ezurio

Wi-Fi Developer Journey:

Python Development on the Veda SL917 Module

Low-power Embedded Wi-Fi and BLE applications with Python

AP Mode Provisioning + Wi-Fi Connectivity Demo

CANVAS
SOFTWARE SUITE

Agenda

- Canvas Software Suite Introduction
- Veda SL917 Hardware Overview
- Development Tools Installation and Board Setup
- Creating a Python Application
- Python REPL and Built-in Modules
- Canvas APIs and Samples
- Python Application Deep Dive

Presenter:
Scott Lederer – Staff Software Engineer, Ezurio

CANVAS
SOFTWARE SUITE

WHAT IS CANVAS SOFTWARE SUITE?

FIRMWARE

Canvas enables Python scripting on Ezurio MCU-based wireless products.

TOOLS

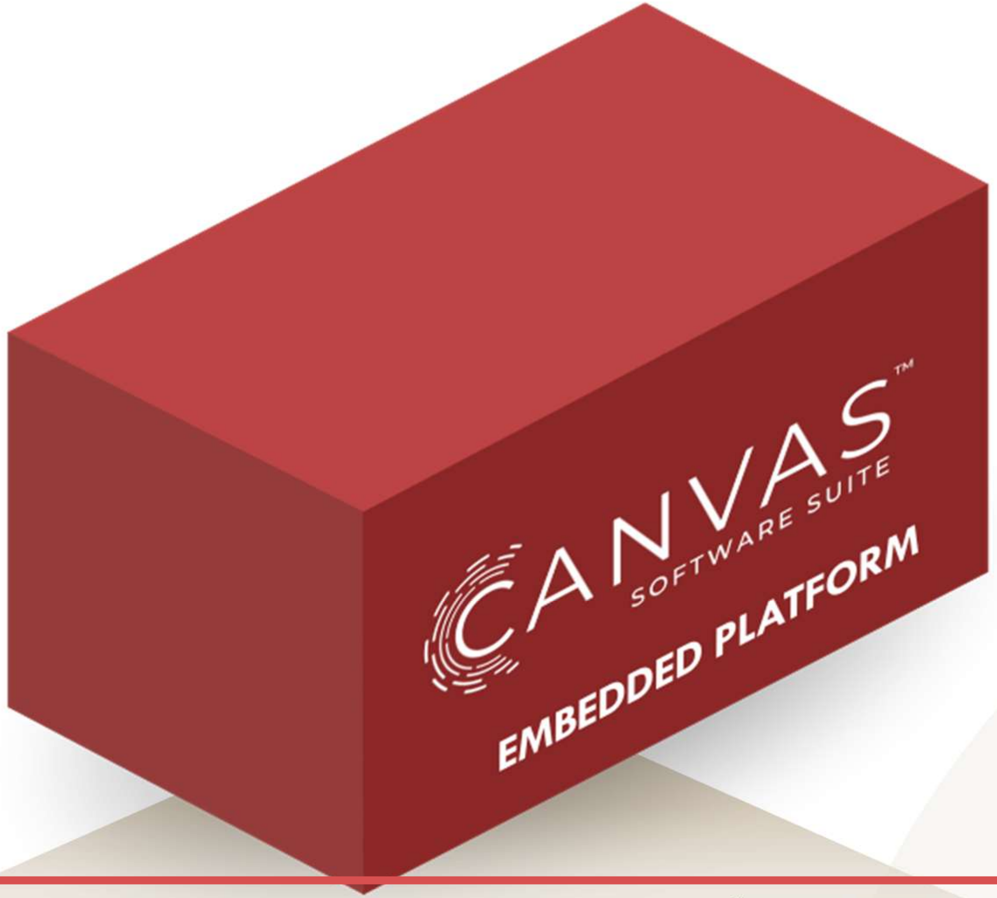
Development tools to help you rapidly iterate your embedded applications.

SAMPLES

Get a head start on your design with existing demos and code snippets.

CANVAS SOFTWARE SUITE - FIRMWARE

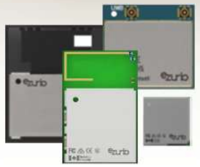
A **Firmware Platform** on the embedded hardware provides a scripting engine and APIs for end-product manufacturers to build applications on top of.



Ezurio Supported Platform Firmware

Ezurio provides wireless connectivity hardware platforms.

Ezurio MCU-based Modules



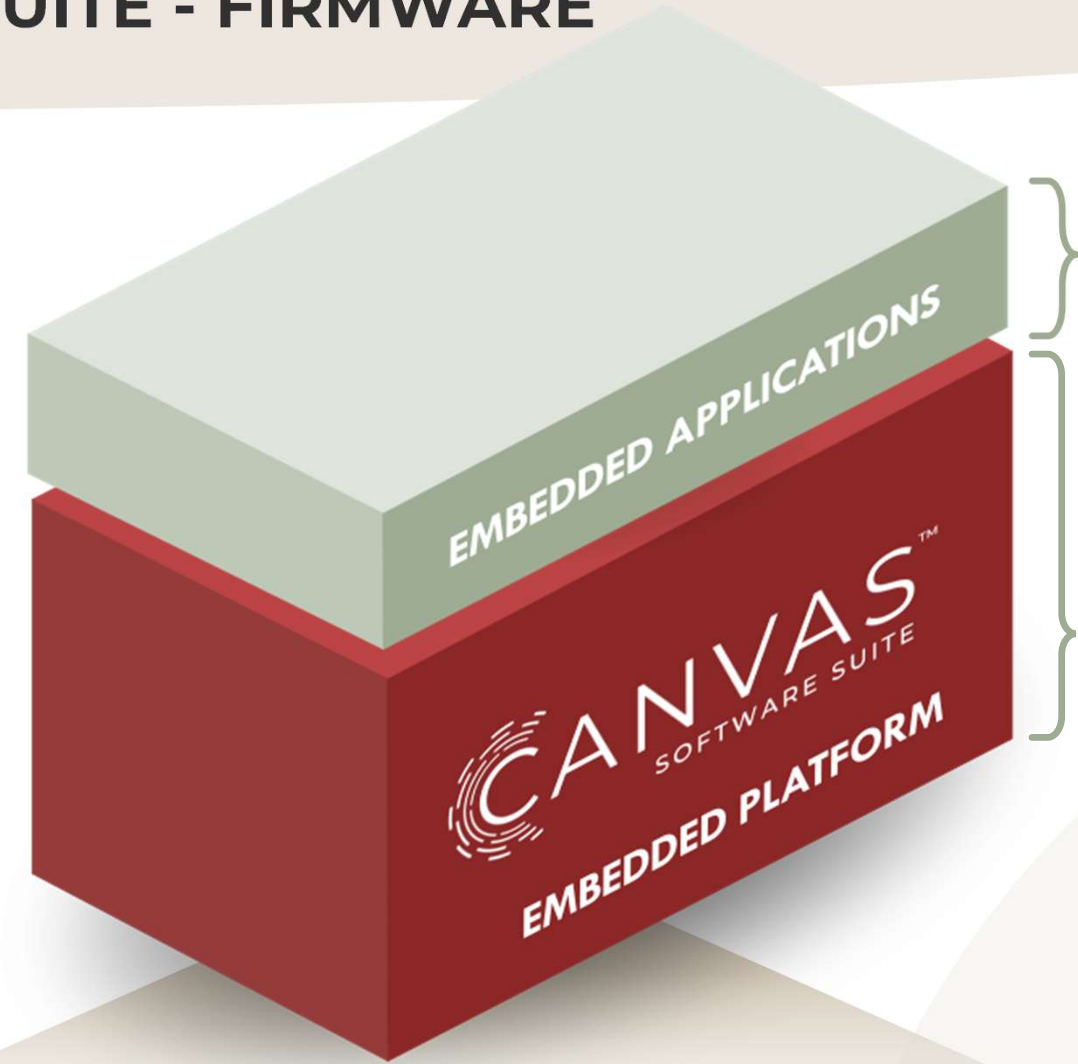
Ezurio IoT Gateways and Sensors

CANVAS SOFTWARE SUITE - FIRMWARE

Wireless end-product manufacturers develop **Application Scripts** targeting their end-product use case.

A **Firmware Platform** on the embedded hardware provides a scripting engine and APIs for end-product manufacturers to build applications on top of.

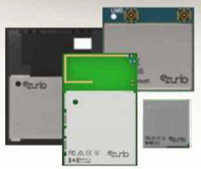
Ezurio provides wireless connectivity hardware platforms.



End-Product Specific Application Scripts

Ezurio Supported Platform Firmware

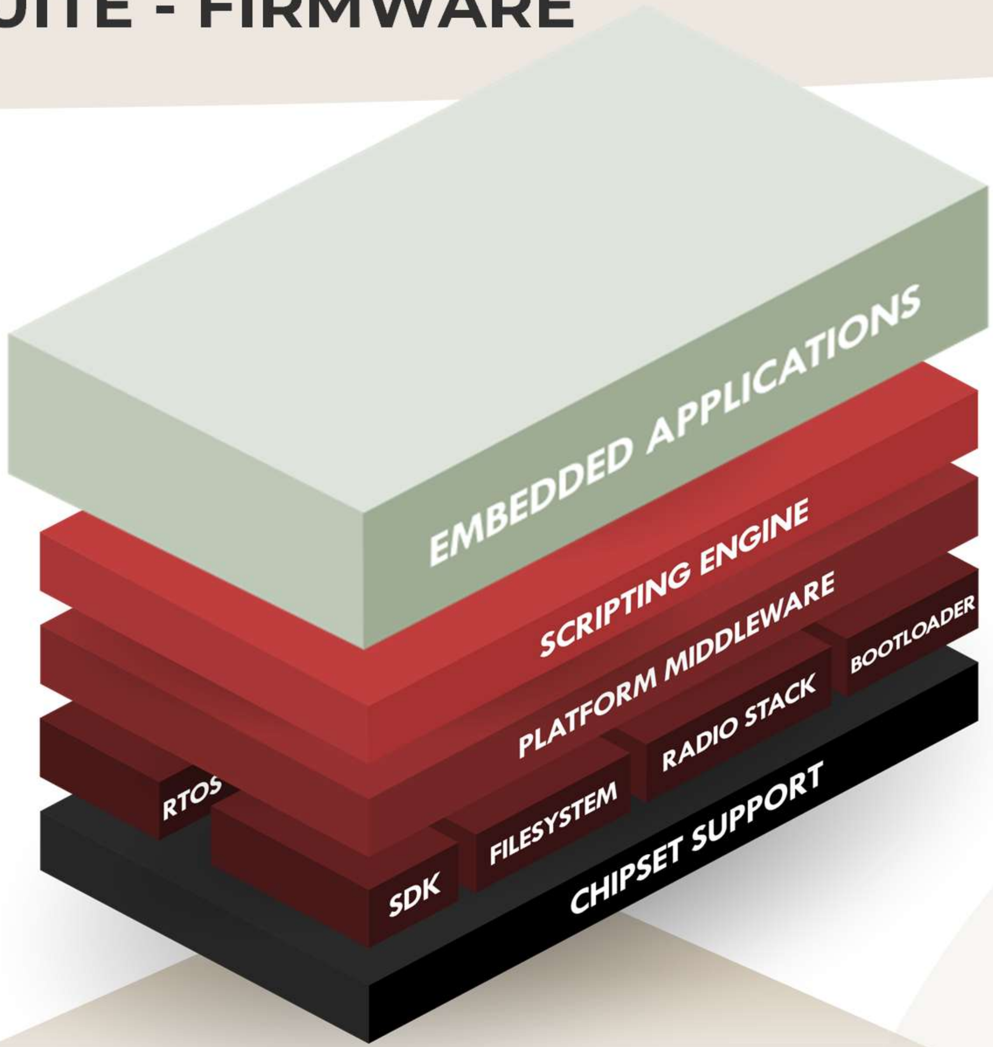
Ezurio MCU-based Modules



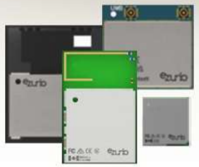
Ezurio IoT Gateways and Sensors

CANVAS SOFTWARE SUITE - FIRMWARE

Canvas firmware abstracts the details of RTOS, chipset and wireless radio stacks, streamlining customer development of wireless applications.



Ezurio
MCU-based
Modules

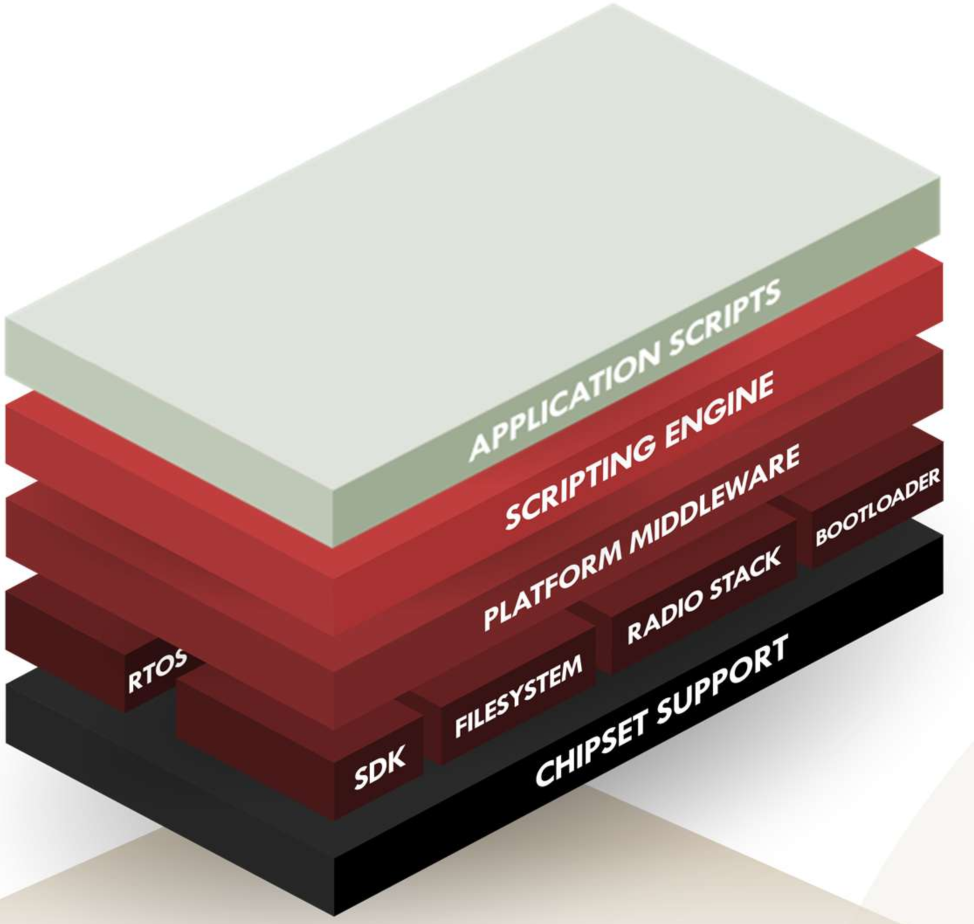


Ezurio
IoT Gateways
and Sensors

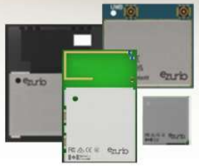
CANVAS SOFTWARE SUITE - FIRMWARE

End-product designers can focus on their target applications and deliver solutions faster with Python.

Canvas firmware abstracts the details of RTOS, chipset and wireless radio stacks, streamlining customer development of wireless applications.



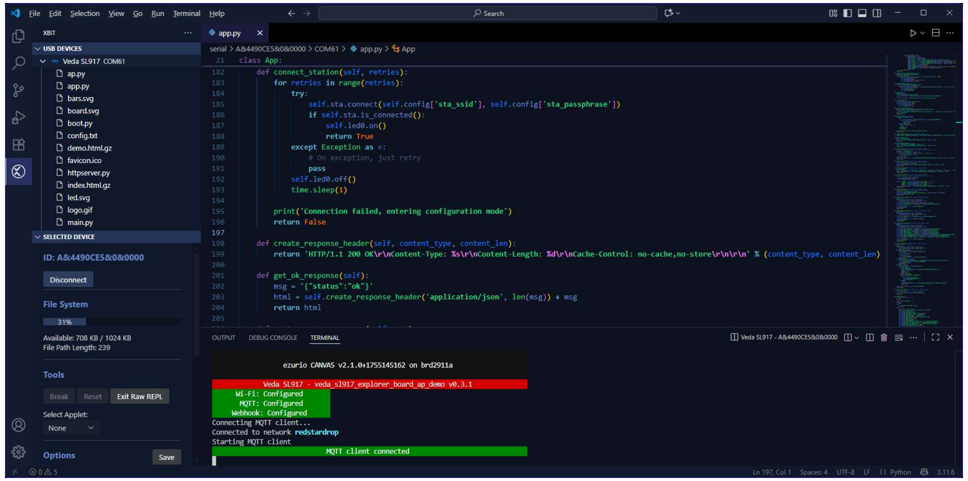
Ezurio MCU-based Modules



Ezurio IoT Gateways and Sensors

CANVAS SOFTWARE SUITE - TOOLS

Xbit for Visual Studio Code



ID: A&4490CE5&0&0000

File System

31%

Available: 708 KB / 1024 KB
File Path Length: 239

Tools

Select Applet: None

Options

```
class App:
    def connect_station(self, retries):
        for retries in range(retries):
            try:
                self.sta.connect(self.config['sta_ssid'], self.config['sta_passphrase'])
                if self.sta.is_connected():
                    self.led0.on()
                    return True
            except Exception as e:
                # On exception, just retry
                pass
            self.led0.off()
            time.sleep(1)

        print('Connection failed, entering configuration mode')
        return False

    def create_response_header(self, content_type, content_len):
        return 'HTTP/1.1 200 OK\r\nContent-Type: %s\r\nContent-Length: %d\r\nCache-Control: no-cache,no-store\r\n\r\n' % (content_type, content_len)

    def get_ok_response(self):
        msg = '{"status":"ok"}'
        html = self.create_response_header('application/json', len(msg)) + msg
        return html
```

```
            ezurio CANVAS v2.1.0+1755145162 on brd2911a

        Veda SL917 - veda_sl917_explorer_board_ap_demo v0.3.1
    Wi-Fi: Configured
     MQTT: Configured
  Webhook: Configured
Connecting MQTT client...
Connected to network redstardrop
Starting MQTT client
                       MQTT client connected
```

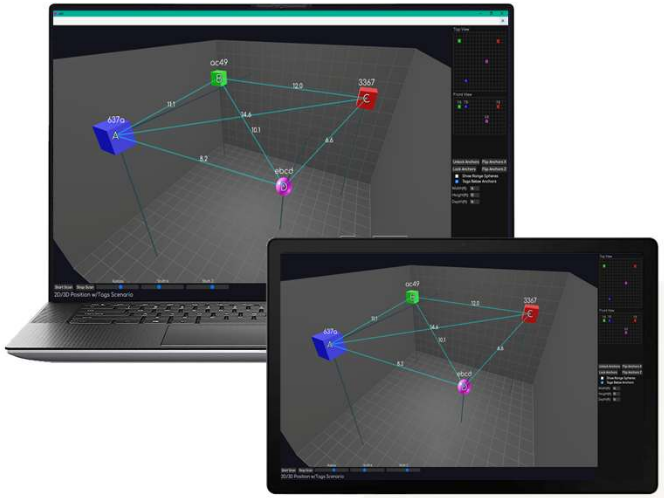
Develop, load and interact with on-device Python scripts within **VS Code**.

Access the Python REPL to quickly test APIs and prove out application code.

Available on Visual Studio Marketplace

Xbit tools for VS Code

Xbit Desktop/Mobile



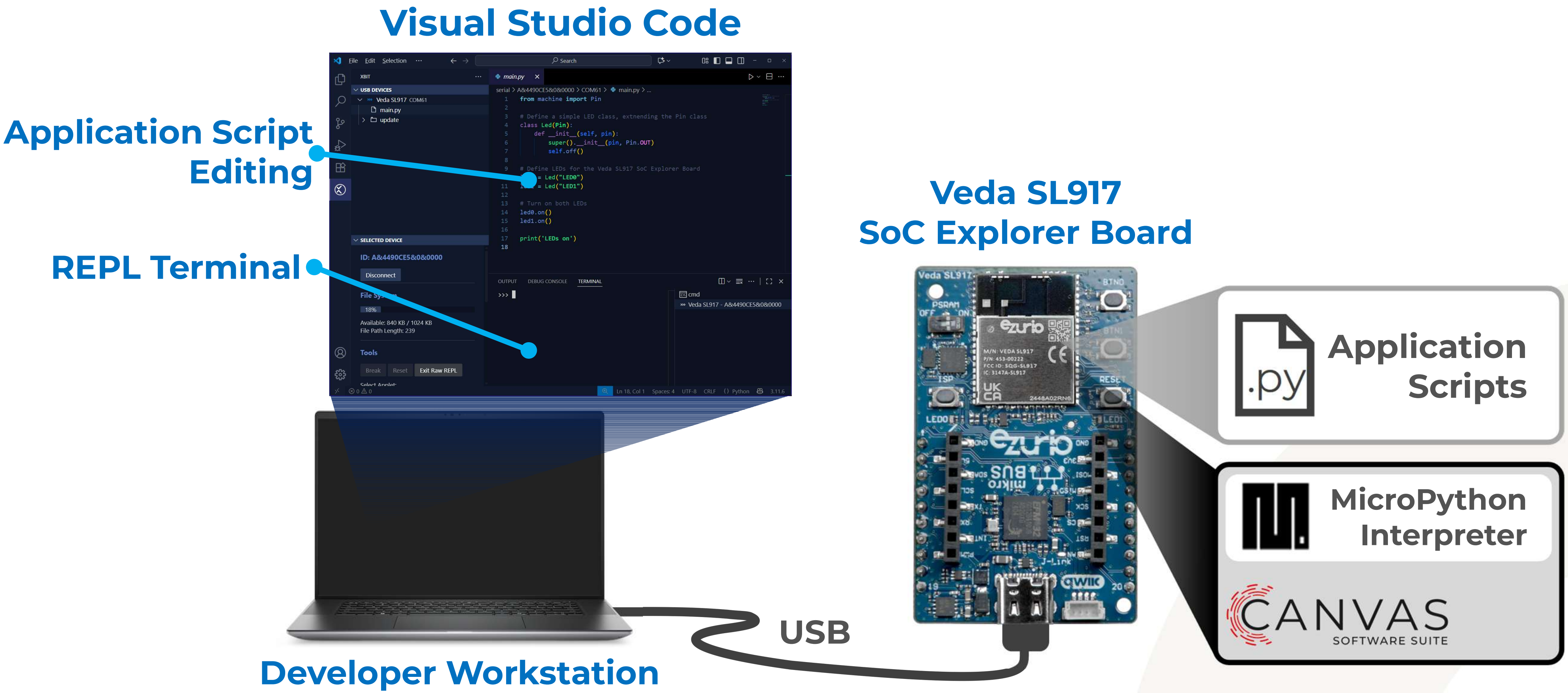
Visualize data and interact with Canvas-enabled wireless products.

Applets demonstrate wireless features. Firmware and application update support.

Download on the App Store

GET IT ON Google Play

CANVAS DEVELOPMENT SETUP



serial > A&4490CE5&0&0000 > COM61 > main.py > ...

```
from machine import Pin

# Define a simple LED class, extnending the Pin class
class Led(Pin):
    def __init__(self, pin):
        super().__init__(pin, Pin.OUT)
        self.off()

# Define LEDs for the Veda SL917 SoC Explorer Board
       = Led("LED0")
led1 = Led("LED1")

# Turn on both LEDs
led0.on()
led1.on()

print('LEDs on')
```

ID: A&4490CE5&0&0000

File System

18%

Available: 840 KB / 1024 KB
File Path Length: 239

```
>>> 
```

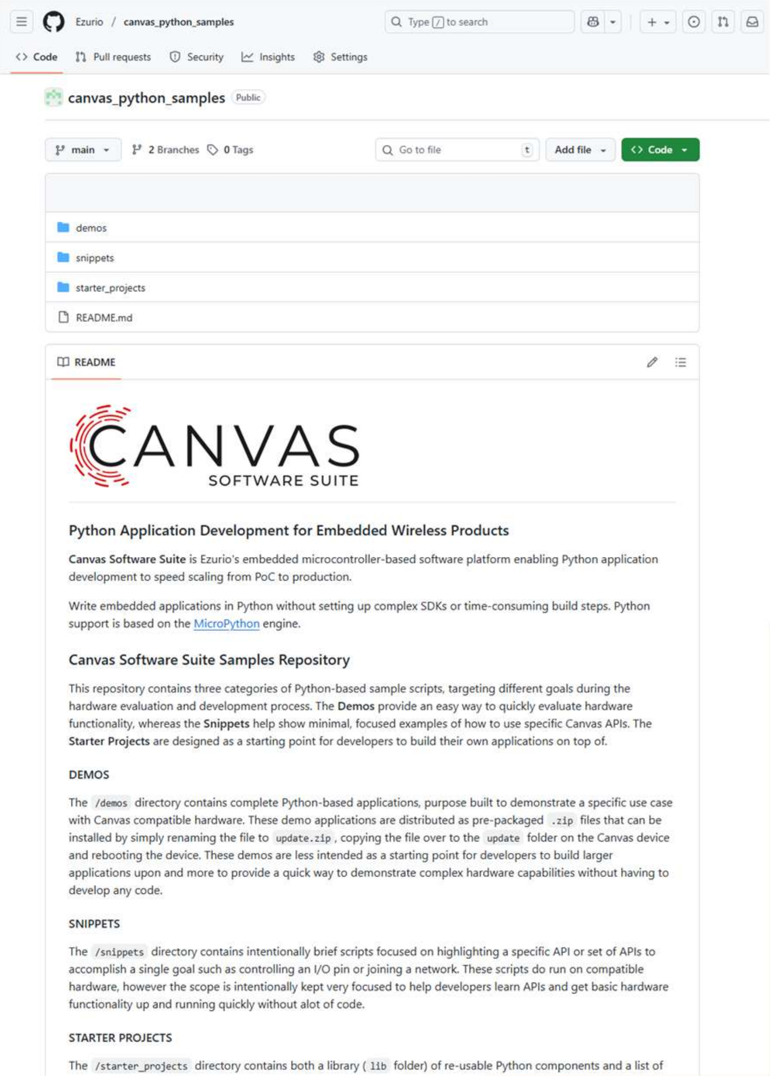
Veda SL917 - A&4490CE5&0&0000

CANVAS SOFTWARE SUITE - SAMPLES

Canvas Samples GitHub

Find snippets, demos and starter applications to learn the Canvas APIs.

Get started quickly with development kits and explore peripherals via mikroBUS and QWIIC interfaces.



Python Application Development for Embedded Wireless Products

Canvas Software Suite is Ezurio's embedded microcontroller-based software platform enabling Python application development to speed scaling from PoC to production.

Write embedded applications in Python without setting up complex SDKs or time-consuming build steps. Python support is based on the MicroPython engine.

Canvas Software Suite Samples Repository

This repository contains three categories of Python-based sample scripts, targeting different goals during the hardware evaluation and development process. The **Demos** provide an easy way to quickly evaluate hardware functionality, whereas the **Snippets** help show minimal, focused examples of how to use specific Canvas APIs. The **Starter Projects** are designed as a starting point for developers to build their own applications on top of.

DEMOS

The `/demos` directory contains complete Python-based applications, purpose built to demonstrate a specific use case with Canvas compatible hardware. These demo applications are distributed as pre-packaged `.zip` files that can be installed by simply renaming the file to `update.zip`, copying the file over to the `update` folder on the Canvas device and rebooting the device. These demos are less intended as a starting point for developers to build larger applications upon and more to provide a quick way to demonstrate complex hardware capabilities without having to develop any code.

SNIPPETS

The `/snippets` directory contains intentionally brief scripts focused on highlighting a specific API or set of APIs to accomplish a single goal such as controlling an I/O pin or joining a network. These scripts do run on compatible hardware, however the scope is intentionally kept very focused to help developers learn APIs and get basic hardware functionality up and running quickly without alot of code.

STARTER PROJECTS

The `/starter_projects` directory contains both a library (`lib` folder) of re-usable Python components and a list of

Veda SL917 Module

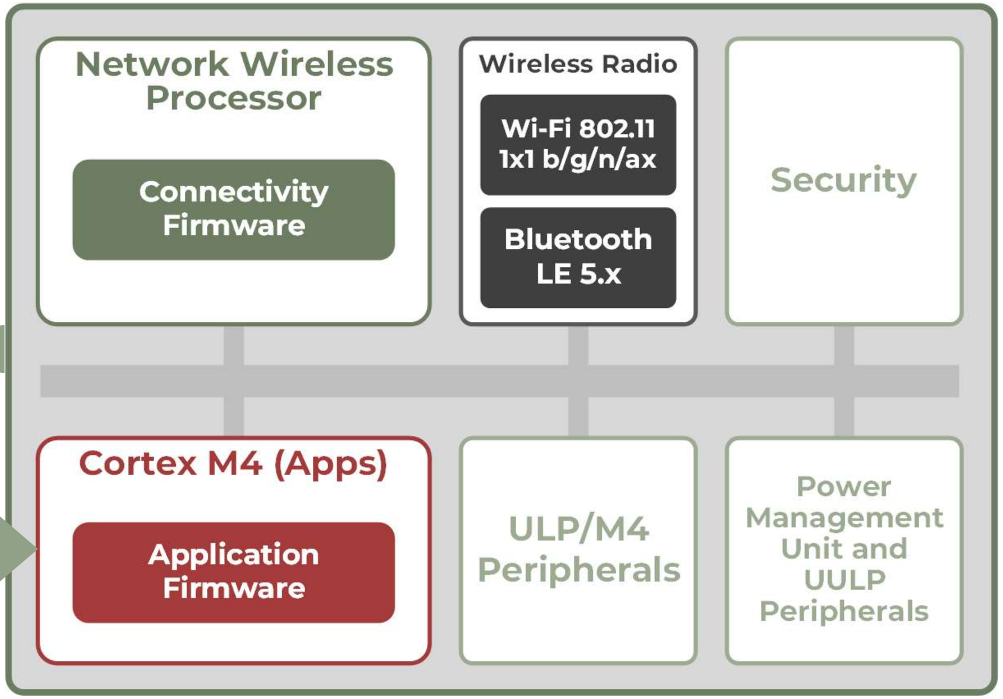
Hardware Overview

VEDA SL917 WI-FI 6 + BLUETOOTH MODULE

Veda SL917 Module

CANVAS
SOFTWARE SUITE

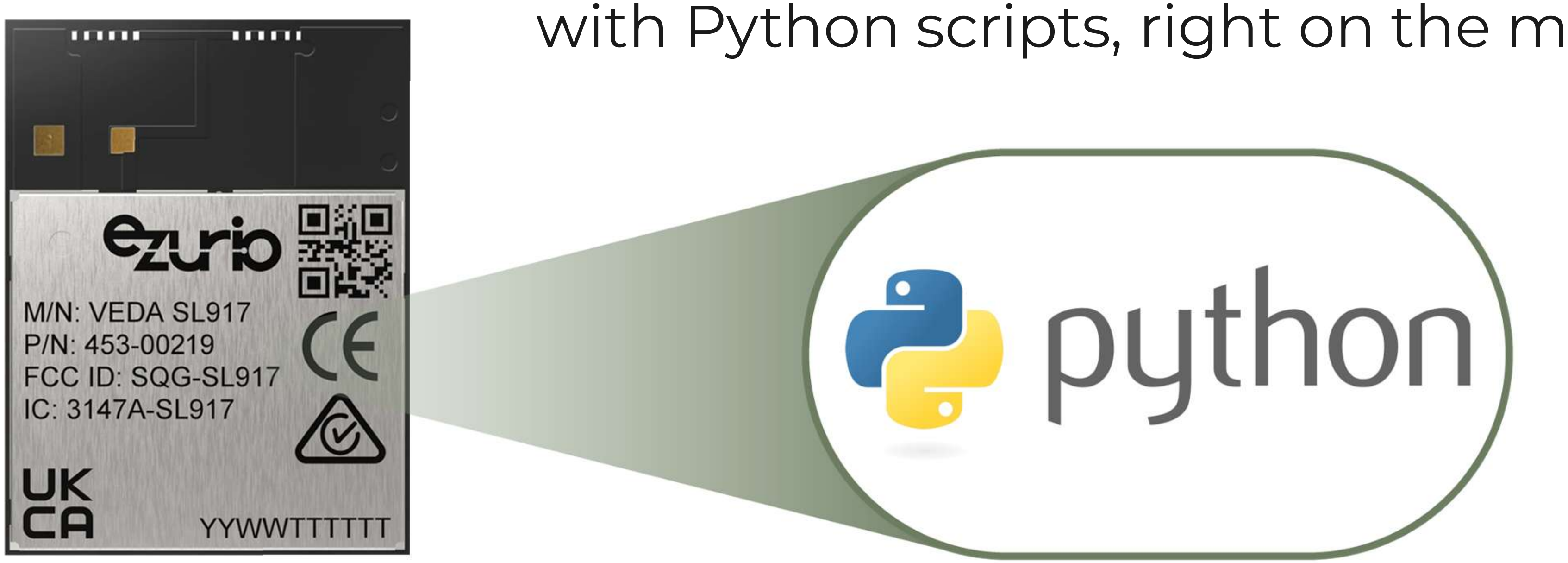
CANVAS FIRMWARE RUNS HERE



SoC Mode Architecture

DEVELOP WITH PYTHON

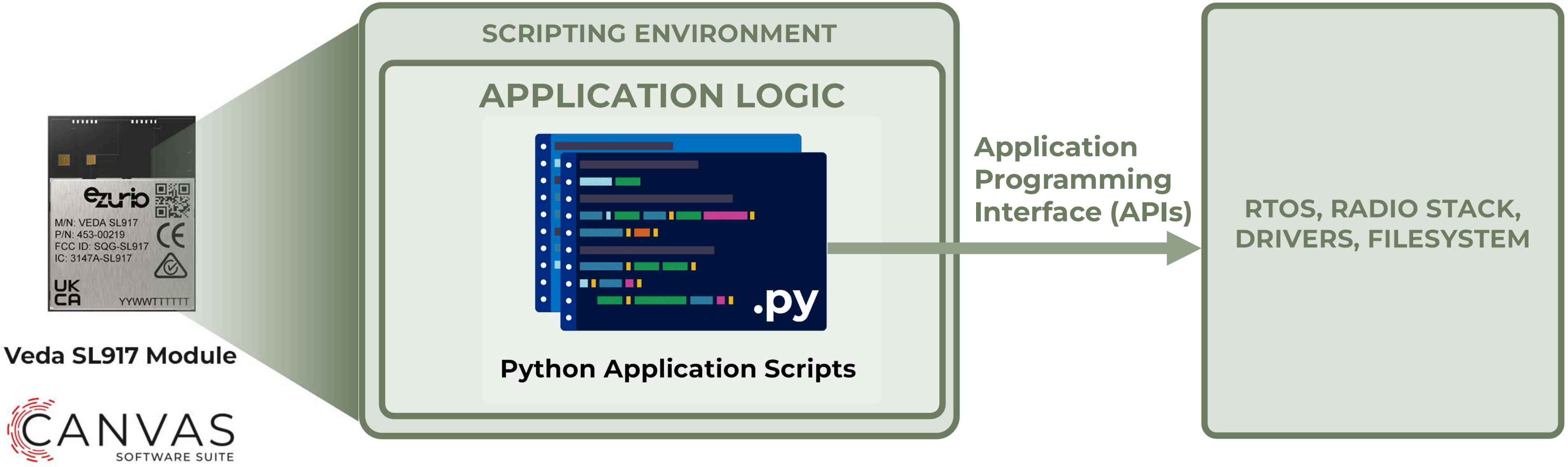
Develop Low Power Wireless Applications
with Python scripts, right on the module!



Veda SL917 Module

CANVAS
SOFTWARE SUITE

APPLICATIONS RUN IN A SCRIPTING ENVIRONMENT



Veda SL917 Module

CANVAS SOFTWARE SUITE

SCRIPTING ENVIRONMENT

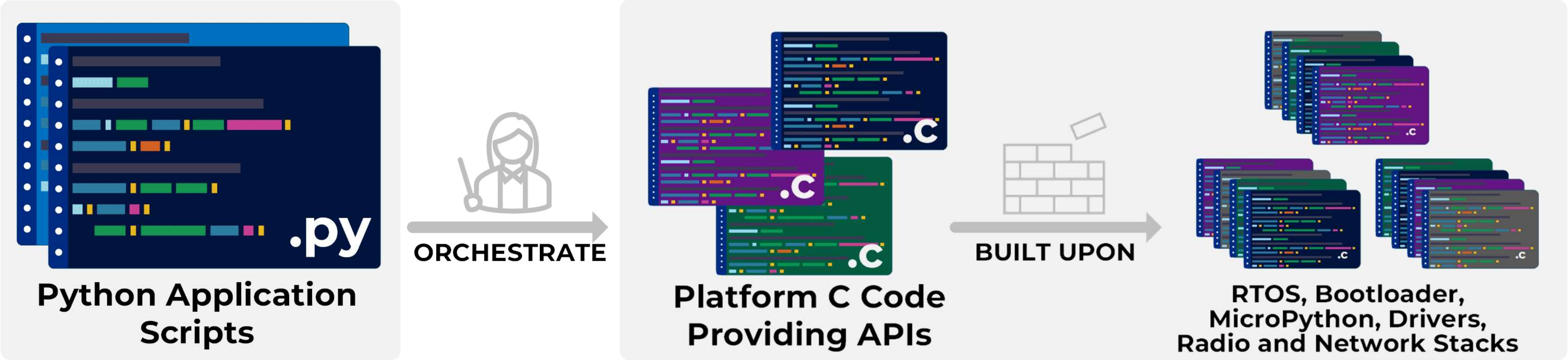
APPLICATION LOGIC

Python Application Scripts

Application Programming Interface (APIs)

RTOS, RADIO STACK, DRIVERS, FILESYSTEM

PYTHON APPLICATIONS ORCHESTRATE LOWER-LEVEL CODE



Python Application Scripts

ORCHESTRATE

Platform C Code Providing APIs

BUILT UPON

RTOS, Bootloader, MicroPython, Drivers, Radio and Network Stacks

APPLICATION SPECIFIC FEATURES, PYTHON SCRIPTS

LESS CODE TO DEVELOP AND MAINTAIN

MAINTAINED BY PRODUCT MANUFACTURER

UNIVERSAL / PLATFORM FEATURES VIA APIS

MAINTAINED BY EZURIO

HARDWARE-SPECIFIC CODE

SDK / LOW-LEVEL CODE REMAINS UNDER THE HOOD

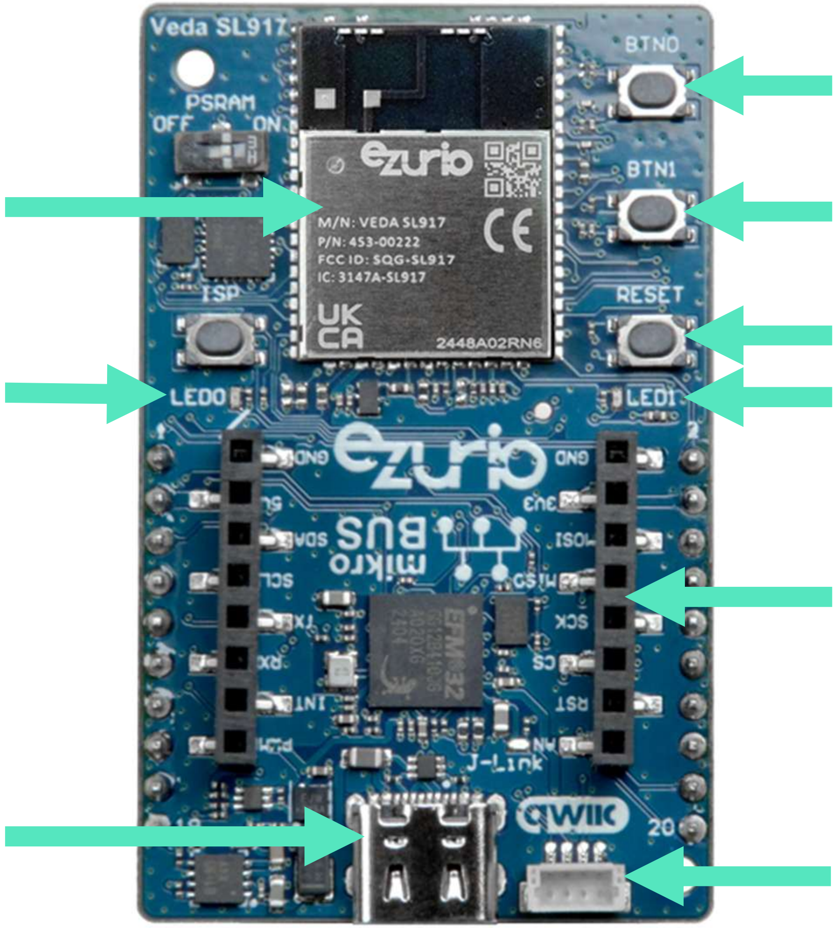
MAINTAINED BY CHIPSET MANUFACTURERS

CANVAS
SOFTWARE SUITE

Veda SL917 SoC Explorer Board

Hardware Overview

Veda SL917 SoC Explorer Board



Veda SL917 Module

LED0

USB-C Debug Interface

BTN0 Button

BTN1 Button

RESET Button

LED1

mikroBUS Socket

QWIIC Interface

Installing Canvas Firmware

Preparing for embedded application development

INSTALLING CANVAS FIRMWARE

1. INSTALL SIMPLICITY COMMANDER (PROGRAMMING TOOL)

2. DOWNLOAD CANVAS FIRMWARE IMAGE FILE(S)

3. FLASH THE FIRMWARE TO THE DEVELOPMENT BOARD

4. INSTALL VISUAL STUDIO CODE AND XBIT EXTENSION

Creating a Python Application

Developing your first Python application with Canvas

RUNNING A PYTHON APPLICATION

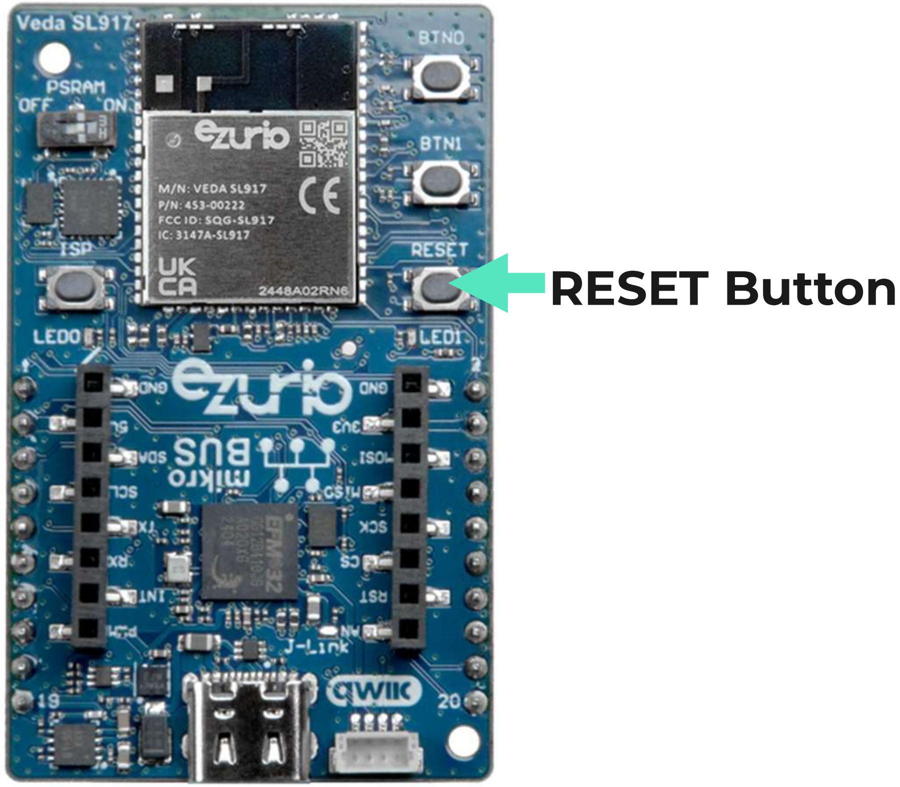
On startup, Canvas firmware will execute **boot.py** followed by **main.py** if they exist.

To start an application after saving a file as **boot.py** or **main.py**, press the RESET button.

TIPS

When a Python script is running, typing **Ctrl+C** into the REPL terminal will send a “break”, triggering the running script to stop.

Typing **Ctrl+D** at the REPL prompt (>>>) will trigger the module to RESET.



Python REPL Overview

Useful tips to get started with development

ACCESSING THE DEVICE PYTHON REPL

1 CLICK THE DEVICE IN THE USB DEVICE VIEW

- Clicking on the Canvas Device in the **USB DEVICES** panel opens the REPL terminal

2 NAVIGATE TO THE TERMINAL TAB

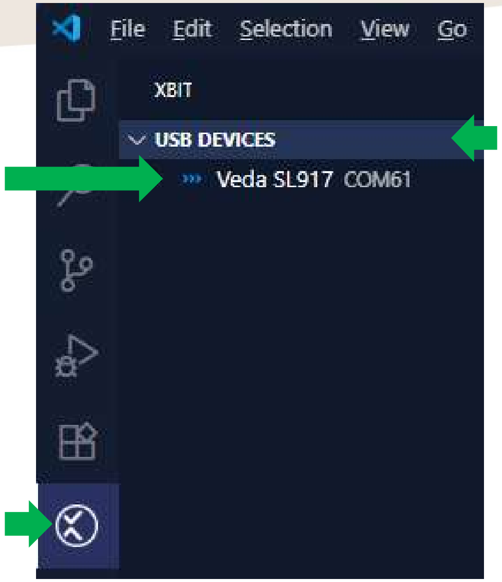
- Near the bottom of the window, click the **TERMINAL** tab

3 INTERACT WITH THE PYTHON REPL

- At the **>>>** prompt, Python commands can be entered and evaluated by the device one line at a time.

- This is useful for evaluating hardware APIs and testing simple operations without writing an entire script.

Visual Studio Code Window

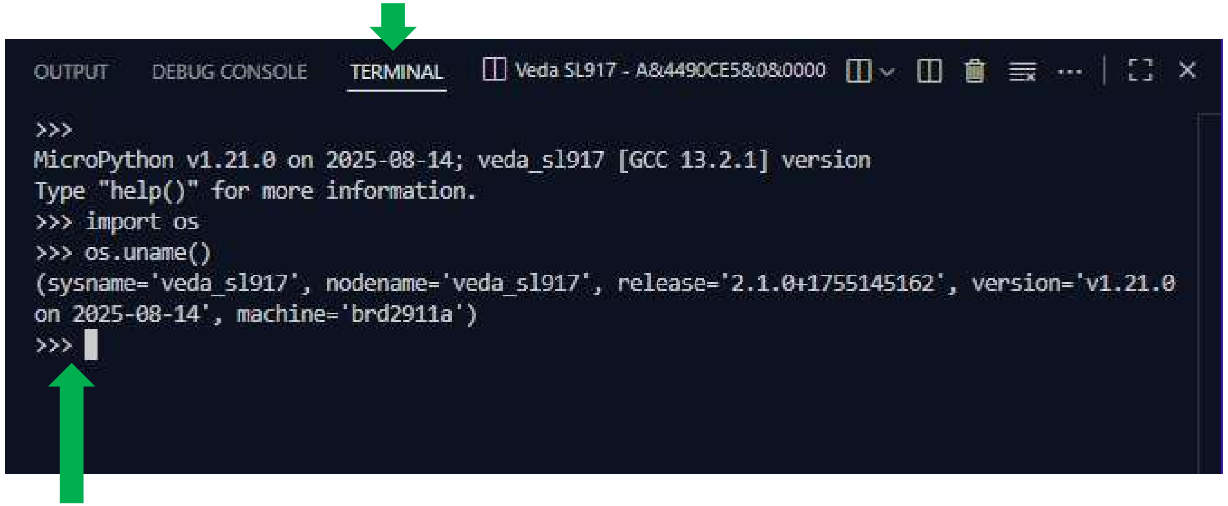


Canvas Device

Xbit USB DEVICES Panel

Xbit VS Code Extension

TERMINAL tab



```
>>>
MicroPython v1.21.0 on 2025-08-14; veda_sl917 [GCC 13.2.1] version
Type "help()" for more information.
>>> import os
>>> os.uname()
(sysname='veda_sl917', nodename='veda_sl917', release='2.1.0+1755145162', version='v1.21.0
on 2025-08-14', machine='brd2911a')
>>> 
```

REPL Prompt

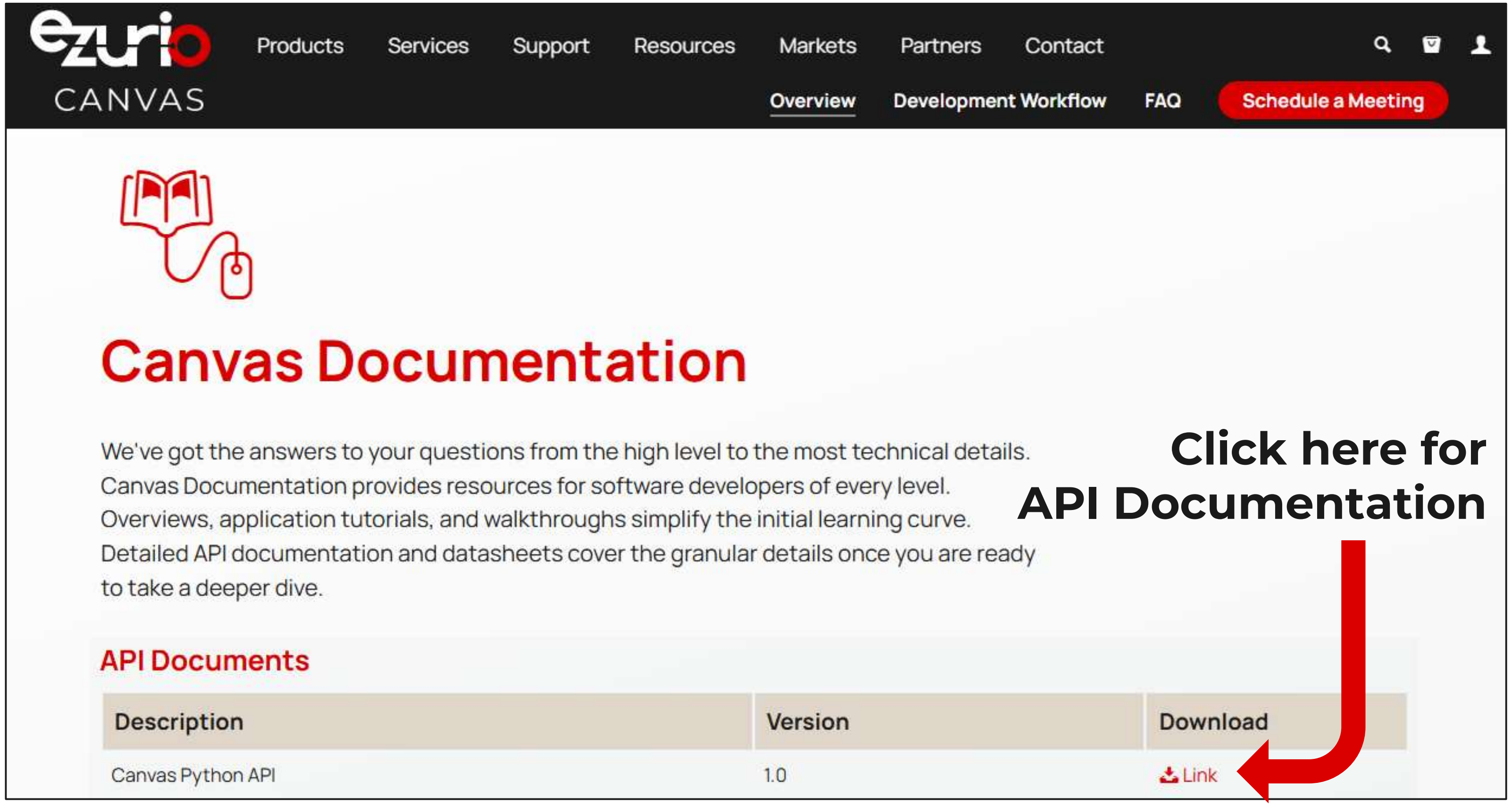
Canvas APIs and Samples

Highlighting Canvas API documentation and Python sample applications

LEARNING THE CANVAS PYTHON API

CANVAS PYTHON API DOCUMENTATION

ezurio.com/canvas/software-suite#docs



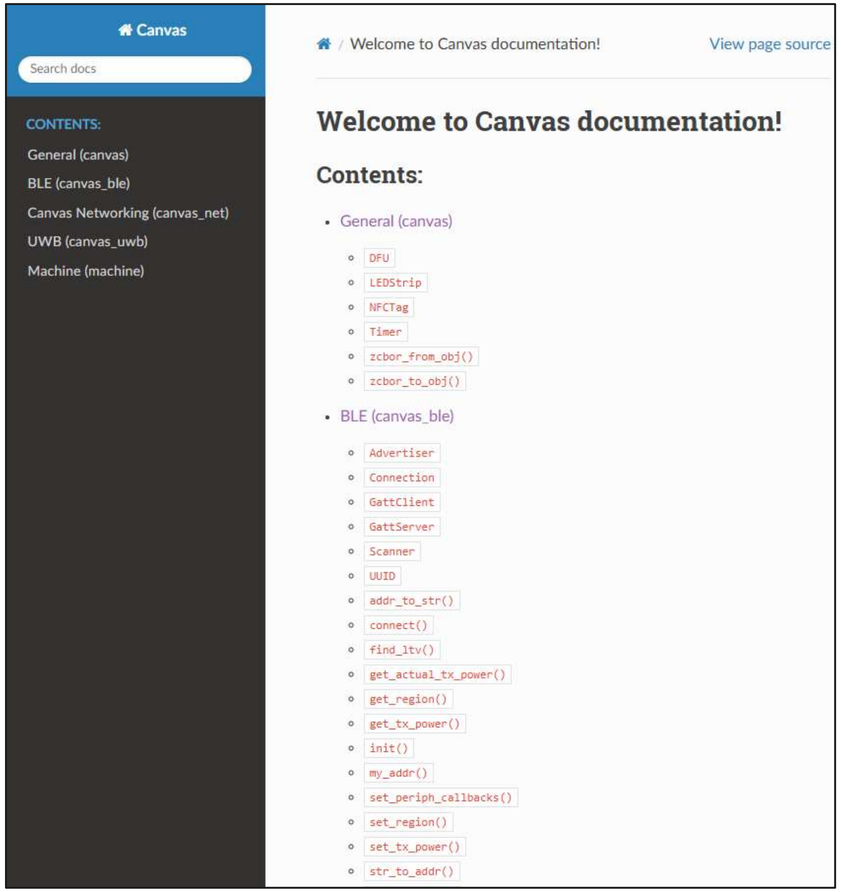
Canvas Documentation

We've got the answers to your questions from the high level to the most technical details. Canvas Documentation provides resources for software developers of every level. Overviews, application tutorials, and walkthroughs simplify the initial learning curve. Detailed API documentation and datasheets cover the granular details once you are ready to take a deeper dive.

API Documents

Description	Version	Download
Canvas Python API	1.0	Link

Click here for API Documentation



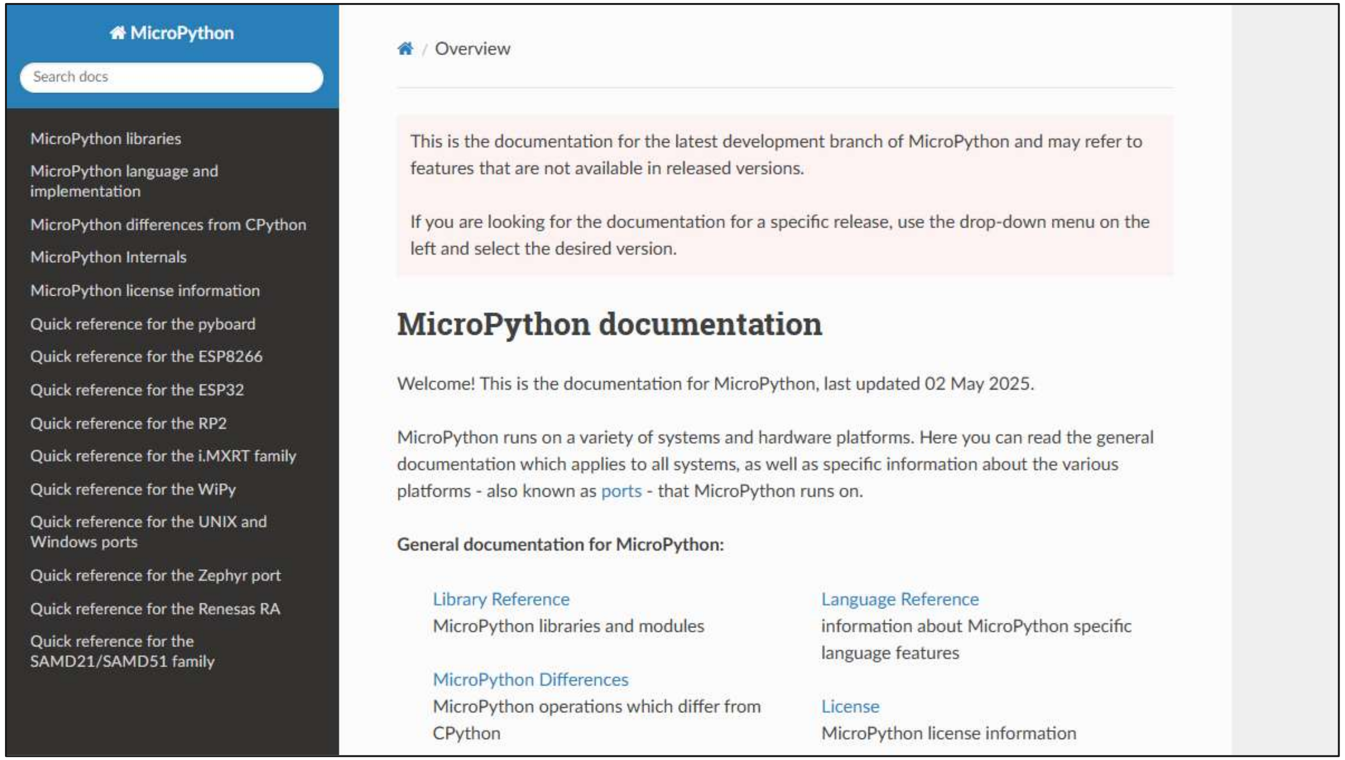
Welcome to Canvas documentation!

Contents:

- General (canvas)
  - `DFU`
  - `LEDStrip`
  - `NFCTag`
  - `Timer`
  - `zcbor_from_obj()`
  - `zcbor_to_obj()`
- BLE (canvas_ble)
  - `Advertiser`
  - `Connection`
  - `GattClient`
  - `GattServer`
  - `Scanner`
  - `UUID`
  - `addr_to_str()`
  - `connect()`
  - `find_ltv()`
  - `get_actual_tx_power()`
  - `get_region()`
  - `get_tx_power()`
  - `init()`
  - `my_addr()`
  - `set_periph_callbacks()`
  - `set_region()`
  - `set_tx_power()`
  - `str_to_addr()`

MICROPYTHON API DOCUMENTATION

docs.micropython.org/en/latest/



MicroPython libraries
MicroPython language and implementation
MicroPython differences from CPython
MicroPython Internals
MicroPython license information
Quick reference for the pyboard
Quick reference for the ESP8266
Quick reference for the ESP32
Quick reference for the RP2
Quick reference for the i.MXRT family
Quick reference for the WiPy
Quick reference for the UNIX and Windows ports
Quick reference for the Zephyr port
Quick reference for the Renesas RA
Quick reference for the SAMD21/SAMD51 family

This is the documentation for the latest development branch of MicroPython and may refer to features that are not available in released versions.

If you are looking for the documentation for a specific release, use the drop-down menu on the left and select the desired version.

MicroPython documentation

Welcome! This is the documentation for MicroPython, last updated 02 May 2025.

MicroPython runs on a variety of systems and hardware platforms. Here you can read the general documentation which applies to all systems, as well as specific information about the various platforms - also known as ports - that MicroPython runs on.

General documentation for MicroPython:

Library Reference
MicroPython libraries and modules

Language Reference
information about MicroPython specific language features

MicroPython Differences
MicroPython operations which differ from CPython

License
MicroPython license information

We recommend reviewing the MicroPython APIs many of which are available from the Canvas Python environment.

Built-in Module Overview

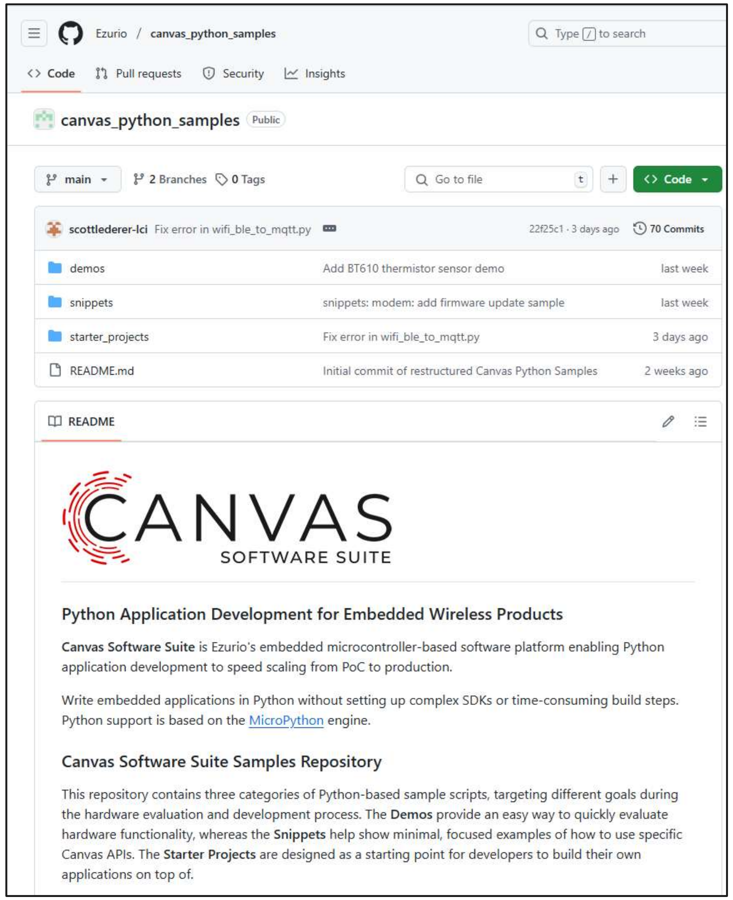
Here is a list of commonly used Python modules available with Canvas:

Python Feature	What it is useful for...
os	Filesystem operations and uname() function for version info
gc	Access garbage collection functions, check heap memory allocation
bytes()	Convenient data type for dealing with binary data
struct	Pack and Unpack data to/from bytes objects
binascii	Utility function for conversion to/from hexadecimal string format
json	Functions to convert JSON to/from Python dictionary objects
canvas.zcbor_from_obj()	Convert an object to a CBOR bytes object
canvas.zcbor_to_obj()	Convert a CBOR bytes object to a dictionary object
canvas.*	General Canvas platform functions
canvas_ble.*	Bluetooth Low Energy related classes
canvas_net.*	Networking related classes
canvas_uwb.*	Ultra-wide band two-way-ranging related classes
machine.*	Hardware-specific features, peripheral access, etc.

CANVAS SAMPLE APPLICATIONS

DISCOVER CANVAS PYTHON SAMPLE APPLICATIONS

github.com/ezurio/canvas_python_samples



canvas_python_samples

scottlederer-lci Fix error in wifi_ble_to_mqtt.py 22f25c1 · 3 days ago 70 Commits

demos	Add BT610 thermistor sensor demo	last week
snippets	snippets: modem: add firmware update sample	last week
starter_projects	Fix error in wifi_ble_to_mqtt.py	3 days ago
README.md	Initial commit of restructured Canvas Python Samples	2 weeks ago

CANVAS SOFTWARE SUITE

Python Application Development for Embedded Wireless Products

Canvas Software Suite is Ezurio's embedded microcontroller-based software platform enabling Python application development to speed scaling from PoC to production.

Write embedded applications in Python without setting up complex SDKs or time-consuming build steps. Python support is based on the MicroPython engine.

Canvas Software Suite Samples Repository

This repository contains three categories of Python-based sample scripts, targeting different goals during the hardware evaluation and development process. The **Demos** provide an easy way to quickly evaluate hardware functionality, whereas the **Snippets** help show minimal, focused examples of how to use specific Canvas APIs. The **Starter Projects** are designed as a starting point for developers to build their own applications on top of.

The Canvas Samples GitHub repository provides example script code, categorized as follows:

Demos

Complete Python applications demonstrating a specific use case with Canvas compatible hardware platforms.

Snippets

Focused scripts highlighting how to perform a specific task. Useful for quickly getting up to speed with APIs.

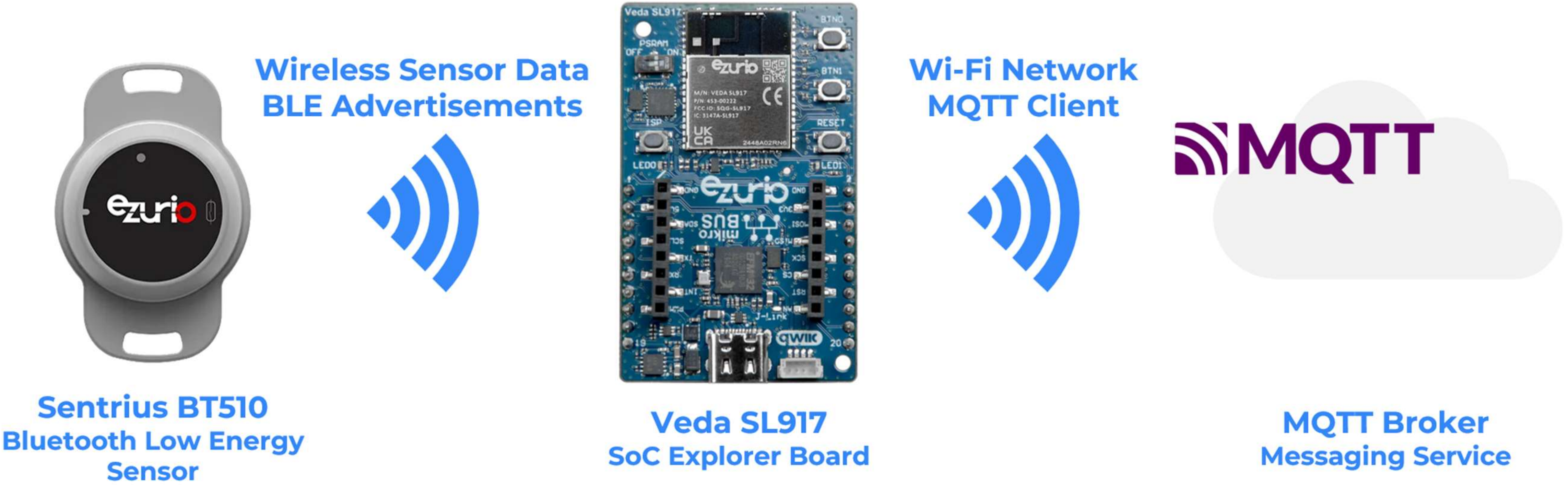
Starter Projects

Provides a library of re-usable modules and set of application project scripts that use them. These projects are a great starting point to build custom applications.

Wi-Fi, Bluetooth and Networking

Capturing BLE advertisements and reporting via MQTT over Wi-Fi

BLE TO WI-FI GATEWAY PROTOTYPE



Wireless Sensor Data
BLE Advertisements

Wi-Fi Network
MQTT Client

Sentrius BT510
Bluetooth Low Energy Sensor

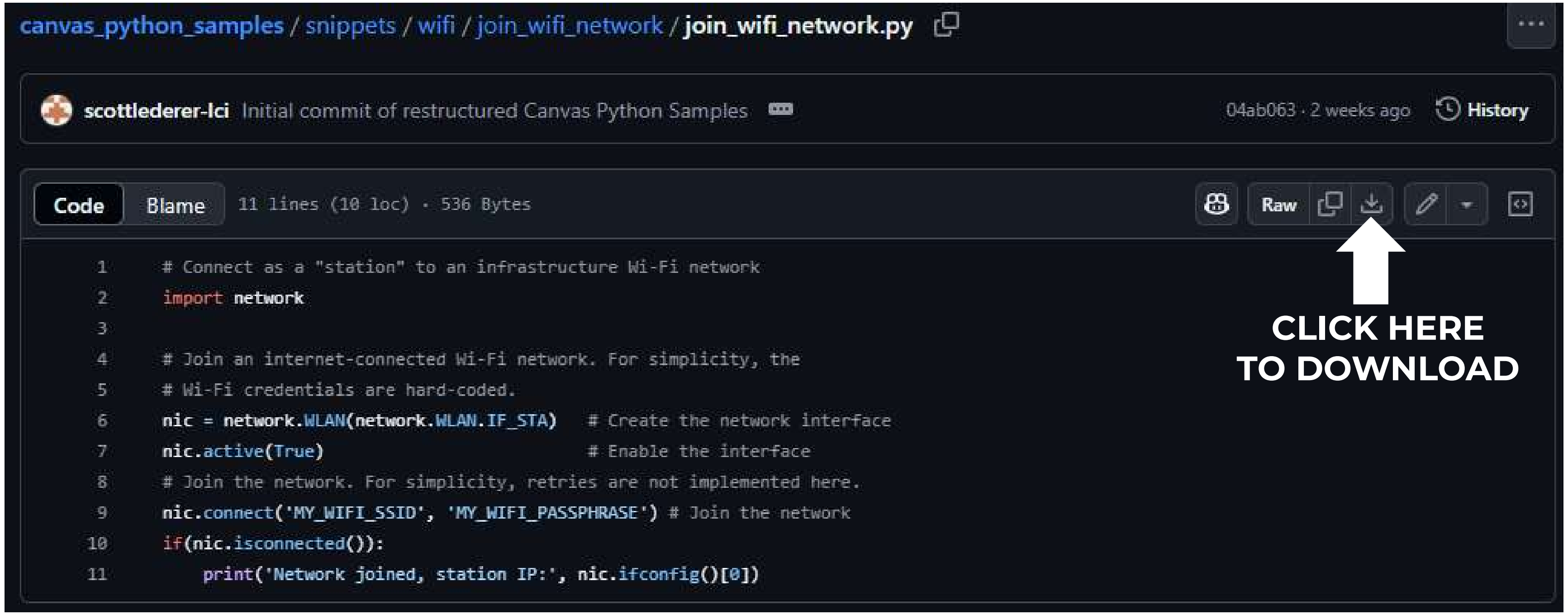
Veda SL917
SoC Explorer Board

MQTT Broker
Messaging Service

THE join_wifi_network SNIPPET

Canvas provides the **join_wifi_network** snippet showing a minimal script capable of joining a nearby Wi-Fi network.

Navigate to **snippets/wifi/join_wifi_network** to find **join_wifi_network.py**


canvas_python_samples / snippets / wifi / join_wifi_network / join_wifi_network.py
scottlederer-lci
04ab063 · 2 weeks ago
History
Code
Blame
11 lines (10 loc) · 536 Bytes
Raw

Initial commit of restructured Canvas Python Samples

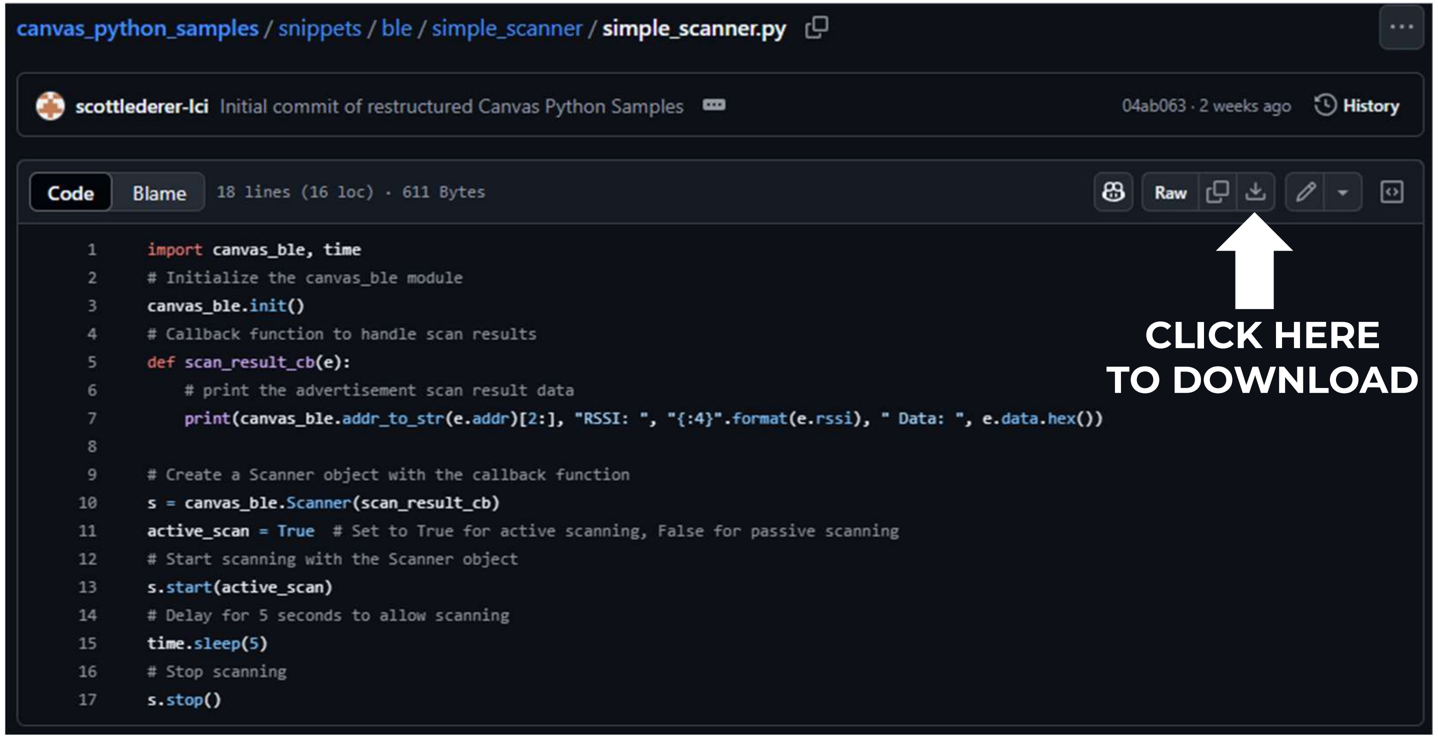
```python
# Connect as a "station" to an infrastructure Wi-Fi network
import network

# Join an internet-connected Wi-Fi network. For simplicity, the
# Wi-Fi credentials are hard-coded.
nic = network.WLAN(network.WLAN.IF_STA)  # Create the network interface
nic.active(True)                         # Enable the interface
# Join the network. For simplicity, retries are not implemented here.
nic.connect('MY_WIFI_SSID', 'MY_WIFI_PASSPHRASE') # Join the network
if(nic.isconnected()):
    print('Network joined, station IP:', nic.ifconfig()[0])
```

CLICK HERE TO DOWNLOAD

THE simple_scanner SNIPPET

Canvas provides the **simple_scanner** snippet to demonstrate using canvas_ble module to perform a Bluetooth Low Energy scan operation.

Navigate to **snippets/ble/simple_scanner** to find **simple_scanner.py**



canvas_python_samples / snippets / ble / simple_scanner / simple_scanner.py

scottlederer-lci Initial commit of restructured Canvas Python Samples

04ab063 · 2 weeks ago History

```python
import canvas_ble, time
# Initialize the canvas_ble module
canvas_ble.init()
# Callback function to handle scan results
def scan_result_cb(e):
    # print the advertisement scan result data
    print(canvas_ble.addr_to_str(e.addr)[2:], "RSSI: ", "{:4}".format(e.rssi), " Data: ", e.data.hex())

# Create a Scanner object with the callback function
s = canvas_ble.Scanner(scan_result_cb)
active_scan = True  # Set to True for active scanning, False for passive scanning
# Start scanning with the Scanner object
s.start(active_scan)
# Delay for 5 seconds to allow scanning
time.sleep(5)
# Stop scanning
s.stop()
```

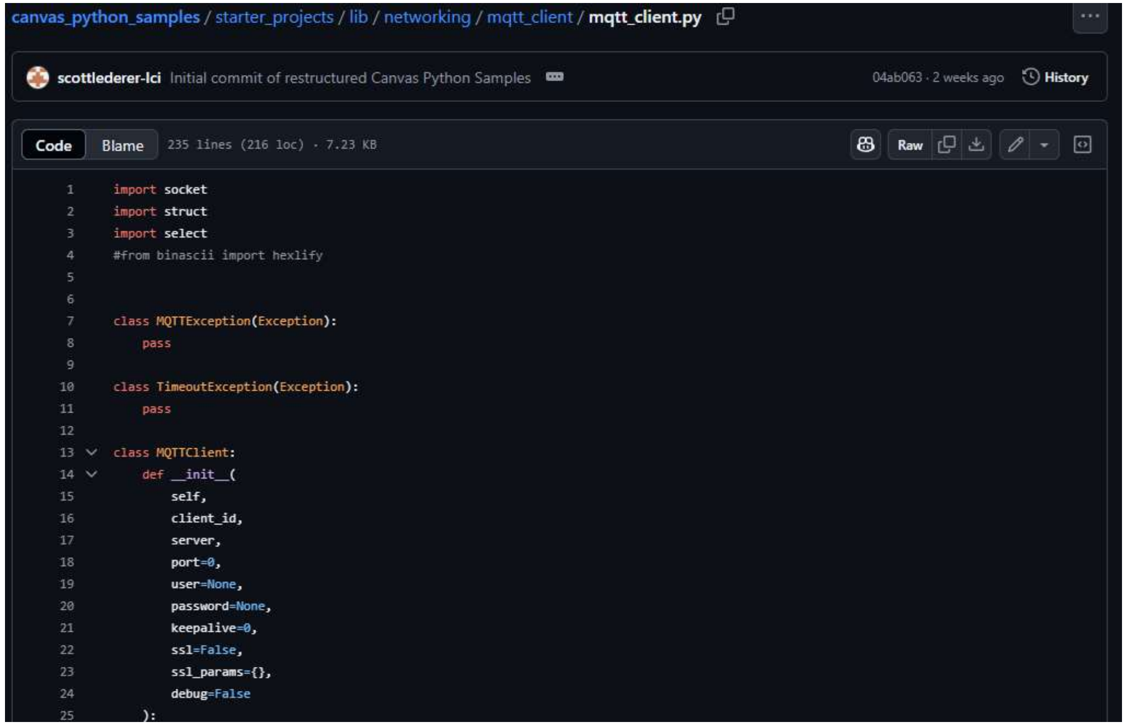
CLICK HERE
TO DOWNLOAD

THE MQTTClient CLASS

Use the **MQTTClient** class for connecting an MQTT broker

github.com/Ezurio/canvas_python_samples

Navigate to **starter_projects/lib/networking/mqtt_client**



canvas_python_samples / starter_projects / lib / networking / mqtt_client / **mqtt_client.py**

scottlederer-lci Initial commit of restructured Canvas Python Samples 04ab063 · 2 weeks ago **History**

235 lines (216 loc) · 7.23 KB

```
import socket
import struct
import select
#from binascii import hexlify


class MQTTException(Exception):
    pass

class TimeoutException(Exception):
    pass

class MQTTClient:
    def __init__(
        self,
        client_id,
        server,
        port=0,
        user=None,
        password=None,
        keepalive=0,
        ssl=False,
        ssl_params={},
        debug=False
    ):
```

Once connected to a Wi-Fi network, the **MQTTClient** class makes it easy to connect to a remote MQTT broker.

Publish data to a topic or subscribe and get a callback when new data is received.

MQTT is widely used by many IoT service providers. Using this class simplifies the integration process.

FURTHER APPLICATION ENHANCEMENTS

- Add network connection security using the **`ssl`** module
- BLE pairing and connection handling with the **`canvas_ble.Connection`** class
- Use the **`Config`** class from **`starter_projects/lib`** to save configuration settings
- Add BLE scan filters with the **`canvas_ble.Scanner.filter_add()`** function
- Add handlers for MQTT subscription events to act on received messages

All of these enhancements are possible with Python
using APIs provided by the Canvas platform!

AP Mode Provisioning + Wi-Fi Connectivity Demonstration

Exploring the capabilities of the Veda SL917

WI-FI DEVELOPER JOURNEY – COMMON REQUIREMENTS

Wi-Fi connected IoT products share many common requirements… → Canvas provides options to get a head start!

An approach to configuring Wi-Fi and Web Service details (Provisioning)

Serve a Web-based UI in AP mode to a mobile device, right from the module, to configure your application. Alternatively, use BLE GATT services and your own mobile app.

Implement a Web Service Client (Protocols)

Use provided SNTP, HTTP and MQTT clients or add your own protocol implementation to interact with your web services. Canvas socket and TLS APIs offer flexibility.

Interface with External Hardware (Peripherals)

Canvas provides API access to many common interfaces for control and data exchange with external hardware components. (e.g., SPI, I2C, GPIO, UART, ADC)

AP MODE PROVISIONING AND WI-FI CONNECTIVITY DEMO

AP Configuration Mode

Setup Wi-Fi from a mobile device web browser

Interactive Hardware Dashboard

Monitor button and LED status in real-time with a mobile device web browser

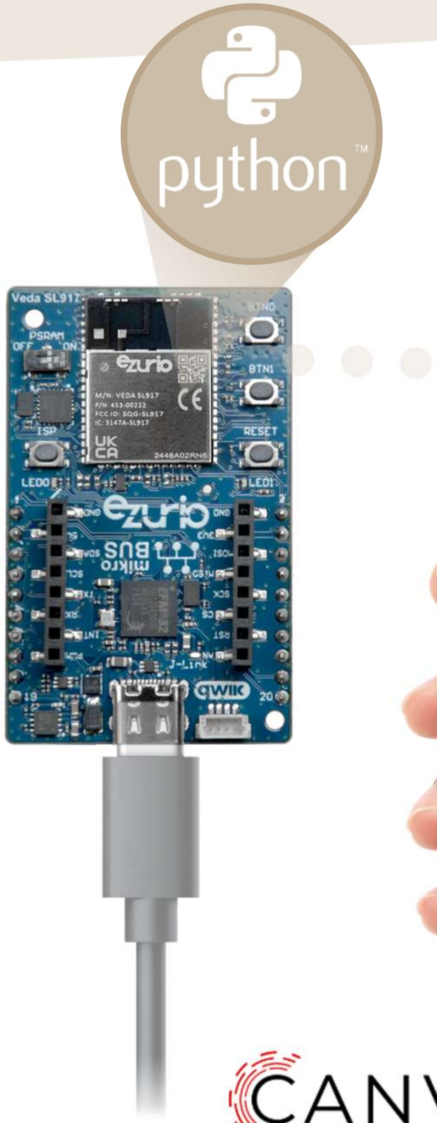
Webhook Demo

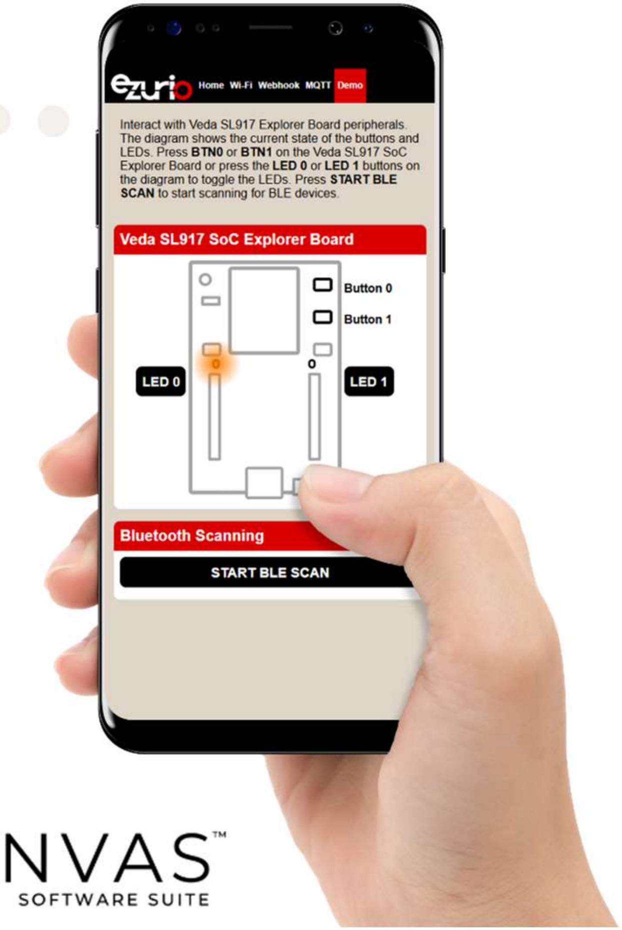
Trigger a Webhook from a button press

MQTT Demo

Connect an MQTT broker, publish status and subscribe for remote LED update commands

Try it yourself!
github.com/Ezurio/canvas_python_samples/tree/main/demos/ap_demo





Interact with Veda SL917 Explorer Board peripherals. The diagram shows the current state of the buttons and LEDs. Press **BTN0** or **BTN1** on the Veda SL917 SoC Explorer Board or press the **LED 0** or **LED 1** buttons on the diagram to toggle the LEDs. Press **START BLE SCAN** to start scanning for BLE devices.

Veda SL917 SoC Explorer Board

Bluetooth Scanning

START BLE SCAN

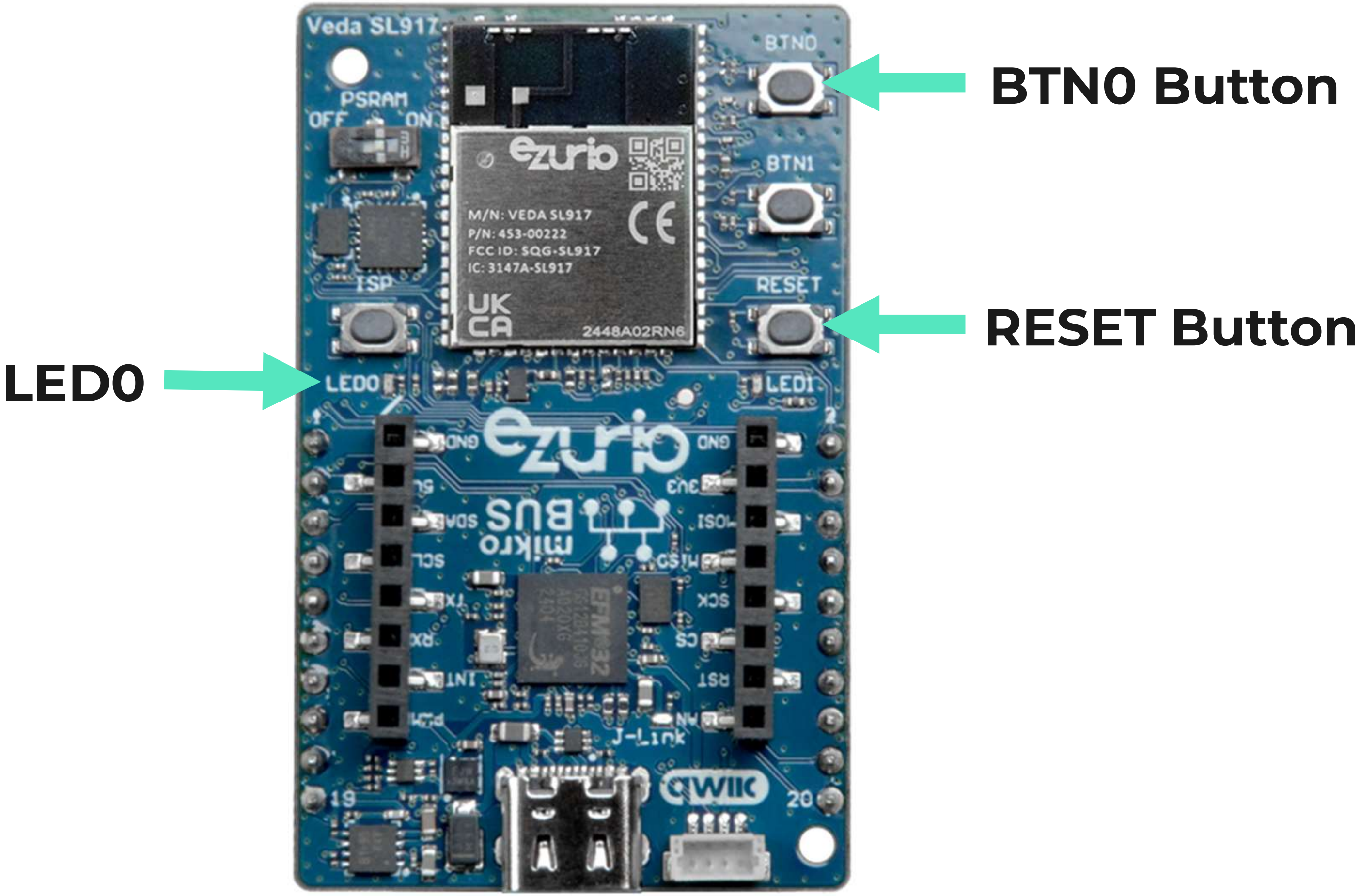
CANVAS™
SOFTWARE SUITE

Configuration Mode Web UI

Configuring Wi-Fi Network and Web Service Clients

ENTERING CONFIGURATION MODE

At any time, you can enter **configuration mode** by holding down BTN0 while pressing then releasing the RESET button and continuing to hold BTN0 until LED0 turns on.



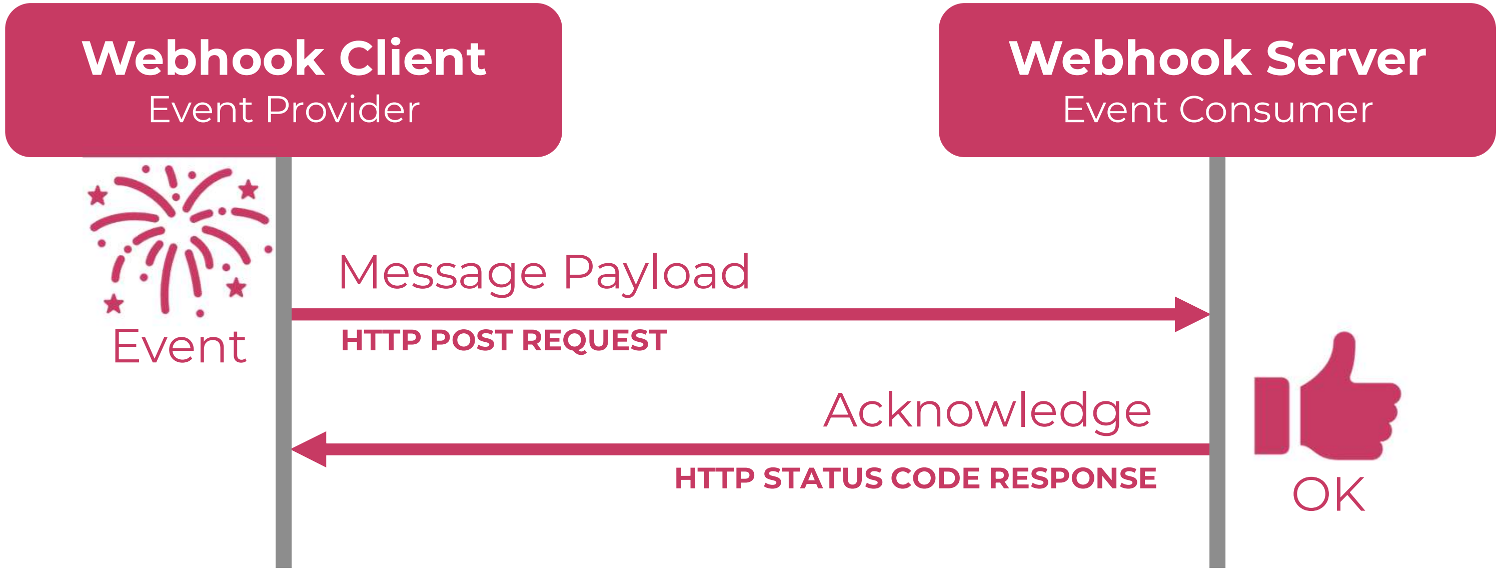
Webhook Demonstration

Report board status via HTTP webhook

WHAT IS A WEBHOOK?

webhooks Webhooks are automated **network messages** used to signal **events**

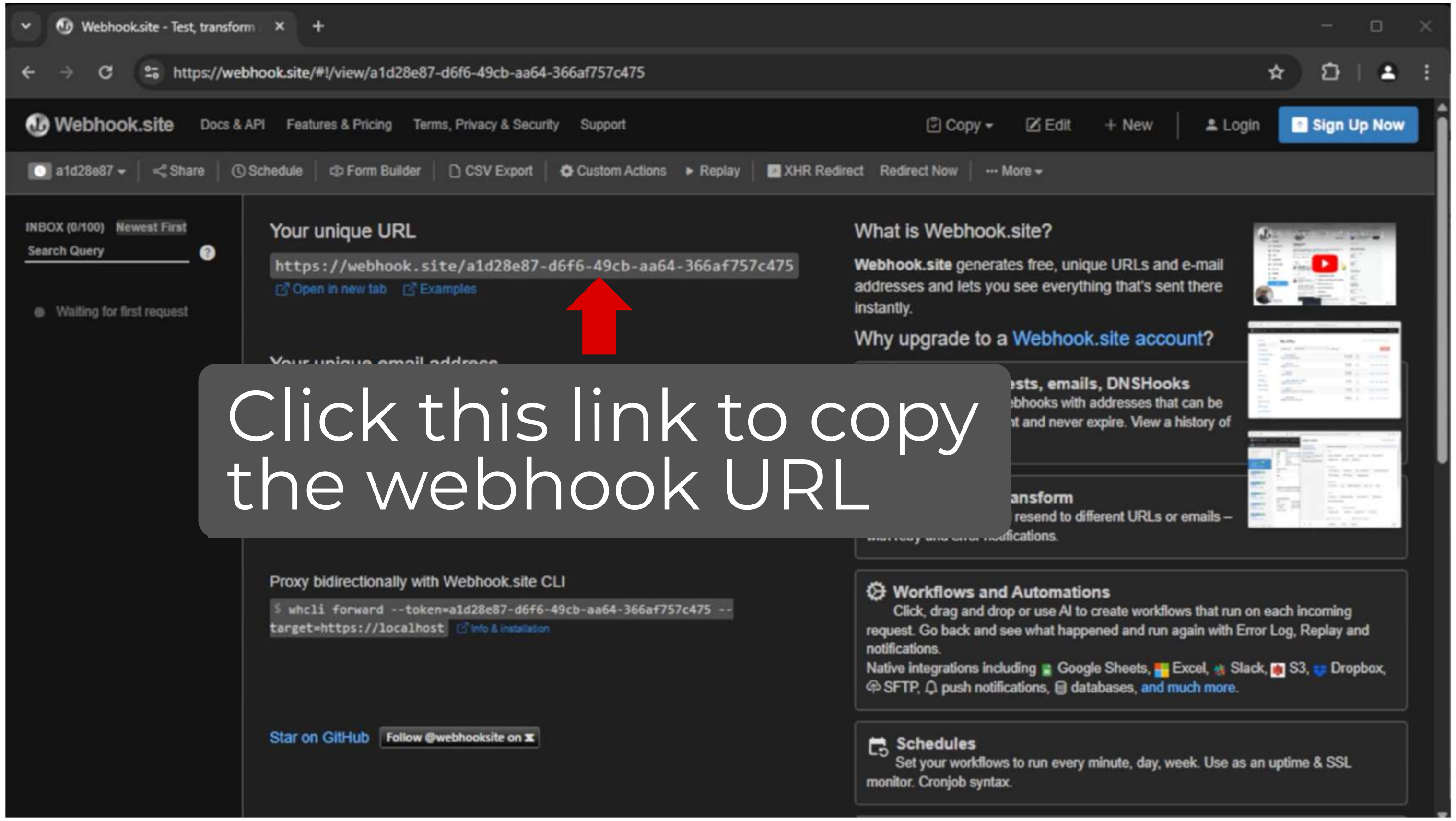
Webhooks utilize the **HTTP protocol** where **clients** POST messages to a **server**



Webhooks are defined by a URL where clients can send an HTTP message.
Webhook servers often only respond with HTTP status codes.

WEBHOOK.SITE

The webhook.site website offers a convenient webhook test interface



Your unique URL

https://webhook.site/a1d28e87-d6f6-49cb-aa64-366af757c475

Your unique email address

Proxy bidirectionally with Webhook.site CLI

```
$ whcli forward --token=a1d28e87-d6f6-49cb-aa64-366af757c475 --target=https://localhost
```

What is Webhook.site?

Webhook.site generates free, unique URLs and e-mail addresses and lets you see everything that's sent there instantly.

Why upgrade to a Webhook.site account?

ests, emails, DNSHooks

bhooks with addresses that can be t and never expire. View a history of

ansform

resend to different URLs or emails – ifications.

Workflows and Automations

Click, drag and drop or use AI to create workflows that run on each incoming request. Go back and see what happened and run again with Error Log, Replay and notifications.

Native integrations including Google Sheets, Excel, Slack, S3, Dropbox, SFTP, push notifications, databases, and much more.

Schedules

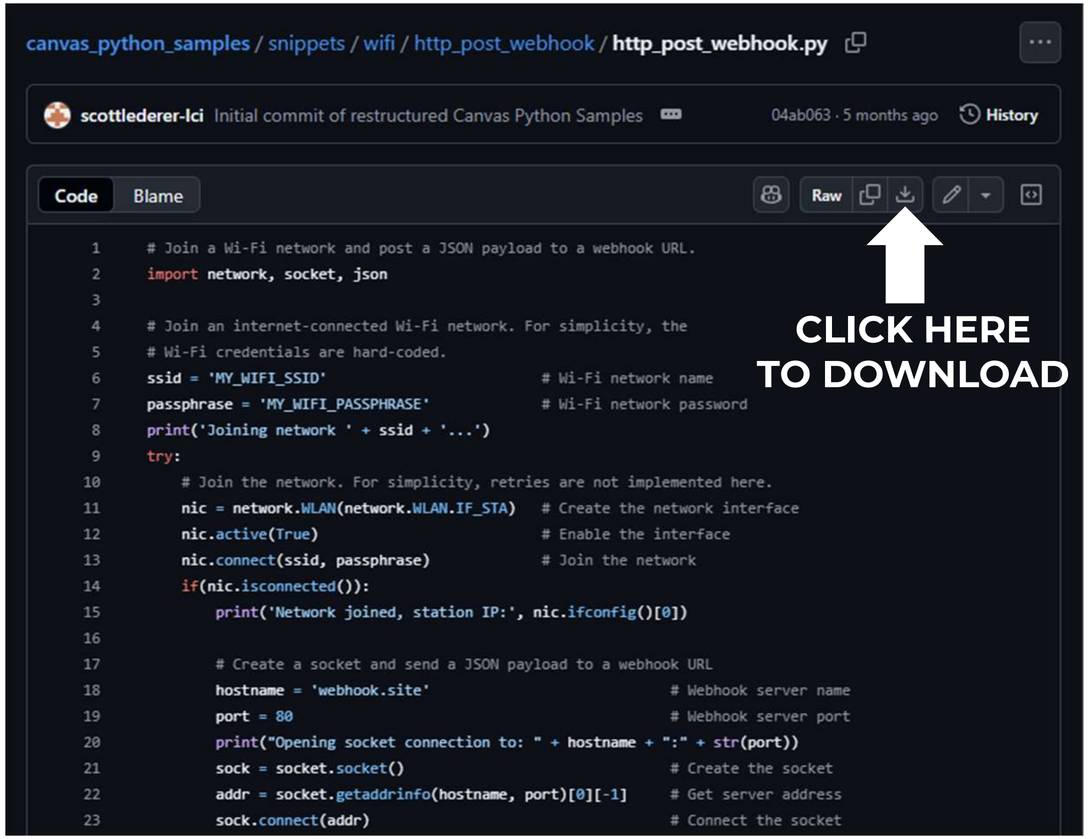
Set your workflows to run every minute, day, week. Use as an uptime & SSL monitor. Cronjob syntax.

Click this link to copy the webhook URL

THE http_post_webhook SNIPPET

Canvas provides the **http_post_webhook** snippet showing a minimal script to send an HTTP POST to a webhook URL.

Navigate to **snippets/wifi/join_wifi_network** to find **join_wifi_network.py**



canvas_python_samples / snippets / wifi / http_post_webhook / **http_post_webhook.py**

scottlederer-lci Initial commit of restructured Canvas Python Samples 04ab063 · 5 months ago History

```python
# Join a Wi-Fi network and post a JSON payload to a webhook URL.
import network, socket, json

# Join an internet-connected Wi-Fi network. For simplicity, the
# Wi-Fi credentials are hard-coded.
ssid = 'MY_WIFI_SSID'                         # Wi-Fi network name
passphrase = 'MY_WIFI_PASSPHRASE'             # Wi-Fi network password
print('Joining network ' + ssid + '...')
try:
    # Join the network. For simplicity, retries are not implemented here.
    nic = network.WLAN(network.WLAN.IF_STA)   # Create the network interface
    nic.active(True)                          # Enable the interface
    nic.connect(ssid, passphrase)             # Join the network
    if(nic.isconnected()):
        print('Network joined, station IP:', nic.ifconfig()[0])

        # Create a socket and send a JSON payload to a webhook URL
        hostname = 'webhook.site'                            # Webhook server name
        port = 80                                            # Webhook server port
        print("Opening socket connection to: " + hostname + ":" + str(port))
        sock = socket.socket()                               # Create the socket
        addr = socket.getaddrinfo(hostname, port)[0][-1]     # Get server address
        sock.connect(addr)                                   # Connect the socket
```

CLICK HERE TO DOWNLOAD

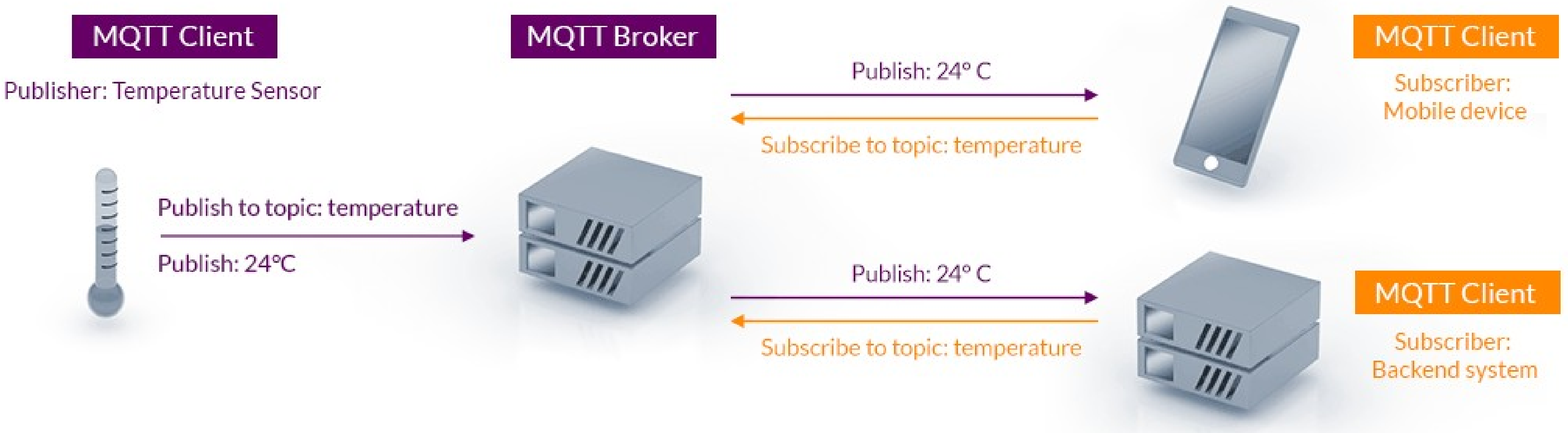
Monitoring and Control via MQTT

Connecting an MQTT broker accessible from a web app

WHAT IS MQTT?

MQTT is a **communications protocol** commonly used by IoT devices

MQTT operates with **MQTT Clients**, and a server called an **MQTT Broker**



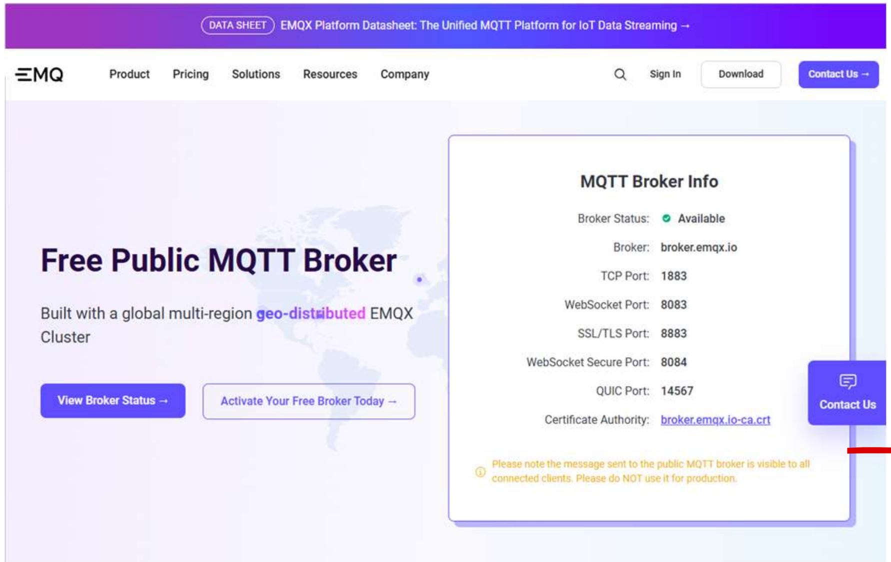
*Publish/Subscribe Architecture Diagram from **mqtt.org***

Clients **Publish** and **Subscribe** for messages on specified **Topics**.
The MQTT Broker keeps track of receiving and relaying messages from/to each topic.

THE EMQX MQTT BROKER

To use the EMQX MQTT broker for this demonstration:

visit: emqx.com/en/mqtt/public-mqtt5-broker



Free Public MQTT Broker

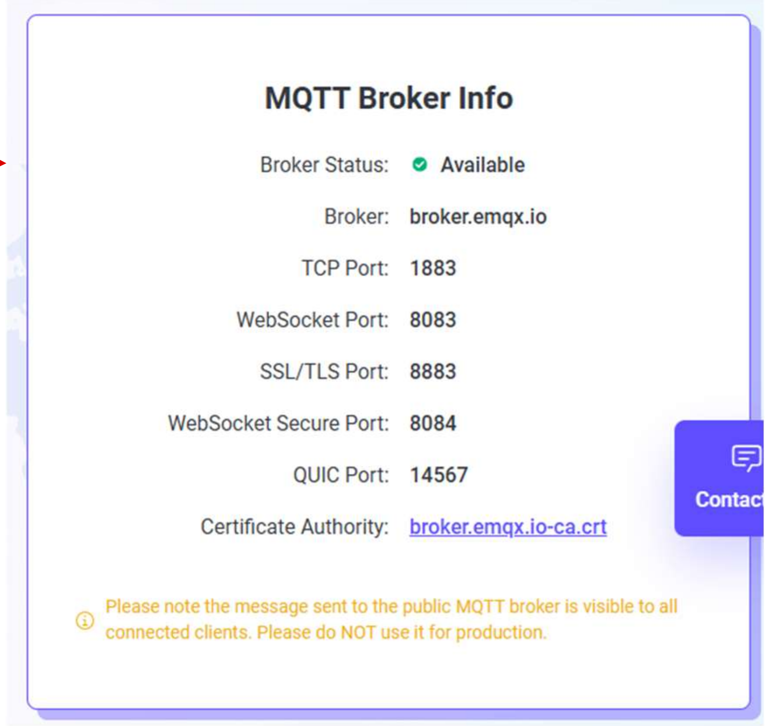
Built with a global multi-region geo-distributed EMQX Cluster

MQTT Broker Info

Broker Status: Available
Broker: broker.emqx.io
TCP Port: 1883
WebSocket Port: 8083
SSL/TLS Port: 8883
WebSocket Secure Port: 8084
QUIC Port: 14567
Certificate Authority: broker.emqx.io-ca.crt

Please note the message sent to the public MQTT broker is visible to all connected clients. Please do NOT use it for production.

emqx.com/en/mqtt/public-mqtt5-broker



MQTT Broker Info

Broker Status: Available
Broker: broker.emqx.io
TCP Port: 1883
WebSocket Port: 8083
SSL/TLS Port: 8883
WebSocket Secure Port: 8084
QUIC Port: 14567
Certificate Authority: broker.emqx.io-ca.crt

Please note the message sent to the public MQTT broker is visible to all connected clients. Please do NOT use it for production.

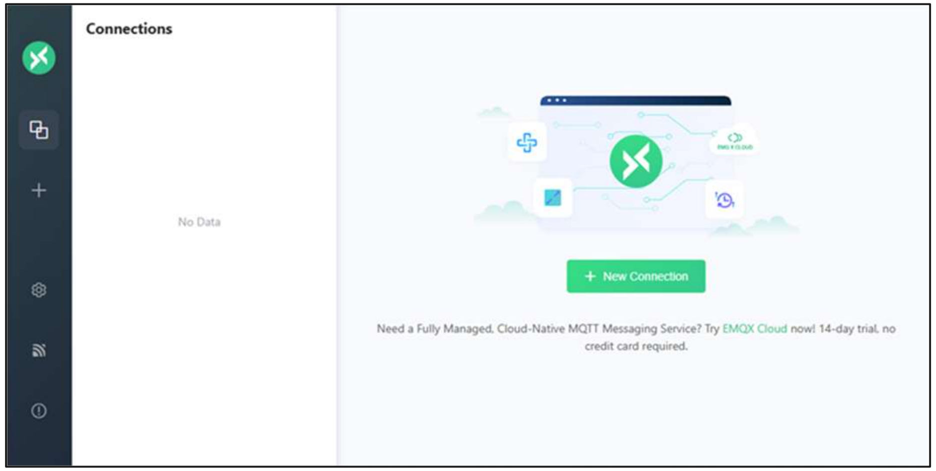
Press the **Connect** button to view connection details.

Take note of the **Broker** (Host) and **TCP Port** settings. These will need to be entered into the Veda SL917 using configuration mode.

THE MQTT X WEB CLIENT

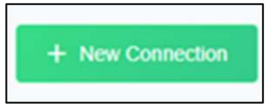
MQTT X offers a web-based MQTT client to publish/subscribe

visit: mqttx.app/web-client

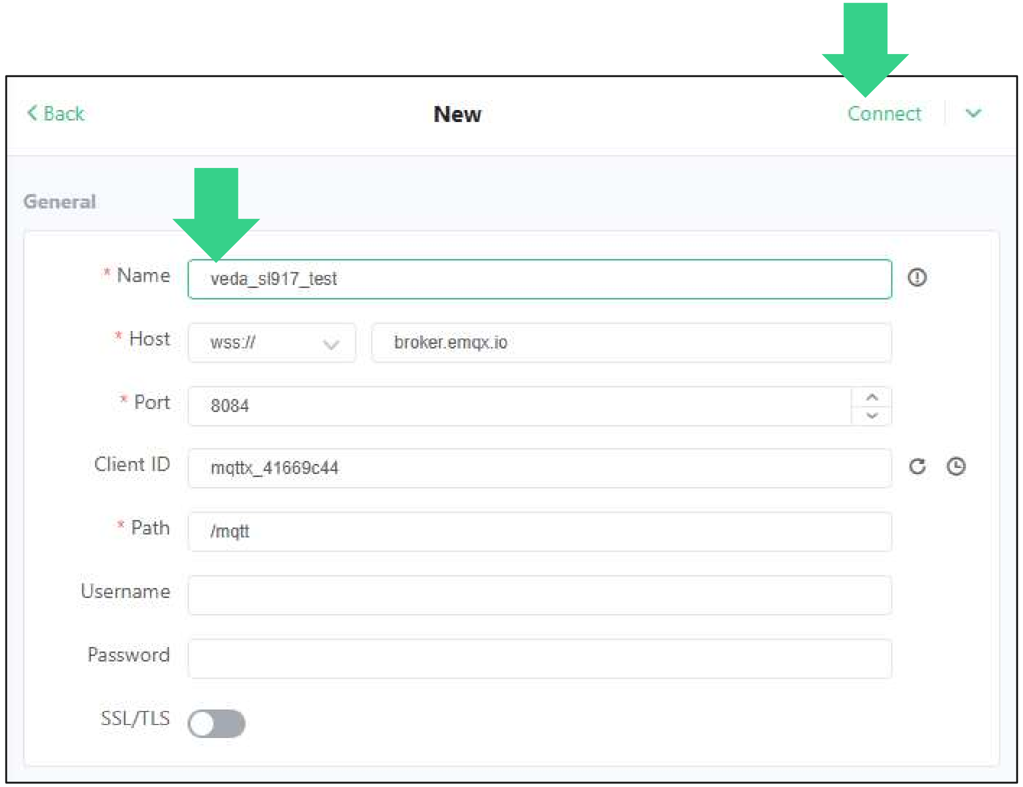


Need a Fully Managed, Cloud-Native MQTT Messaging Service? Try EMQX Cloud now! 14-day trial, no credit card required.

Web-based MQTT Client



Click the **New Connection** button



* Name	veda_sl917_test
* Host	wss:// broker.emqx.io
* Port	8084
Client ID	mqttx_41669c44
* Path	/mqtt
Username	
Password	
SSL/TLS	○

Enter a Name for the connection, then click **Connect**

REVIEWING THE DEMO APPLICTION

Canvas AP Mode Provisioning and Wi-Fi Connectivity Demo

Canvas Demo Application Features:

- Configuration Mode with on-module HTTP Server providing Web UI
- Connect to a Wi-Fi network
- Send board status to an HTTP webhook URL
- Connect with an MQTT broker
- Monitor board status by subscribing to an MQTT topic
- Control LEDs by sending RPC commands to an MQTT topic

REVIEWING CANVAS SOFTWARE SUITE - Veda SL917

Low-power Embedded Wi-Fi and BLE applications with Python

- **Veda SL917 SoC Explorer Board**
  - No additional cost for Canvas software

CANVAS SOFTWARE SUITE

- **Let Canvas Accelerate Your Next Design**
  - Develop with Python and get a head start with samples and powerful production-focused features
- **Try Canvas Firmware Today**
  - Canvas Firmware images available on GitHub
- **Contact Ezurio for Support**
  - Canvas experts can assist you with your design

Veda SL917
SoC Explorer Board
453-00222-K1

Available Now

Veda SL917
Module
453-00221
453-00222

ARROW MOUSER ELECTRONICS DigiKey

LINKS TO ONLINE RESOURCES

CANVAS FIRMWARE

https://github.com/Ezurio/canvas_python_firmware

CANVAS PYTHON SAMPLES

https://github.com/Ezurio/canvas_python_samples

SIMPLICITY COMMANDER

https://silabs.com/developer-tools/simplicity-studio/simplicity-commander

VISUAL STUDIO CODE

https://code.visualstudio.com

CANVAS PYTHON API DOCUMENTATION

https://ezurio.com/canvas/software-suite#docs