Parking IoT

Memoria del proyecto final

Realizado por Ismael Aguilera Cervera y Pablo Sánchez Serrano

Contenido

1. Descripción del proyecto. 2
2. Componentes utilizados. 3
3. Conexión con la API. 4
   3.1. ThingSpeak 4
   3.2. Página Web. 4

1. Descripción del proyecto.

El proyecto consiste en el diseño, desarrollo e implementación de la maqueta de un parking IoT que se presenta en la Figura 1. Se ha diseñado una barrera para el parking, la cuál podrá abrirse pulsando un botón situado en la entrada. En cambio, para salir, un sensor de presencia detecta a los coches, haciendo que la barrera se levante automáticamente en presencia de un vehículo. Siempre que haya plazas disponibles en el parking, una luz verde estará iluminada. En caso contrario, se iluminará otra luz roja.

Figura 1. Maqueta del parking IoT.

El sistema lleva un conteo de los vehículos que han estacionado dentro. Cada vez que ingrese o salga un vehículo, se envía información a una página web. Dicha información se envía a través de una API vía conexión WiFi, cuya información se puede consultar desde el navegador a través de la propia API, como se puede ver en la Figura 2. El número de plazas ocupadas y disponibles se puede consultar mediante una página web, mostrada en la Figura 3.

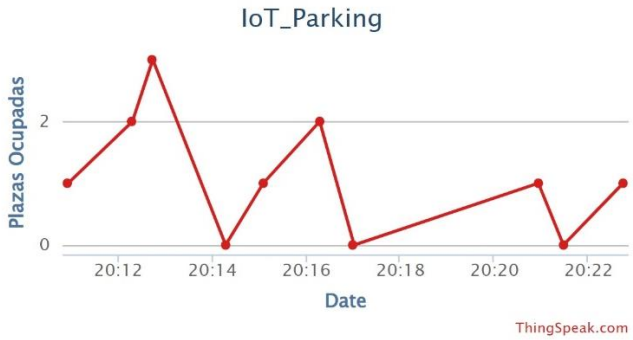

Figura 2. Gráfica del número de plazas ocupadas en ThingSpeak.

Figura 3: Página web.

2. Componentes utilizados.

Arduino: se ha utilizado este microcontrolador para el proyecto, el cual nos permite llevar un control centralizado del sistema. Gracias al IDE de Arduino, se pueden leer mensajes de log durante la ejecución del programa.

Botón: conectado al pin 3 y a una resistencia, se utiliza para dar la orden de subida de la barrera, permitiendo entrar al vehículo.

Servo: realiza un giro de 90º cuando un vehículo quiera entrar o salir, en función de si se ha pulsado el botón o el sensor de presencia ha detectado alguno. Está conectado al pin 4. Se le ha pegado una lámina y se ha situado a la pared para hacer la función de barrera.

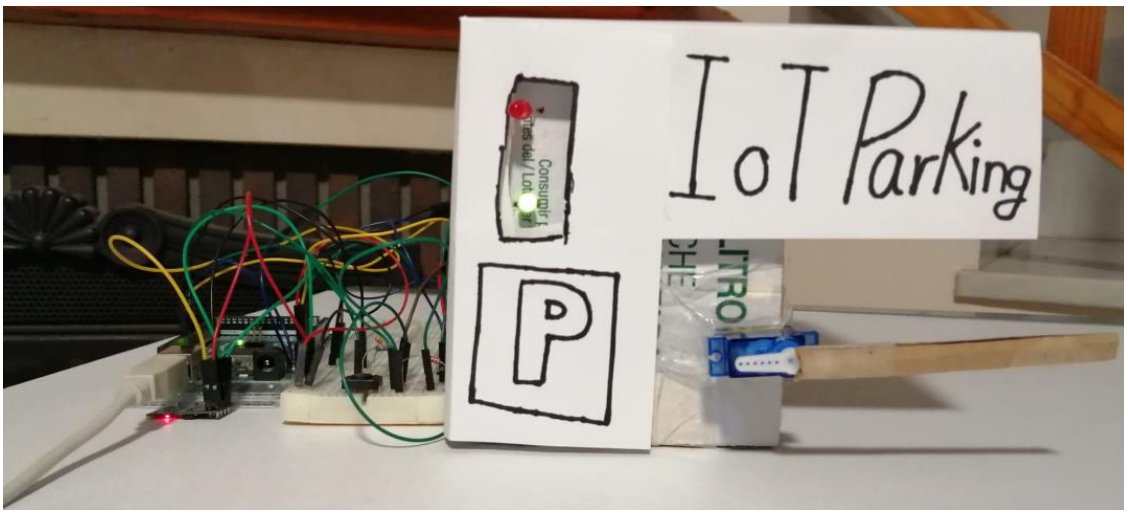

Figura 4: Vista frontal.

Sensor de presencia: está colocado en el interior, como se puede apreciar en la Figura 5, de forma que el sistema pueda saber que hay un vehículo esperando. A través de un agujero en la pared se conecta al pin 5 del Arduino.

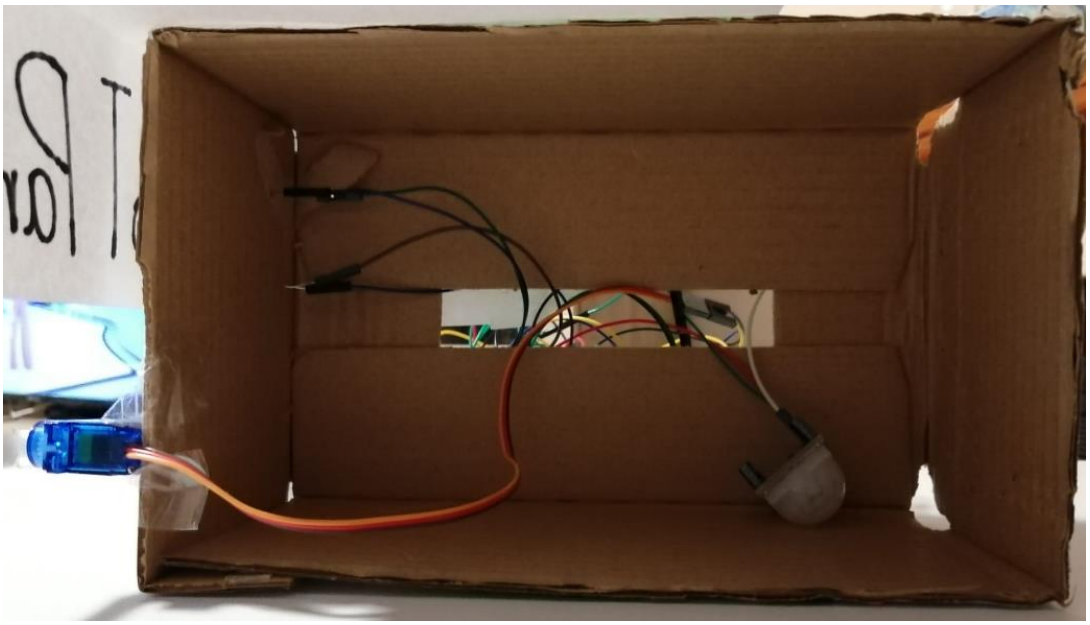

Figura 5: Vista interior donde se sitúa el sensor de presencia.

LEDs: son dos leds. Uno rojo y otro verde, conectados a los pines 6 y 7. Sirven para indicar si el parking está lleno (rojo) o hay algún hueco (verde).

ESP8266: dispositivo que otorga conexión WiFi, conectado a los pines 51 y 52. Gracias a él y a la librería *SoftwareSerial.h* se podrá hacer la conexión con la API de ThigSpeak, como se explica en el punto 3.

La implementación de la lógica que utiliza estos componentes se encuentra en el código proporcionado en la carpeta comprimida adjunta.

3. Conexión con la API.

3.1. ThingSpeak.

La integración con ThingSpeak permite enviar datos desde el Arduino a una plataforma en la nube, donde se pueden visualizar y analizar en tiempo real. ThingSpeak ofrece una API que permite enviar datos a través de conexiones HTTP. En este proyecto, se utiliza la API de ThingSpeak para enviar información sobre el número de plazas ocupadas en el parking, lo que permite generar gráficas como la mostrada en la Figura 2. Para establecer la conexión con ThingSpeak, se requiere configurar el ESP8266 para enviar los datos recopilados por el sistema mediante la API proporcionada por ThingSpeak.

3.2. Página Web.

La página web proporciona una interfaz para visualizar el estado del parking IoT desde cualquier navegador. Muestra información como el número de plazas ocupadas y disponibles, como se muestra en la Figura 3. Para implementar esta funcionalidad, se desarrolla una aplicación web que consume los datos proporcionados por el ESP8266 a través de su conexión WiFi accediendo a la API de ThingSpeak. La página web se actualiza dinámicamente para reflejar los cambios en tiempo real, permitiendo a los usuarios monitorear el estado del parking IoT.