Universidad Autónoma de Madrid
Escuela politécnica superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

1718_072_IC DISEÑO E IMPLEMENTACIÓN DE UN HUB DE CONTROL DOMÓTICO

Autor: Pallarés Jiménez, Ignacio
Tutor: Delgado Mohatar, Óscar
Ponente: Anguiano Rey, Eloy

JULIO 2018

1718_072_IC DISEÑO E IMPLEMENTACIÓN DE UN HUB DE CONTROL DOMÓTICO

Autor: Pallarés Jiménez, Ignacio
Tutor: Delgado Mohatar, Óscar
Ponente: Anguiano Rey, Eloy

Grupo de la EPS (opcional)
Dpto. de XXXXX
Escuela Politécnica Superior
Universidad Autónoma de Madrid
JULIO 2018

Resumen

Resumen

La domótica consiste en la automatización del hogar. Los sistemas domóticos, son aquellos capaces de domotizar una vivienda, y proporcionan servicios de comunicación, seguridad, eficiencia energética...etc. Sin duda alguna, la comunicación entre estos sistemas es algo esencial, existiendo redes cableadas e inalámbricas para ello, pudiendo ser controlados estos sistemas desde dentro y fuera del hogar.

BRIMO es un proyecto de código abierto para la gestión y el control de dispositivos domóticos en el hogar. Es una alternativa open source de bajo coste para todas aquellas personas que deseen domotizar su hogar de una manera barata y sencilla. No utiliza protocolos privados, y cualquiera que lo desee puede utilizar y modificar la aplicación a su gusto.

Además, cualquier persona puede crear sus dispositivos (sensores, actuadores o cámaras) de manera sencilla, siempre que éstos sigan los requisitos establecidos. Brimo nos ayuda, gracias a una interfaz web sencilla e intuitiva, a ordenar nuestros dispositivos, visualizar sus estados y mandar comandos a los dispositivos que los acepten.

La aplicación está pensada para ejecutarse en entornos ligeros, concretamente en una Raspberry Pi 3 (precio asqequible), pero también puede ser ejecutada en cualquier ordenador tras una simple configuración. Utiliza una arquitectura REST sobre el protocolo HTTP para comunicarse con los dispositivos, y una arquitectura MVC (Model View Controller) para la interacción con el usuario.

La función principal de Brimo es de "bridge", punto común entre el usuario y los dispositivos, se encarga de poner en contacto al usuario con los dispositivos.

Palabras Clave

Domótica, código abierto, REST, raspberry, HTTP, sensores, MVC, actuadores, bridge.

Abstract

Domotic consists of home automation. Domotic systems are those capable of automating a home, and provide communication services, security, energy efficiency ... etc. Communication between these systems is essential, existing wired and wireless networks for it, that allows us to controll them from inside and outside the home.

BRIMO is an open-source project for the management and control of domotic devices in the home.It is a low-cost open source alternative for all people who want to domotize their home in a cheap and simple way. It does not use private protocols, and anyone can use or modify the application on its preferences.

Besides, anyone is able to create its own devices (sensors, actuators or cameras) in a simple way following the application requirements. Brimo helps us, thanks to a simple and intuitive web interface, to arrange our devices, seeing their status and to send them commands.

The application is designed to be runned on lightweight devices, specifically into a Raspberry Pi 3 (low cost), but it could be also runned on any computer after a simple configuration. It uses REST architecture over HTTP protocol to communicate with devices and a MVC (Model View Controller) architecture for the interaction with users.

The main function of Brimo is to act as bridge, the common point between users and devices: Brimo is the responsible of the communication between them.

Key words

Domotic, open-source, low cost, REST, raspberry, HTTP, sensors, MVC, actuators.

Agradecimientos

Índice general

Índice de Figuras x

Índice de Tablas xii

1. Introducción 1

1.1. Motivación del proyecto . . . 1
1.2. Objetivos y enfoque . . . 2
1.3. Metodología y plan de trabajo . . . 2

2. Reconocimiento de iris. Estado del arte 3

2.1. Introducción . . . 3
2.2. Historia, nacimiento y evolucin. . . . 3
2.3. La anatoma del ojo . . . 3
2.3.1. Aspectos diferenciadores del iris . . . 3
2.4. Adquisicin del Iris . . . 3
2.4.1. Introduccin . . . 3
2.4.2. Esquemas de adquisicin tradicionales . . . 3
2.4.3. Consideraciones sobre la iluminacin . . . 3
2.4.4. Posicionamiento del Iris . . . 3
2.4.5. Sistemas comerciales de adquisicin . . . 3
2.5. Localizacin y segmentacin del Iris . . . 3
2.5.1. Introduccin . . . 4
2.5.2. Metodologa de J. Daugman y derivadas . . . 4
2.5.3. Metodologa de R. Wildes y derivadas . . . 4
2.5.4. Otras metodologas . . . 4
2.5.5. Comparativa de metodologas . . . 4
2.5.6. Deteccin de pestaas y ruido . . . 4
2.6. Normalizacin del tamao . . . 4
2.6.1. Daugman's Rubber Sheet Model . . . 4
2.6.2. Image Registration . . . 4
2.6.3. Normalizacin en ngulo . . . 4

2.6.4. Mejora del contraste y eliminacin de ruido 4
2.7. Algoritmos de Codificacin 4
2.7.1. Metodologa de Daugman: Filtros de Gabor 4
2.7.2. Metodologas alternativas a la de Daugman 4
2.7.3. Metodologas de Wildes. Vectores de caractersticas reales (no binarios) . . 4
2.8. Algoritmos de Matching 4
2.8.1. Introduccin 4
2.8.2. Distancia de Hamming 4
2.8.3. Distancia eucldea ponderada 4
2.8.4. Correlacin normalizada 4
2.9. Problemtica y retos futuros 4
2.9.1. Segmentacin 5
2.9.2. Captura ideal no invasiva 5
2.10. Competiciones o Evaluaciones de Iris 5
2.10.1. The Iris Challenge Evaluation (ICE) 5
2.10.2. The Noisy Iris Challenge Evaluation (NICE) 5
2.11. Bases de datos 5
2.11.1. CASIA 5
2.11.2. BioSec Baseline y BioSecurID 5

3. Sistema, diseño y desarrollo **7**
3.1. Segmentacin 7
3.2. Normalizacin 7
3.3. Codificacin 7
3.4. Matching 7

4. Protocolos **9**
4.1. Comunicación con los dispositivos 9
4.1.1. Registro de nuevos dispositivos 9

5. Experimentos Realizados y Resultados **11**
5.1. Bases de datos y protocolo 11
5.2. Sistemas de referencia 11
5.3. Escenarios de pruebas 11
5.4. Experimentos del sistema completo 11

6. Conclusiones y trabajo futuro **13**

Glosario de acrónimos **15**

Bibliografa **16**

A. Manual de utilización **19**

B. Manual del programador **21**

Índice de Figuras

1.1. Ejemplo pie de figura 2 2

Índice de Tablas

1 Introducción

1.1. Motivación del proyecto

Los sistemas domóticos por lo general utilizan una arquitectura centralizada: un controlador (bridge) es el encargado de enviar y recibir información de los dispositivos domóticos y las interfaces. Se utilizan sistemas centralizados debido a que abaratan mucho el coste de los dispositivos domóticos, así los dispositivos tienen poca electrónica y programación, y la responsabilidad principal reside en el bridge. Este enfoque tiene sentido cuando se trata de muchos dispositivos en un hogar, que es el caso ideal, si solo tuviésemos un sensor carecería de sentido tener un sensor y un bridge para manejarlo.

El problema principal que existe con los sistemas centralizados se encuentra en la **compatibilidad** entre dispositivos y bridges. Por lo que he observado [1], todavía falta mucha estandarización en el ámbito de la domótica: cada fabricante usa sus medios y protocolos haciendo incompatibles bridges y dispositivos. Además, estos dispositivos no suelen ser muy asequibles. Por lo tanto, nos encontramos ante la necesidad de comprar todos los dispositivos de una misma marca o tener muchos bridges, lo que nos obligaría a manejar cada dispositivo desde su correspondiente bridge.

La domótica puede hacernos la vida en el hogar mucho más sencilla, ayudándonos a ahorrar tiempo y dinero que podremos invertir en otras cosas. Los hogares todavía están muy poco automatizados, y mi principal motivación ha sido acercar la domótica a las personas y aprender acerca de ella. Gracias a nuestro sistema manejamos todos los dispositvos a través de un solo bridge de manera sencilla y eficaz.

1.2. Objetivos y enfoque

El objetivo último de nuestro proyecto es desarrollar un sistema que sea capaz de recibir y enviar información de dispositivos domóticos y sea capaz de interactuar con el cliente.

Los **requisitos** que debe cumplir nuestro sistema son:

- Ligero. Un bridge no debería necesitar demasiada capacidad de procesamiento y de memoria, y es necesario que no sea muy costoso, por lo tanto, la ligereza es requisito indispensable.
- Compatibilidad. Necesitamos que nuestro bridge no sea únicamente compatible con un tipo de sensor, o un modelo de cámara
- Interfaz sencilla y adaptable a cualquier dispositivo. Necesitamos que la interfaz de nuestro bridge sea compatible con cualquier dispositivo sin perder funcinalidad.
- Seguridad. La seguridad en la domótica es algo indispensable, confío en que el día de mañana incluso las cerraduras de nuestras casas serán automáticas, y no podemos dejar la responsabilidad de la seguridad de nuestra a casa a un sistema con vulnerabilidades de seguridad.
- Escalable. Nuestro sistema ha de ser escalable y debemos pensar en todo momento en ampliaciones y trabajos futuros. La domótica evoluciona a pasos agigantados y podríamos añadir funcionalidades a nuestro sistema practicamente a diario. No obstante, es necesario acotar firmemente los límites de nuestro proyecto para ceñirnos a las horas que corresponden a un TFG, aunque debemos tener muy en cuenta en todo momento trabajos futuros y ampliaciones. Además, debemos tener en cuenta la escalabilidad: domótica en un hospital, en una ciudad...etc.

1.3. Metodología y plan de trabajo

Otro ejemplo de imagen:

Figura 1.1: Ejemplo pie de figura 2

2

Reconocimiento de iris. Estado del arte

2.1. Introducción

2.2. Historia, nacimiento y evolucin.

2.3. La anatoma del ojo

2.3.1. Aspectos diferenciadores del iris

2.4. Adquisicin del Iris

2.4.1. Introduccin

2.4.2. Esquemas de adquisicin tradicionales

2.4.3. Consideraciones sobre la iluminacin

2.4.4. Posicionamiento del Iris

2.4.5. Sistemas comerciales de adquisicin

2.5. Localizacin y segmentacin del Iris

2.5.1. Introduccin

2.5.2. Metodologa de J. Daugman y derivadas

2.5.3. Metodologa de R. Wildes y derivadas

2.5.4. Otras metodologas

2.5.5. Comparativa de metodologas

2.5.6. Deteccin de pestaas y ruido

2.6. Normalizacin del tamao

2.6.1. Daugman's Rubber Sheet Model

2.6.2. Image Registration

2.6.3. Normalizacin en ngulo

2.6.4. Mejora del contraste y eliminacin de ruido

2.7. Algoritmos de Codificacin

2.7.1. Metodologa de Daugman: Filtros de Gabor

2.7.2. Metodologas alternativas a la de Daugman

Filtros Log-Gabor

Wavelets

Haar Wavelet

Transformada Discreta del Coseno (DCT)

2.7.3. Metodologas de Wildes. Vectores de caractersticas reales (no binarios)

2.8. Algoritmos de Matching

2.8.1. Introduccin

2.8.2. Distancia de Hamming

2.8.3. Distancia eucldea ponderada

2.8.4. Correlacin normalizada

2.9. Problemtica y retos futuros

2.9.1. Segmentacin

2.9.2. Captura ideal no invasiva

2.10. Competiciones o Evaluaciones de Iris

2.10.1. The Iris Challenge Evaluation (ICE)

2.10.2. The Noisy Iris Challenge Evaluation (NICE)

2.11. Bases de datos

2.11.1. CASIA

2.11.2. BioSec Baseline y BioSecurID

3

Sistema, diseño y desarrollo

3.1. Segmentacin

3.2. Normalizacin

3.3. Codificacin

3.4. Matching

4

Protocolos

Este capítulo esta dedicado a describir los protocolos utilizados en este proyecto, apoyados sobre HTTP e implementados con un servidor CherryPy. El back-end de nuestro bridge va a diferenciar entre dos tipos de comunicación: comunicación directa con los dispositivos y comunicación con el front-end de la interfaz.

4.1. Comunicación con los dispositivos

En esta sección se describirá la comunicación con los dispositivos: registro de dispositivos y actualización de la información. El sentido de la comunicación es dispositivo -> bridge, y todos los tipos de dispositivos siguen este protocolo.

MÉTODO	ENDPOINT	DESCRIPCION	SEGURIDAD
POST	/devices/register	Registra un nuevo dispositivo.	NO
PUT	/device/:id	Edita la información de un dispositivo.	NO

4.1.1. Registro de nuevos dispositivos

La primera vez que un dispositivo se conecta al bridge es necesario que realice una primera llamada de registro. De esta manera, el bridge dará de alta el dispositivo y sabrá de qué tipo de dispositivo se trata.

5

Experimentos Realizados y Resultados

5.1. Bases de datos y protocolo

5.2. Sistemas de referencia

5.3. Escenarios de pruebas

5.4. Experimentos del sistema completo

6

Conclusiones y trabajo futuro

Glosario de acrónimos

- **Sistema domótico**: Un sistema domótico es el conjunto de controladores y dispositvios que hacen posible la automatización del hogar.
- **Dispositivo domótico**: Dispositivo que nos ayuda a la automatización del hogar actuando o recopilando información. Ejemplos: sensores de temperatura, relés, cámaras...etc.
- **Bridge**: Dispositivo que nos ayuda a controlar y administar diferentes dispositivos domóticos. Los dispositivos se conectan al bridge, y el cliente interactua directamente a través de él.
- **REST**: Arquitectura software que se apoya en el protocolo HTTP. Se utiliza en arquitecturas cliente-servidor. El cliente tiene operaciones básicas y predefinidas: GET, POST, PUT, DELETE... Y el sevidor responde a las peticiones con su correspondiente código HTTP. Cada recurso del servidor es direccionable a través de su URI.
- **Angular5**: Framework de código abierto mantenido por Google para la creación y mantenimiento de Single Page Applications (SPA). Desarrollado en TypeScript.
- **SPA**: Del inglés Single Page Application: aplicación web que se ejecuta en una sola página, sin necesidad de refrescar el navegador, haciendo más fluida la navegación.
- **MVC**: Modelo Vista Controlador: arquitectura software que separa los datos (Modelo) de la interfaz de usuario (Vista), su comunicación y lógica se encuentra en el controlador.
- **Responsive**: Diseño web cuyo objetivo es adaptar la apariencia de la página web a diferentes dispositivos.

Bibliografía

[1] Autor Apellidos. Titulo del artículo. *Revista de publicación*, pages 65–73, 2008.

A

Manual de utilización

B

Manual del programador