entre el microcontrolador ESP32 y el servidor remoto, permitiendo la captura y transferencia de imágenes de manera confiable. Además, se busca demostrar la viabilidad de utilizar el ESP32 como una solución integral para aplicaciones de captura de imágenes y comunicación en tiempo real.

A lo largo de la implementación del proyecto, se tendrán en cuenta aspectos como la eficiencia energética, la estabilidad y la escalabilidad del sistema. También se explorarán posibles mejoras y consideraciones adicionales, como el procesamiento de imágenes en el servidor y la integración con técnicas de inteligencia artificial para análisis de imágenes.

En resumen, este proyecto tiene como objetivo desarrollar un sistema basado en el microcontrolador ESP32 para la captura y transferencia de imágenes a través de HTTP.

Se espera que este sistema pueda aplicarse en diferentes escenarios, como vigilancia, automatización del hogar, control de accesos, entre otros, brindando una solución confiable y versátil para la captura y transmisión de imágenes en tiempo real.

Es importante destacar que los objetivos y alcances del proyecto pueden ajustarse durante su desarrollo en función de las necesidades y limitaciones identificadas a lo largo del proceso.

Chapter 2

Estado del arte

2.1 Tecnologías utilizadas

Cada vez es mayor el peso que la tecnología esta teniendo en el día a día de la sociedad. La interacción con dispositivos tecnológicos está convirtiéndose poco a poco en algo cotidiano. Son cada vez mas las tareas que pueden ser suplidas por dispositivos electrónicos que ayudan a que tareas del día a día, incluso labores más peligrosas puedan desarrollarse de una forma segura y eficiente.

Uno de los sectores en auge en esta última década es el sector de los sistemas embebidos y la capacidad de procesamiento de estos mismos, es cada vez mayor la capacidad de dotar de inteligencia a dispositivos cada vez con dimensiones mas reducidas pero con una capacidad de rendimiento que crece de forma exponencial.

El proyecto que se describirá a lo largo de la memoria se encarga de dotar de inteligencia artificial un robot con capacidad de teleoperación y reconocimiento del entorno en el cual se esté operando este mismo. Para poder conseguir esto, se ha tenido que poner a punto un robot que sea capaz de moverse y además sea capaz de procesar y reconocer los objetos de su entorno procesando esta información en tiempo real.

Para conseguir esto se ha dotado al robot de visión artificial, la cual recogerá la información necesaria de una cámara incorporada en el robot y que se complementará con otros sensores que ayudaran al dispositivo móvil a poder moverse de una forma autónoma evitando colisiones.

2.1.1 Entorno Linux

Existen diferentes recursos que se pueden utilizar a la hora de poder levantar un servidor HTTP, así como diferente hardware en el que poder hacerlo. En el proyecto que se describe, la intención es dotar al proyecto de la mayor flexibilidad posible, la plataforma elegida ha sido un entorno Linux.

Linux es conocido como un sistema operativo basado en código abierto y gratuito, dicho sistema operativo se sostiene sobre un kernel, donde reside la verdadera funcionalidad de Linux, por tanto, una mejor descripción sería decir que, Linux, es un kernel sobre el cual se basa un sistema operativo. Fue creado por Linus Torvalds en 1991 como un proyecto personal y se ha convertido en uno de los sistemas más populares y ampliamente utilizados en el mundo.

Gran parte de la popularidad y del auge en el uso de Linux, es su naturaleza de código abierto, más comunmente conocido como "open source", lo que significa que el código fuente del kernel está disponible para que cualquier persona lo pueda ver, modificar y distribuir. Esto fomenta que la comunidad de Linux sea completamente abierta, por lo que favorece la mejora de su funcionamiento así como al añadido continuo de nuevas funcionalidades, las cuales poco a poco abarcan más campos de desarrollo.

Otra de las razones por las cuales Linux es tan utilizado en la industria, reside en, la capacidad de adaptación del kernel de Linux a cualquier arquitectura hardware del dispositivo que se quiera poner en marcha para un proyecto. De hecho, es muy común utilizar hardware ligeramente diferente en las etapas de prototipado y en la puesta en marcha en producción. Por lo cual, se puede utilizar una misma versión de kernel para diferentes arquitecturas y el código que se escriba no difiere de un hardware a otro, o si lo hace, son cambios superficiales. Esto ocasiona que, podamos utilizar la misma versión del Kernel de Linux tanto en el PC en el que estemos desarrollando, así como en el hardware el cual pondremos en funcionamiento.

2.2 Estudio previo

El proyecto se apoya en el microcontrolador ESP32, el protocolo HTTP/HTTPS, el servidor en Python con Flask y una variedad de bibliotecas adicionales para ofrecer una solución completa y eficiente. Estas tecnologías permiten la captura de imágenes, la comunicación segura con el servidor y el procesamiento de los datos recibidos. La elección de estas tecnologías se basa en su adecuación al propósito del proyecto, su capacidad de cumplir con los requisitos de rendimiento y seguridad, así como su popularidad y soporte en la comunidad de desarrollo.

2.2.1 Microcontrolador ESP32

El microcontrolador ESP32, desarrollado por Espressif Systems, es el componente central de este proyecto. Este dispositivo se destaca por su alto rendimiento, bajo consumo de energía y amplias capacidades de conectividad. Cuenta con un procesador de doble núcleo, conectividad Wi-Fi y Bluetooth, así como interfaces GPIO para interactuar con diversos periféricos. El ESP32 ofrece una plataforma sólida y versátil para implementar aplicaciones IoT y sistemas embebidos.

Espressif Systems tiene una amplia gama de microcontroladores, dentro de los microcontroladores disponibles, hay diferentes criterios a tener en cuenta que pueden influir en la elección. Aquí se muestra una pequeña descripción de estos criterios a tener en cuenta.

Potencia de procesamiento

Dentro de la gama de microcontroladores de Espressif, concretamente ESP32, existen diferentes velocidades de reloj y configuraciones de núcleos. Es relativamente complejo evaluar la potencia que va a consumir nuestro programa antes siquiera de escribir el código que va a ejecutar dicha aplicación. Aun así, existen diferentes recursos en la red donde podemos consultar proyectos similares, para tener en cuenta el hardware que se ha utilizado en dicho proyecto.

Memoria

Otro recurso que se debe de tener en cuenta cuando se esta evaluando la tecnología a utilizar, es la capacidad de almacenamiento del dispositivo asi como el tamaño de memoria RAM que se dispone en el dispositivo. Es importante plantear que se debe de almacenar en el microcontrolador, por ejemplo en el caso que se describe, se debe de almacenar el binario generado a partir de código escrito además de credenciales Wifi, imagenes capturadas de la cámara de forma temporal, certificados de servidores, etc...

A parte de lo descrito en el párrafo anterior, el proyectistew se ha de asegurar que haya suficiente memoria RAM para las operaciones y cálculos requeridos en el proyecto.

Periféricos y conectividad

No se ha de olvidar el nivel de conectividad que se le quiera dar al microcontrolador utilizado en el proyecto. En el caso de este proyecto en particular, se deben de considerar diferentes periféricos o diferentes canales de conectividad, a continuación se describen los canales periféricos que se han tenido en cuenta.

WiFi Ya que el dispositivo debe de funcionar de forma remota, una de las principales funcionalidades a nivel de conectividad debe de ser la conectividad WiFi, ya que va a ser un dispositivo que va a funcionar el 100% del tiempo de forma remota, se necesita no perder la comunicación con el dispositivo en ningún momento, además de que se debe de poder establecer un canal de comunicación bidireccional.

Bluetooth Bluetooth (incluyendo Bluetooth Low Energy),

Sensórica Tal y como se describe en el proyecto, el sensor principal alrededor del cual gira el proyecto es la cámara. Por tanto, se ha de buscar un microcontrolador de la gama que cuente con conectividad SSBC, que son las siglas de la interfaz de bus de cámara serial. Además de esta interfaz, se ha comprobado, que el microcontrolador escogido, cuente con interfaces comunes para poder dotar de escalabilidad al proyecto, tales como UART, I2C, SPI, ADC's, GPIO... Se esta forma, se garantiza que en un futuro el proyecto se pueda escalar y se le puedan añadir los sensores adpatados a nuevas necesidades del proyecto.

Consumo de energía La eficiencia energética es otro de los factores a tener en cuenta a la hora de escoger un dispositivo, en este caso en concreto, es cierto que Si la eficiencia energética es un factor importante en tu proyecto, verifica las especificaciones de consumo de energía del microcontrolador ESP32. Algunos microcontroladores ofrecen modos de bajo consumo que pueden ser útiles en aplicaciones con restricciones energéticas.

2.2.2 Servidor HTTP Python

Para la comunicación entre el microcontrolador y el servidor, se ha utilizado el protocolo HTTP (Hypertext Transfer Protocol) y su versión segura HTTPS (HTTP Secure). Estos protocolos permiten la transferencia de datos entre el cliente (microcontrolador) y el

servidor de forma segura y eficiente. El uso de HTTPS garantiza la encriptación de los datos transmitidos, protegiendo la confidencialidad y la integridad de la información.

Table 2.1: Comparación de servidor en Flask con otros lenguajes

Lenguaje	Complejidad de programación	Consumo de recursos
Python (Flask)	Baja	Moderado
Python (MongoDB)	Baja a moderada	Moderado
Node.js	Baja a moderada	Moderado a alto
Java (Spring)	Moderada	Alto
Ruby (Ruby on Rails)	Moderada	Moderado a alto
C/C++ (Laravel)	Moderada a alta	Bajo

2.2.3 Módulos y librerías adicionales

Además de las tecnologías mencionadas, se han utilizado diversas bibliotecas y módulos adicionales para la implementación y funcionalidad del sistema. Estas incluyen bibliotecas para el manejo de imágenes, como PIL (Python Imaging Library), para el procesamiento y análisis de datos, como NumPy y OpenCV, y para la encriptación y autenticación en el lado del servidor, como OpenSSL.

Chapter 3

Arquitectura del microcontrolador

3.1 Descripción arquitectura ESP32

3.2 Protocolos utilizados

3.3 Implementación de la captura de las fotos

3.4 Funcionamiento como cliente HTTP

3.5 Funcionamiento como servidor HTTP

3.6 Implementación de la captura de las fotos

Chapter 4

Servidor HTTP

Chapter 5

Resultados y discusión

Chapter 6

Conclusiones

6.1 Revisión de costes

6.2 Conclusiones

6.3 Trabajo futuro

Appendix A

Apéndice

A.1 Ejemplos del lenguaje de marcado Latex

This document is an example of BibTeX using in bibliography management. Three items are cited: *The L^AT_EX Companion* book [?], the Einstein journal paper [?], and the Donald Knuth's website [?]. The L^AT_EX related items are [?, ?]¹.

Texto en el párrafo 1.

Texto en el párrafo 2.

Texto en el párrafo 3.

- Consideración 1
- Consideración 2

1. Punto 1

2. Punto 2

A continuación se muestra una ecuación:

$$\int_0^1 \frac{1}{x^2 + 1} dx$$

Podemos incluir imágenes en formato: png, pdf o jpg.

En la figura A.1 se muestra un diagrama realizado con <https://www.yworks.com/products/yed>:

Imagen 1

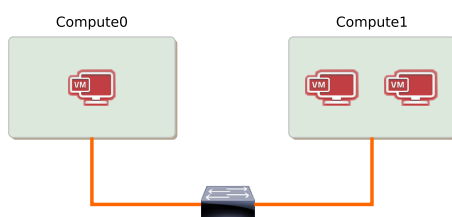
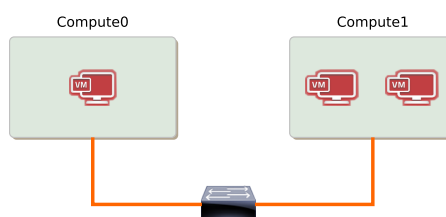


Imagen 2



¹Esto está tomado de https://www.overleaf.com/learn/latex/Bibliography_management_with_bibtex

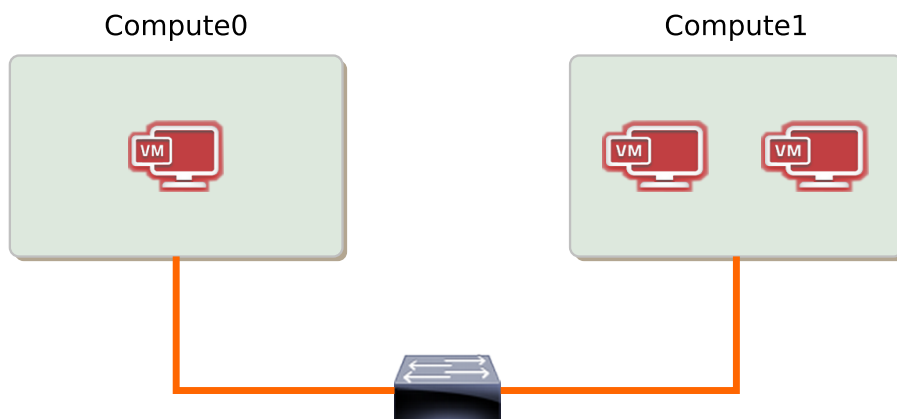


Figure A.1: Esta es una figura que latex decide donde colocar (floating) en el documento.

Este es un ejemplo de una tabla:

Columna 1	Columna 2
1	2

O la misma tabla centrada:

Columna 1	Columna 2
1	2

Para generar el fichero PDF:

```
pdflatex ejemplo-memoria.tex
bibtex ejemplo-memoria
pdflatex ejemplo-memoria.tex
```

También se puede usar `latexmk` que automáticamente regenera la bibliografía.

```
latexmk -pdf ejemplo-memoria.tex
```