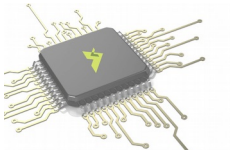
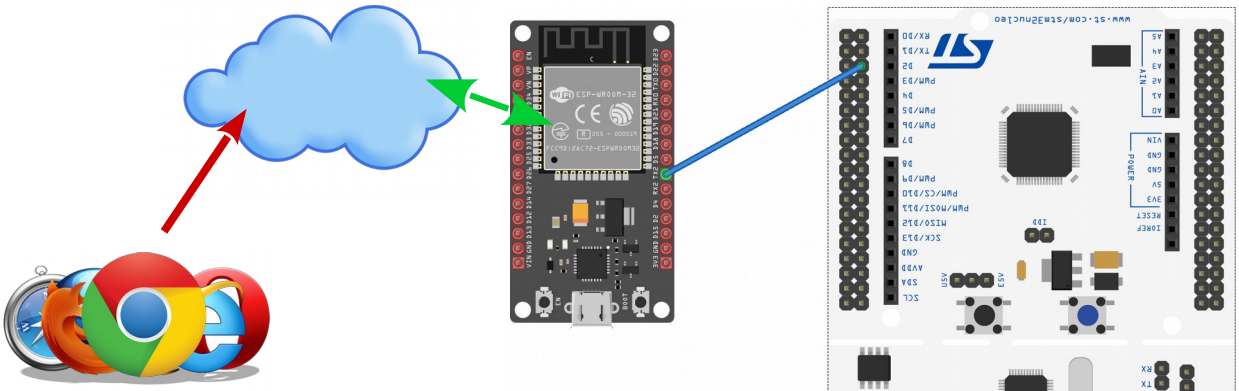


– Lab 8: ESP32, MicroPython and API REST –

Puntuación: 4/25 (en fecha)
2/25 (fuera de fecha)
Modalidad: Parejas
Sesiones: 1

Este lab pretende incorporar un nuevo micro de la familia ESP32, en concreto el ESP32-WROOM-32, junto con el lenguaje de programación Python en su versión MicroPython.

La comunicación entre el micro ESP32 y el F411RE se realizará mediante la UART como muestra la figura (dependiendo de cada diseño, es posible que los pines utilizados sean diferentes) , es decir, el ESP32 será el encargado de enviar información al F411RE. A su vez el ESP32 estará conectado via WiFi, de forma que los usuarios pueden enviar órdenes al F411RE a través de una API REST (se proporciona plantilla).



http://192.168.1.100/red/0
http://192.168.1.100/green/0
http://192.168.1.100/restore/0

Las órdenes enviadas al F411RE contienen un formato específico, compuesto por: *identificador de grupo* de 8 bits, seguido de un *identificador de acción* de 8 bits (RED:0, GREEN:1 y RESTORE: 2). El *identificador de grupo* deberá ser comprobado en el F411RE descartando la acción a realizar si no corresponde con el identificador de grupo asignado (número de la caja entregada al inicio del curso), mientras que el *identificador de acción* permitirá cambiar el comportamiento del semáforo de la siguiente forma:

- El usuario envía *green* (acceder a *http://<ip>/green/<id>,* donde *ip* es la ip asignada e *id* es el identificador de grupo), el semáforo para vehículos permanecerá en verde hasta que se reciba una acción de *restore*.
- El usuario envía *red* (acceder a *http://<ip>/red/<id>,* donde *ip* es la ip asignada e *id* es el identificador de grupo), el semáforo para vehículos permanecerá en rojo, pasando por ámbar, hasta que se reciba una acción de *restore*.

Por ejemplo, el mensaje *00* indica que la acción va dirigida al grupo 0 para que su semáforo permanezca en verde hasta nueva orden. Se debe tener en cuenta que la secuencia de cambios del semáforo continuará de acuerdo a las anteriores entregas. Para el ejemplo anterior, si el semáforo para vehículos se encuentra en rojo antes de pasar al verde se deberá poner intermitente el semáforo de peatones. Recordad que hasta que no llegue una acción de restore el semáforo no atenderá peticiones de peatones, es decir si alguien pulsa el botón no se tendrá en cuenta hasta que no se restaure el funcionamiento normal.

Las transiciones deben realizarse lo más rápido posible para conseguir un alto grado de capacidad de respuesta (responsiveness). Por lo tanto, las transiciones rápidas serán de la siguiente forma:

Semáforo vehículos	Comando	Acción
Verde	green	Bloquear el pulsador y mantener el estado actual
Rojo	green	Iniciar parpadeo de semáforo peatones. Seguidamente se pasa a verde y se bloquea el pulsador
Ámbar	green	No se pasa a rojo el semáforo de vehículos ni a verde el de peatones, directamente se vuelve a verde vehículos y se bloquea el pulsador
Verde	red	Misma acción que si se pulsara el botón (comprobar si coches cercanos, pasar a ámbar, ...). También se bloquea el pulsador
Rojo	red	Bloquear el pulsador y mantener el estado actual
Ámbar	red	Pasar el semáforo de vehículos a rojo tras el tiempo del ámbar y bloquear el pulsador

Se deberá entregar el código desarrollado (directorios *src* e *include* para F411RE, *srcPy* para MicroPython) junto con una breve explicación del mismo.

Nota: La parte del F411RE se puede realizar haciendo uso de la aplicación bluetooth que instalasteis en la entrega del lab 6 para posteriormente sustituir dicha comunicación por el cable que conectan ambas placas.