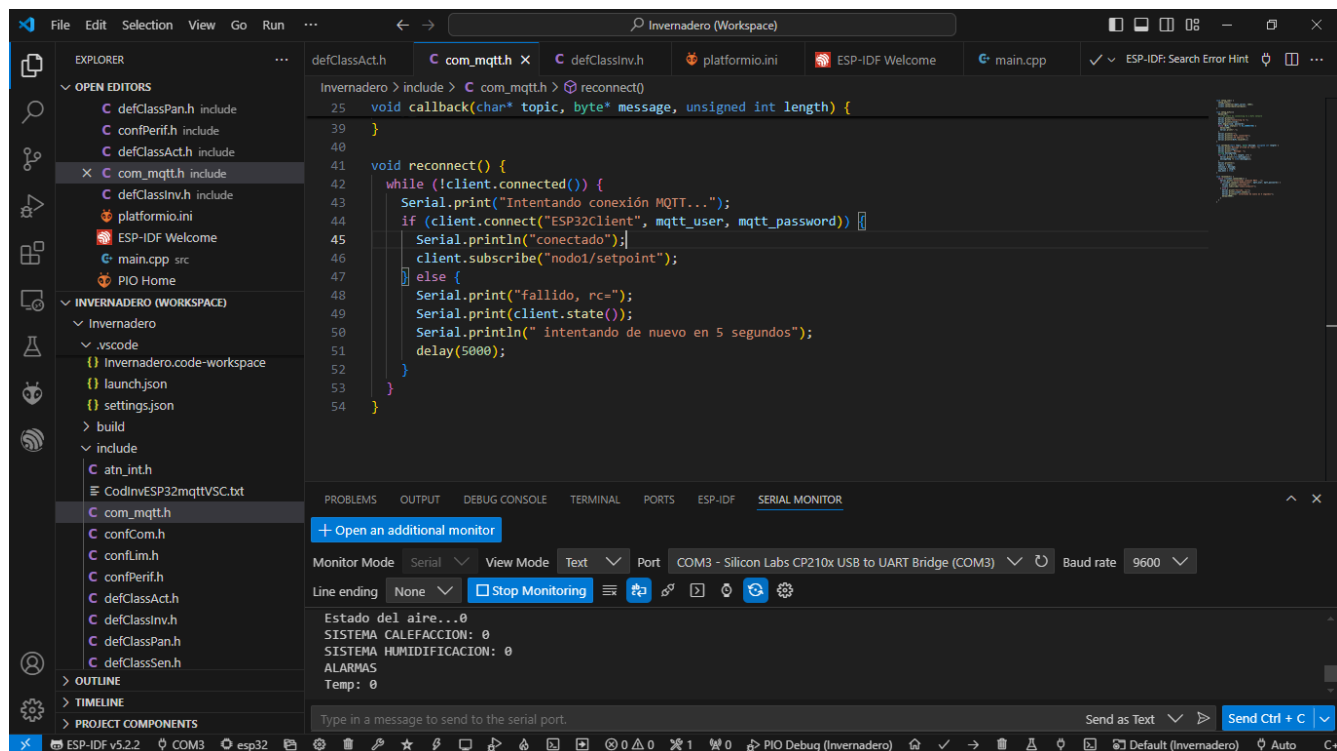


LED[7]	Salida digital	GPIO 2, 5, 16, 17, 18, 19, 23	GPIO 2, 5, RX2, TX2, 18, 19, 23	
Pantalla OLED	I2C	SDA, SCL	GPIO 21, 22	wire 1
UART0		Rx0, Tx0	GPIO 3, 1	Serial
SPI	VSPI	COPI, CIPO, SCK, CS	GPIO 23, 19, 18, 5	
ESP-PROG	JTAG	MTDI, MTCK, MTMS, MTDO	GPIO 12, 13, 14, 15	
VIN	Fuente ext	4 a 7 V		
ENABLE	Reset	Tiene pull-up		

El ajuste fue definido en el archivo de ubicación de periféricos del proyecto. Si bien, el entorno de trabajo de Arduino Ide resulta conveniente gracias a la facilidad de integración de librerías y curva de aprendizaje desarrollada por proyectos previos, es necesario el uso de Visual Studio Code como entorno de desarrollo integrado para la propuesta. Después de la instalación se probó la instalación del complemento de ESP-IDF como soporte para los ESP32, así como la instalación de Arduino para Vscode. El primero presentó incompatibilidad en el código generado y el segundo demostró tener limitaciones para la integración del depurador. Finalmente se optó por realizar la programación desde los recursos proporcionados por Platformio, que permite la compilación, descarga al dispositivo embebido, el monitoreo a través del puerto serial y la depuración en línea de código.

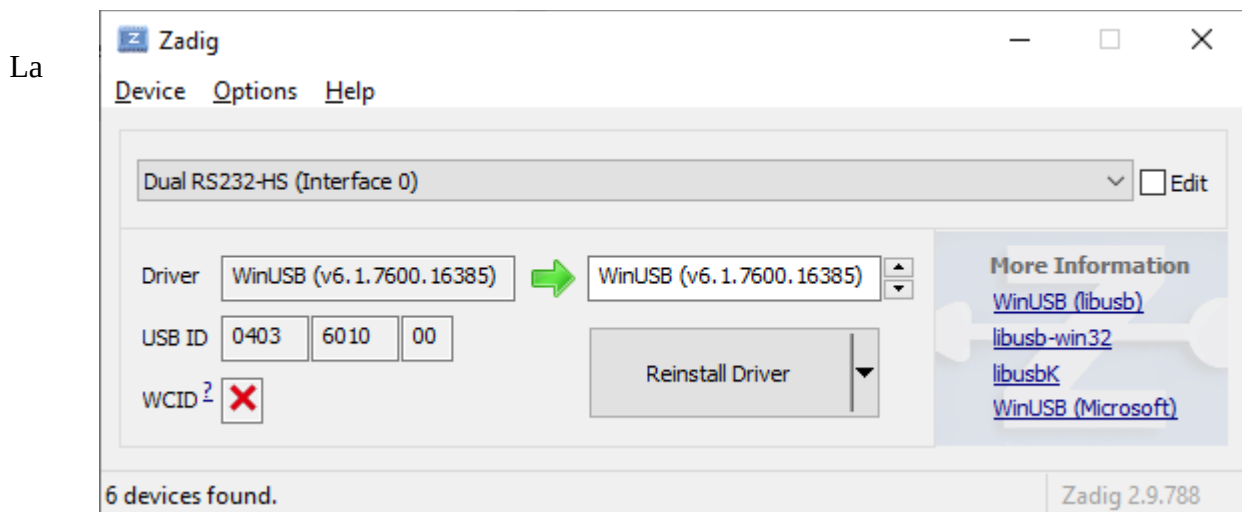


Visual Studio Code con edición de código y monitoreo a través de puerto serial.

Configuración de platformio para habilitación de depurador. En el archivo platformio.ini se agregan las siguientes líneas:

```
[env:esp32doit-devkit-v1]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
monitor_speed = 115200
lib_deps = bblanchon/ArduinoJson@^7.1.0, adafruit/DHT sensor library@^1.4.6,
knolleary/PubSubClient@^2.8, adafruit/Adafruit Unified Sensor@^1.1.14
upload_port = COM3
debug_tool = esp-prog
debug_init_break = tbreak setup
```

Por otra parte, desde el entorno del sistema operativo ha sido necesaria la ejecución del programa zadig para definir de los dos puertos disponibles del depurador esp-prog la opción de depuración JTAG y la de programación.

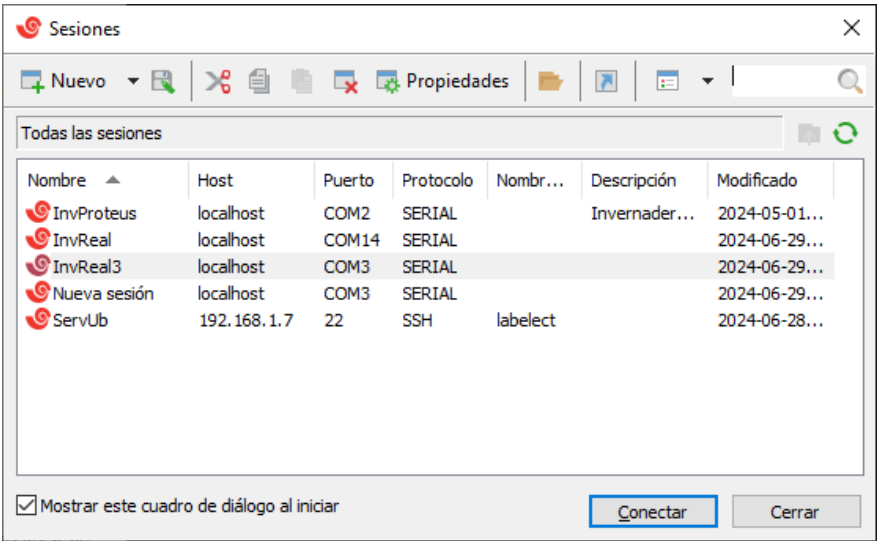


interfaz cero es asociado el driver desde la aplicación, habilitando la depuración a través del puerto JTAG.

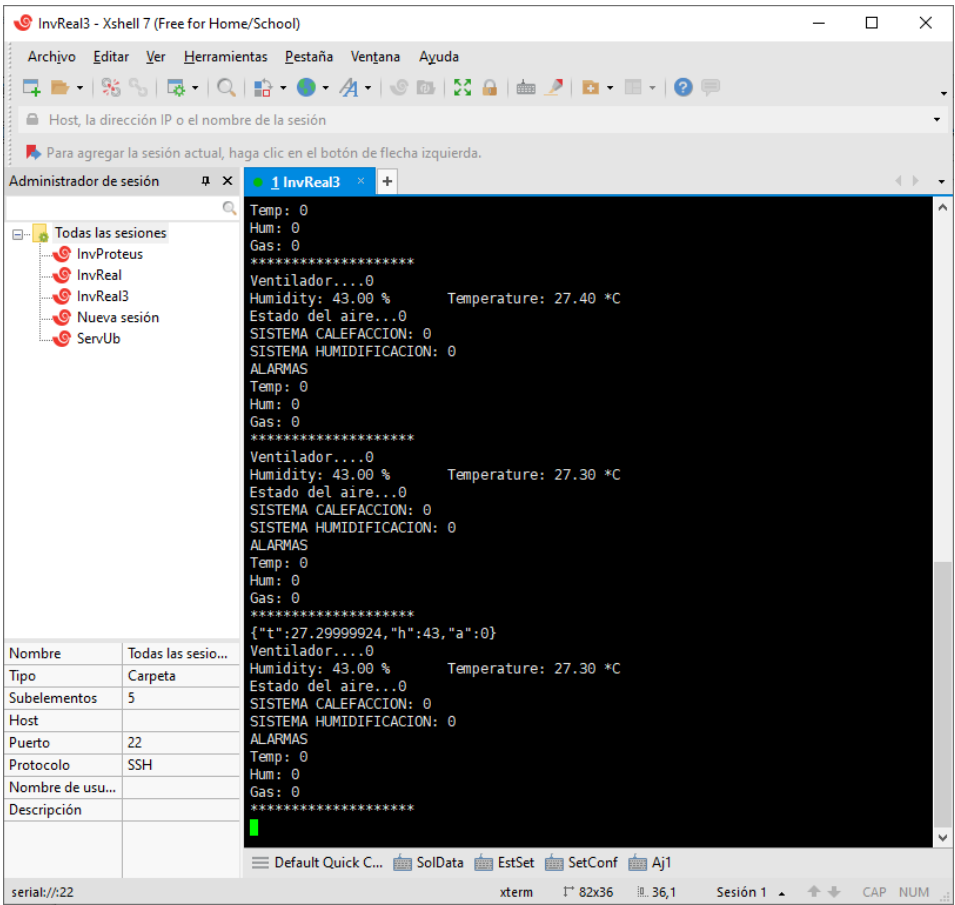
Por otra parte se integran las funciones de publicación y suscripción a través del protocolo de MQTT. Se publican tres variables temperatura, humedad y calidad del aire a través del tópico nodo1/var, usando la estructura JSON empaquetando las tres variables en un mensaje. La suscripción se realiza al tópico nodo1/setpoint, de manera que recibe las publicaciones asociadas procedentes desde el servidor donde se ha habilitado un dashboard con una página dedicada a la presentación de valores límites, siendo posible la publicación de una forma donde se dirige la nueva información que tiene un carácter persistente para los últimos datos proporcionados de configuración.

La migración generó una serie de inconvenientes asociados con la configuración e inicialización de puertos, en principio identificados con la visualización disponible en el panel de leds, pero que requirió el seguimiento en el código. Algotos problemas exigieron la depuración, para este caso mediante el uso de esp-prog, que es una herramienta que permite la prueba de ejecución de código mediante la ubicación de breakpoints para el seguimiento del flujo del programa y del estado de variables. En todo caso, el uso de depuración a través del puerto serial permitió el desarrollo de pruebas y verificación de resultados. Se uso putty para la creación de sesiones seguras mediante interfaz tipo texto, sin embargo, la limitación para preservar comandos definidos en el protocolo de intercambio de datos, dio lugar a la elección de xshell como entorno de depuración serial.

Se habilita además la funcionalidad de monitoreo que dirige el estado del sistema a través del puerto serial, quedando pendiente la programación de una pantalla OLED para el despliegue de información del proceso.



Definición de sesiones en Xshell para comunicación con el controlador.



Se observa la actualización periódica cada segundo de la información, así como la publicación de las medidas cada cinco segundos.

Se incorpora pantalla Ole i2c, se corrige error de activación inestable de actuadores definiendo correctamente los pines de interrupción. La situación es corregida con ayuda del depurador.

Para agregar información de clima procedente de la web, se sugiere los siguientes vínculos:

<https://github.com/ThingPulse/esp8266-weather-station?tab=readme-ov-file>

Implementación de una estación de clima con esp8266 o esp32. Definen la librería del nodo.

De igual forma se encuentran desarrollos con OTA para esp32 para el ajuste de parámetros de red inalámbrica a través de interfaz de usuario.