Implementando un Fichador Personalizado para Acciona Facility Services
Implementing a Custom Clocking Device for Acciona Facility Services

Trabajo de Fin de Grado
Curso 2023–2024

Autor
Santiago Elias Rabbia

Director
Gonzalo Méndez Pozo

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Implementando un Fichador Personalizado para Acciona Facility Services
Implementing a Custom Clocking Device for Acciona Facility Services

Trabajo de Fin de Grado en Ingeniería Informática

Autor

Santiago Elias Rabbia

Director

Gonzalo Méndez Pozo

Convocatoria: Junio 2024

Grado en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

Fecha publicación

Resumen

Implementando un Fichador Personalizado para Acciona Facility Services

Cambios en las regulaciones modificaron el procedimiento por el cual los empleados fichan al entrar a sus trabajos en Acciona Facility Services, volviendo obsoletos muchos dispositivos existentes. Se necesitaba una respuesta rápida, por lo que se desarrolló un dispositivo de fichaje rentable que utiliza microcontroladores, un PCB personalizado y una caja impresa en 3D.

Este dispositivo IoT debe ser responsivo, intuitivo de usar y ofrecer un servicio casi ininterrumpido, enviando información a través de WiFi o datos celulares. El intercambio de datos se realiza por medio de API alojada en la nube, desarrollada usando Java Spring, que pueden ser montada como un contenedor de Docker, o como una Google Cloud Function.

Palabras clave

fichador, microcontrolador, pcb, impresión 3D, cloud API, IoT, programación asíncrona, Docker, GCP, Java Spring

Abstract

Implementing a Custom Clocking Device for Acciona Facility Services

Regulatory changes have prompted a shift in employee clock-in procedures at Acciona Facility Services, rendering many existing devices obsolete. A swift response was necessary, leading to the development of a cost-effective clocking device utilizing microcontrollers, a custom PCB, and a 3D-printed case.

This IoT device has to be responsive, intuitive to use and offer close-to uninterruped service, sending information through WiFi or cellular data. Data exchange is facilitated through cloud-hosted APIs developed using Java Spring, hostable as Docker containers or as Google Cloud Functions.

Keywords

clocking device, microcontroller, pcb, 3D print, cloud API, IoT, asynchronous programming, Docker, GCP, Java Spring

Contents

List of figures

Chapter 1

Introduction and objectives

Recent legislative changes introduced by both the European Union and the Spanish government have mandated alterations to employee clocking procedures. Notably, Spain now requires employees to clock in upon arrival at work, necessitating the implementation of reliable time-tracking systems. Additionally, the European Union has banned the use of clocking devices employing biometric identification methods such as fingerprint scanning or facial recognition.

Acciona Facility Services manages many thousands of employees throughout Spain, and it was of utmost importance to expeditely replace the non-compliant devices for different ones, as many companies were already being imposed hefty fines, from tens to hundreds of thousands of Euros, for not meeting the new regulatory requirements.

The company already had providers for many types of clocking devices, and many of them were compliant, but were priced in the hundreds of Euros each. Now, faced with having to replace them, the prospect of replacing *thousands* of units at considerable expense loomed large. Moreover, outsourcing these devices often meant committing to complex time-tracking ecosystems, adding further complications and increasing the complexity of the data the company manages.

These issues were brought to light to the innovation team at Acciona Facility Services, and were then tasked with bringing a solution to market.

1.1. Objectives

Given the preceding challenges and considerations, the development of a cost-effective device and the establishment of a cloud infrastructure were deemed necessary.

The objectives of this project are as follows:

- **Design and Prototyping**: Design a device utilizing microcontrollers and additional electronic modules, such as LCD displays and NFC readers, to fulfill the specified requirements.

- **PCB Design**: Develop a Printed Circuit Board (PCB) layout to facilitate easy interconnection of the various electronic modules used in the device, ensuring efficient and reliable performance.
- **API Development**: Create an Application Programming Interface (API) capable of deployment as a Docker container or Google Cloud Function, enabling seamless communication between the device and cloud-based services.
- **Microcontroller Research**: Investigate the constraints and capabilities of microcontrollers, particularly RP2040-based microcontrollers, to inform design decisions and optimize performance.
- **3D Printing and Modeling**: Explore the fundamentals of 3D printing and 3D modeling necessary for designing a suitable enclosure for the device. Additionally, provide an overview and comparison of various materials suitable for enclosure fabrication, considering factors such as durability, cost, and aesthetic appeal.

Chapter 2

Breadboard Prototype

The first phase of developing the new clocking device involved thorough research into available components and microcontrollers in the market. Factors such as cost, functionality, and potential drawbacks were carefully evaluated to inform the selection process. Subsequently, selected components were tested by constructing a basic prototype on a breadboard to assess functionality and performance. Demonstrating the viability of the project at this stage was crucial for ensuring its continued development and success.

2.1. Hardware selection

Firstly, considering that the development was urgent, it would only be possible to use an already made controller. There were two routes to take:

- Using a **single-board computer**.

- Using a **microcontroller**.

The team's familiarity with *Raspberry Pi* products and their reputation for reliability led to the decision to utilize their offerings for the project. Given the lightweight processing requirements, the *Raspberry Pi Zero W*, a cost-effective single-board computer with built-in WiFi capabilities, emerged as a suitable option. Alternatively, the *Raspberry Pi Pico W*, a microcontroller, presented another viable choice.

However, it's important to note that while the *Pi Zero* offers more features and functionality, it comes at a higher cost compared to the *Pi Pico*. In fact, it costs almost three times as much. Therefore, careful consideration is warranted to determine whether the additional expense justifies the benefits.

2.1.1. Single-board Computers

A single-board computer (*SBC*) is a complete computer built on a single circuit board. It integrates all the necessary components required for a functional computer system, including a central processing unit (CPU), memory (RAM), storage (usually in the form of a MicroSD card), input/output ports, and sometimes additional features such as networking capabilities (e.g., Ethernet or WiFi), audio/video output, GPIO (General Purpose Input/Output) pins for connecting external devices, and even USB ports.

SBCs are designed to be compact and efficient, which is the case of the *Pi Zero*, measuring about 65mm by 30mm while drawing about one Watt of power. Additionally, SBCs can run an operating system, such as Ubuntu.

This ability to run an entire operating system significantly enhances their versatility compared to microcontrollers out-of-the-box. Many peripheral devices can simply be plugged into a USB port and function seamlessly without requiring additional configuration. For instance, a 3G SIM card adapter, which will be necessary for future stages of the project, can be effortlessly integrated into the system, letting the operating system take charge of the communication with the mobile network, abstracting all these problems from the programmer.

2.1.2. Microcontrollers

A microcontroller is a compact integrated circuit (IC) that contains a central processing unit (CPU), memory (both volatile RAM and non-volatile flash memory), input/output peripherals (such as digital and analog I/O pins), and various other hardware components necessary for interfacing with external devices. Unlike single-board computers, microcontrollers are typically designed for specific tasks and embedded applications, often with real-time requirements.

One key characteristic of microcontrollers is their ability to execute dedicated firmware or software code stored in their internal memory. This code typically controls the behavior of the microcontroller, processes inputs from sensors or other external devices, and generates outputs to control actuators or display information.

The *Raspberry Pi Pico W* is a development board that utilizes the *RP2040* microcontroller chip. This board offers a range of features beyond its microcontroller, including onboard flash memory for program storage, versatile GPIO pins for interfacing with external devices, built-in USB connectivity for programming and power supply, and WiFi connectivity.

In comparison to single-board computers, the *Pi Pico* does not have an operating system. Instead, firmware can be loaded onto it, and it is this firmware that provides functionality to the board. This firmware allows the programmer to use programming languages such as C or MicroPython, which then control the microcontroller's behavior and interactions with external devices.

The absence of an operating system reduces the overhead associated with system management and resource allocation, resulting in faster boot times and improved reliability for time-critical tasks.

Taking into account our previously established requirements, which prioritize minimal points of failure, low computational demands, and cost-effectiveness, the logical preference leans towards the utilization of a microcontroller. For instance, single-board computers often rely on SD cards for storage, which can be prone to failure after prolonged use due to factors such as wear and tear or data corruption. In contrast, microcontrollers typically have simpler storage mechanisms, such as on-board flash memory, which are less susceptible to such issues. Thus, lower operating costs.

As a conclusion, a **microcontroller will be used**, and in particular, the ***Raspberry Pi Pico W***.

2.1.3. Electronic Modules

Chapter 3

Conclusiones y Trabajo Futuro

Conclusiones del trabajo y líneas de trabajo futuro.

Antes de la entrega de actas de cada convocatoria, en el plazo que se indica en el calendario de los trabajos de fin de grado, el estudiante entregará en el Campus Virtual la versión final de la memoria en PDF.

Contribuciones Personales

En caso de trabajos no unipersonales, cada participante indicará en la memoria su contribución al proyecto con una extensión de al menos dos páginas por cada uno de los participantes.

En caso de trabajo unipersonal, elimina esta página en el fichero `TFGTeXiS.tex` (comenta o borra la línea `\include{Capitulos/ContribucionesPersonales}`).

Estudiante 1

Al menos dos páginas con las contribuciones del estudiante 1.

Estudiante 2

Al menos dos páginas con las contribuciones del estudiante 2. En caso de que haya más estudiantes, copia y pega una de estas secciones.

Bibliography

Y así, del mucho leer y del poco dormir, se le secó el celebro de manera que vino a perder el juicio.
(modificar en Cascaras\bibliografia.tex)

Miguel de Cervantes Saavedra

Bautista, T., Oetiker, T., Partl, H., Hyna, I. y Schlegl, E. *Una Descripción de LaTeX 2ε*. Versión electrónica, 1998.

Knuth, D. E. *The TeX book*. Addison-Wesley Professional., 1986.

Krishnan, E., editor. *LaTeX Tutorials. A primer*. Indian TeX Users Group, 2003.

Lamport, L. *LaTeX: A Document Preparation System, 2nd Edition*. Addison-Wesley Professional, 1994.

Mittelbach, F., Goossens, M., Braams, J., Carlisle, D. y Rowley, C. *The LaTeX Companion*. Addison-Wesley Professional, segunda edición, 2004.

Oetiker, T., Partl, H., Hyna, I. y Schlegl, E. *The Not So Short Introduction to LaTeX 2ε*. Versión electrónica, 1996.

Appendix A

Título del Apéndice A

Los apéndices son secciones al final del documento en las que se agrega texto con el objetivo de ampliar los contenidos del documento principal.

Appendix B

Título del Apéndice B

Se pueden añadir los apéndices que se consideren oportunos.

Este texto se puede encontrar en el fichero Cascaras/fin.tex. Si deseas eliminarlo, basta con comentar la línea correspondiente al final del fichero TFGTeXiS.tex.

–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.

Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes

–Buena está – dijo Sancho –; fírmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.

Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes