UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

S6 | Interrupciones y MQTT

Máquinas Digitales

Hugo Miranda Cano

Interrupciones

- Video del monitor serie mostrando el conteo de veces que se ha presionado el botón, usando el programa "IntButton". Se tiene que ver que se está presionando el botón.

Video en GiHub

- Responder: ¿Qué ventaja tienen las interrupciones contra el polling? ¿Por qué es más adecuado usarlas para leer un botón?

Una interrupción es una suspensión temporal de la ejecución de un proceso para pasar a ejecutar una subrutina de servicio de interrupción , la cual no es parte del programa, posteriormente se reanuda el programa.

Una de las ventajas que tiene las interrupciones del polling es que en las interrupciones no se requiere estar sondeando o revisando si algo va mal, cosa contraria con el polling. En el caso del programa realizado en la clase, al utilizar polling el contador se actualizaba incrementando considerablemente el número de conteos cuando se apretó el botón de las que realmente se presionó. podríamos decir que el tiempo de respuesta no era inmediato, sin en cambio, al utilizar una interrupción, se ejecutaba el pedazo de código haciendo que el botón no tuviera ese ruido y por ende las pulsaciones o actualizaciones eran mas fieles , después de ejecutarse el código el programa se reanudaba teniendo así la mejor actualización del contador.

- Explicar qué es el rebote del botón (bouncig) y explicar sus técnicas de mitigación: Por software y por hardware.

Para entrar en contexto: el bouncing es la señal de ruido producido por pasar de un estado a otro estado.

Podemos retirar o quitar este ruido de dos maneras primero debounce por software y segundo por hardware. Comenzaremos por el debounce por hardware

El aplicar un debounce por hardware no incrementa el tiempo de ejecución del programa, es una solución más factible, pero la desventaja es que se vuelve más complejo nuestro esp32. Para aplicar el debounce ya mencionado se coloca un condensador (1 microfaradio) en paralelo con el dispositivo.

Ahora la segunda manera de quitar ese ruido es el debounce por software tiene la ventaja de no requerir componentes adicionales. Se resuelve únicamente modificando el código del programa, una desventaja es que incrementa el tiempo de ejecución del programa y la complejidad del programa.

La manera mas sencilla de aplicar debounce por software es comprobar el tiempo entre disparos de la interrupción, si el tiempo es inferior a un determinado umbral de tiempo simplemente ignoramos la interrupción. Para aplicar el debounce por software, se modifica la función ISR.

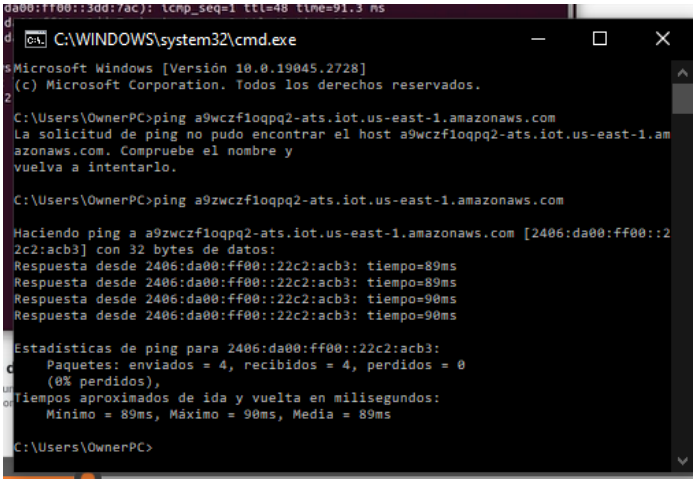
- (Opcional) Responder: ¿Cuál es la duración máxima default de una función que se invoca por interrupción en ESP32? ¿Se puede usar un Serial.println() dentro de un ISR?
- Push del programa "IntButton""

MQTT

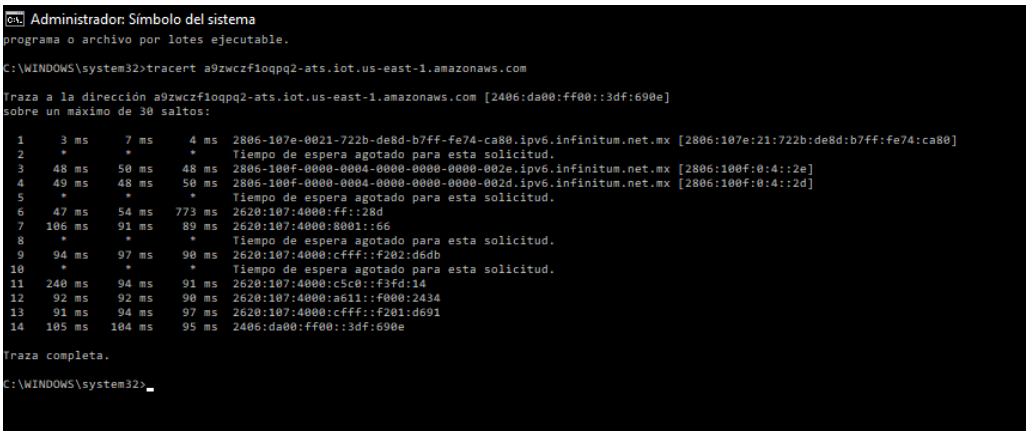
- Captura con las pruebas de alcance del endpoint del servidor MQTT (Broker MQTT) de Amazon. Las pruebas se hacen desde la terminal del sistema operativo: 1) ping, 2) traceroute(Linux) o tracert(Windows) y 3) telnet. Explicar cuál es el propósito de cada una.

Telnet es un programa que permite la comunicación directa con la consola del otro lado. (Mensaje sin cifrar).

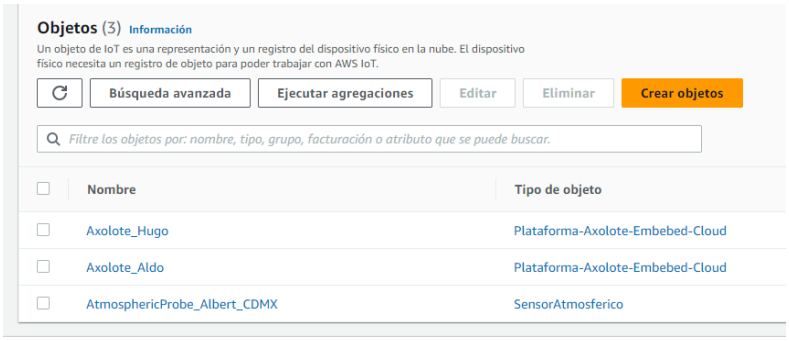
Ping sirve para ver si se puede conectar a determinado punto de enlace.

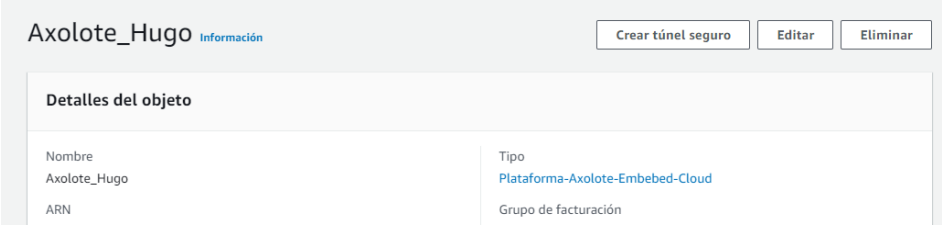


Tracert(Windows) nos indica que servidores y routers hay entre mi computadora y el servidor.(servidores intermedios) también nos indican el tiempo entre routers.

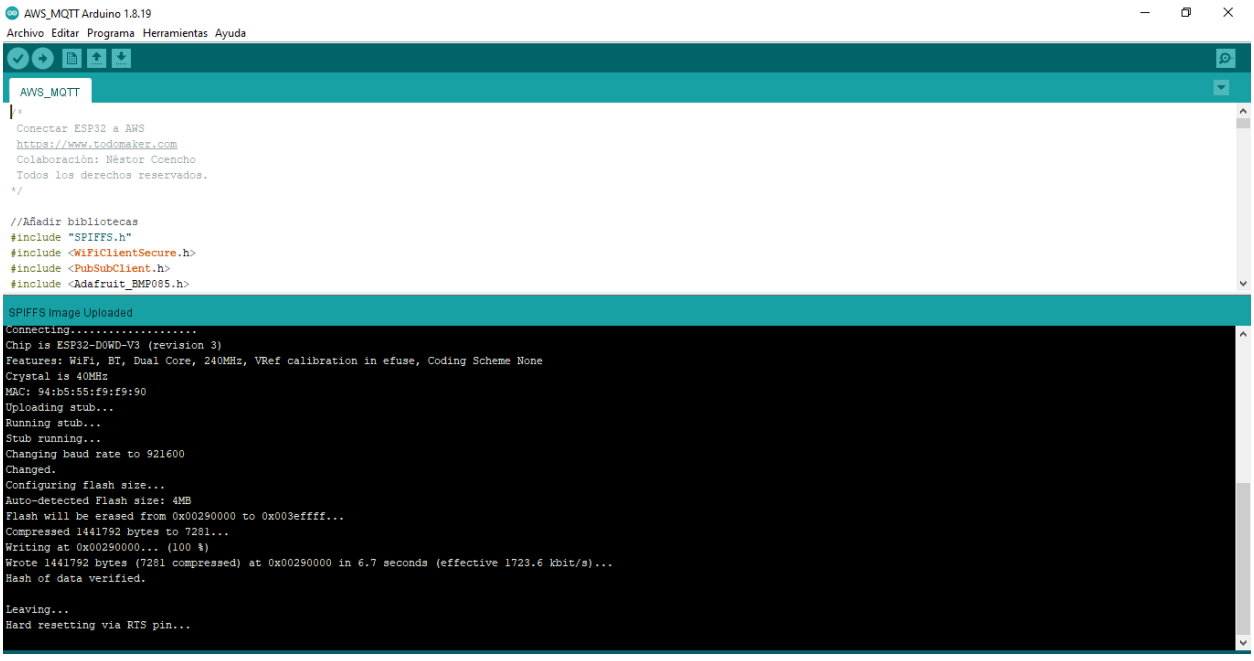


- Captura del objeto Axolote_{tu_nombre} en la consola de administración de IoT Core.

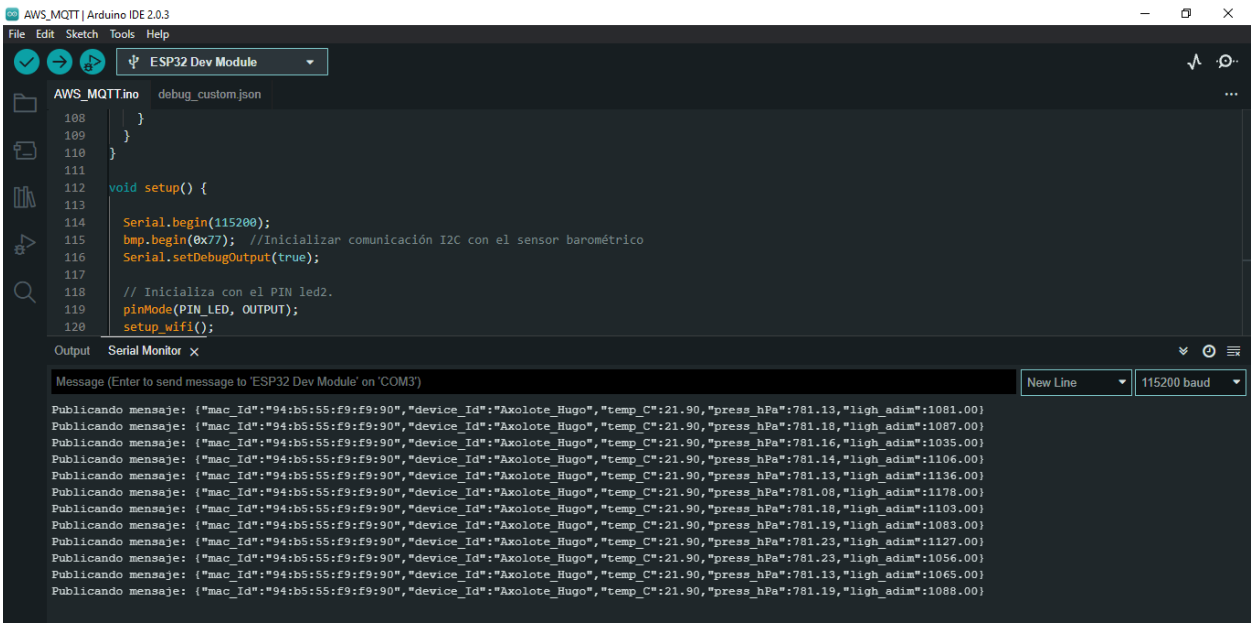




Subimos certificados al esp32 sketch data upload

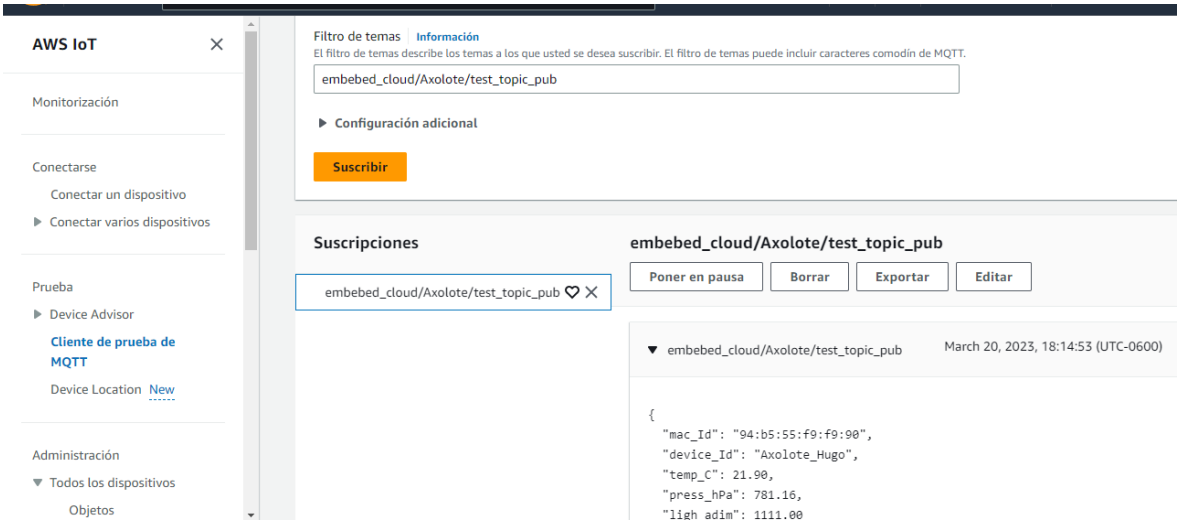


- Captura de la terminal serie donde se muestre que el ESP envía lecturas al servidor MQTT.



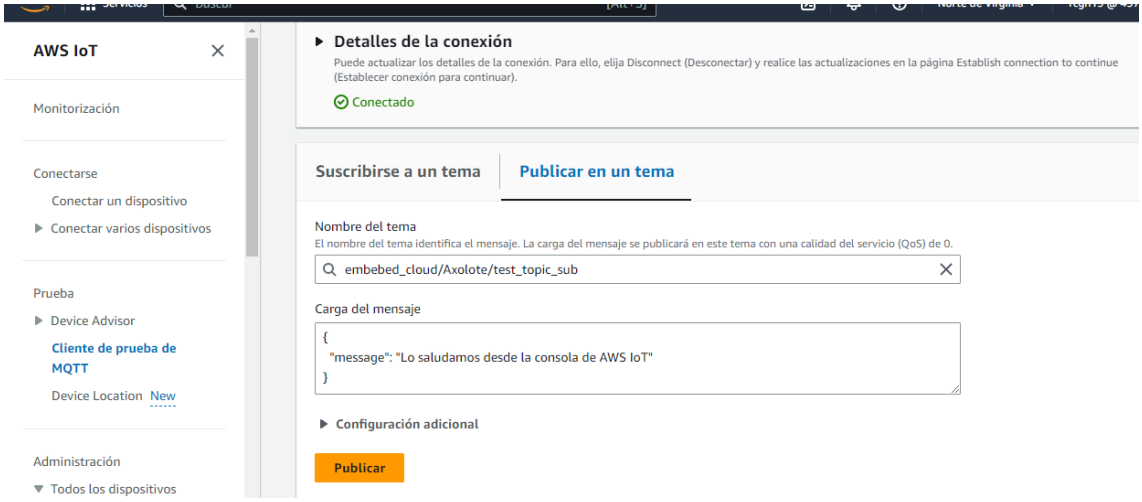
- Captura del cliente de prueba de IoT Core leyendo los mensajes que llegan al servidor MQTT.

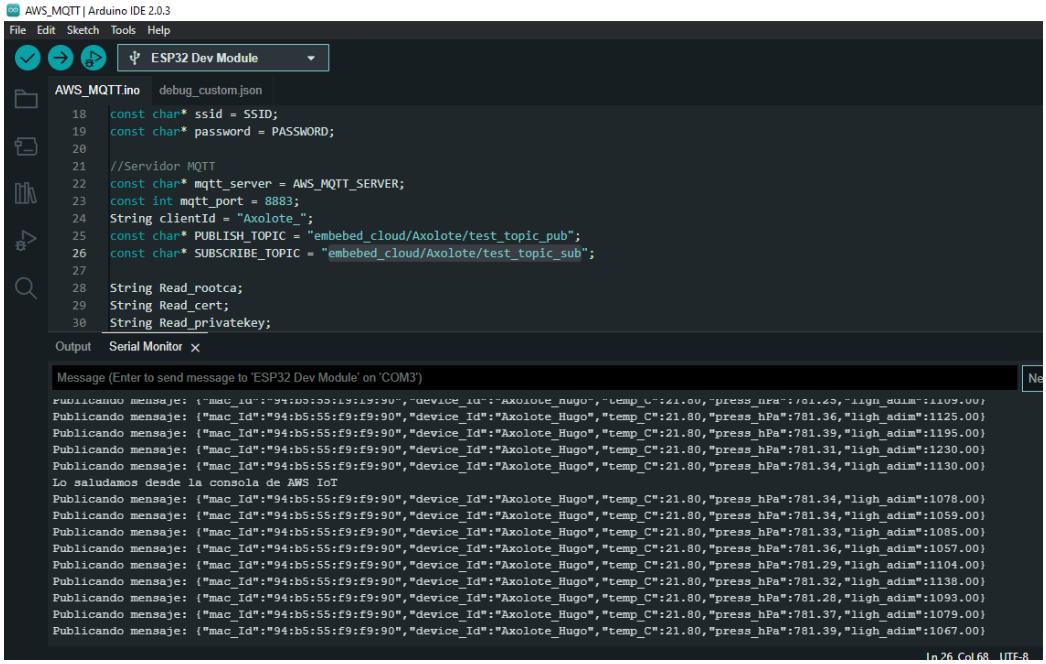
Creando un cliente de prueba



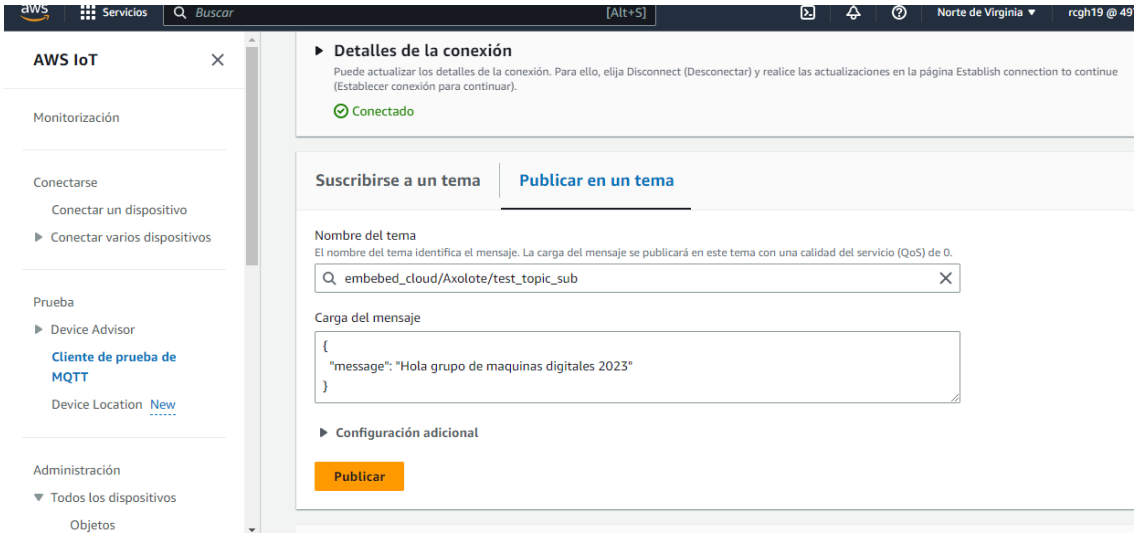
- Video que muestre que se envía un mensaje desde el cliente de prueba hacia el tópico al que está suscrito el ESP. En la terminal serial debe verse que el ESP recibe el mensaje y lo despliega.

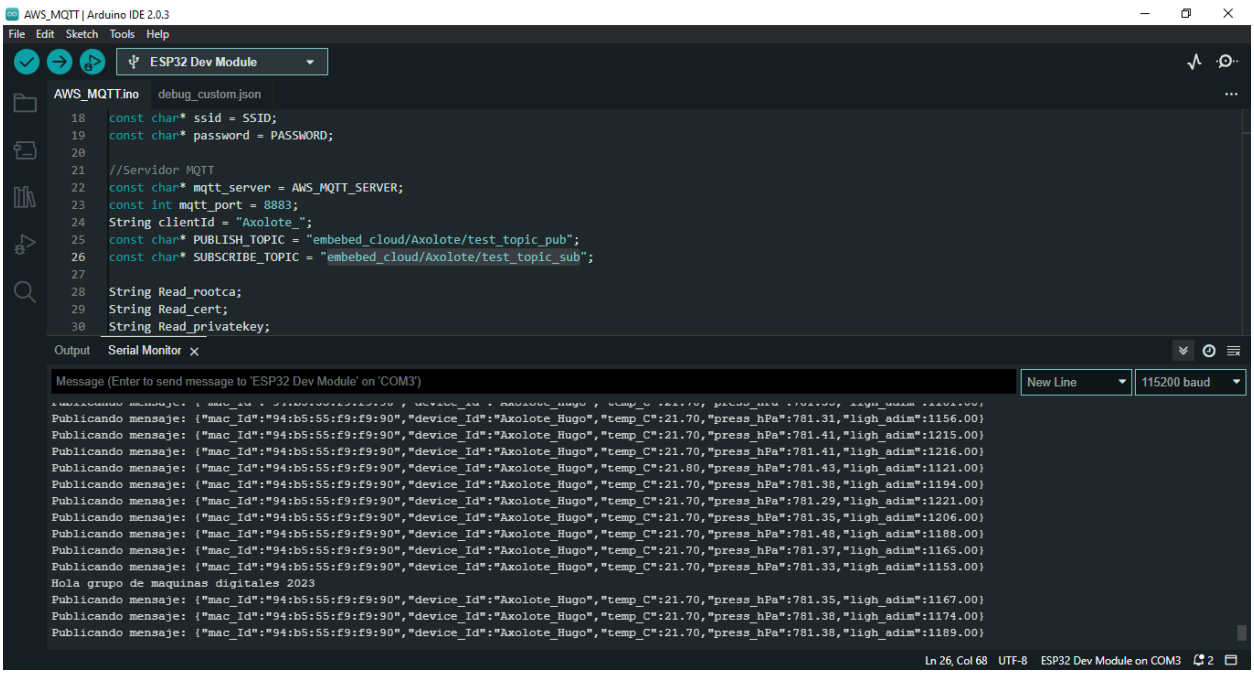
Captura de mensaje de cliente a esp32





Segundo mensaje





- Push con el programa "aws_mqtt".