

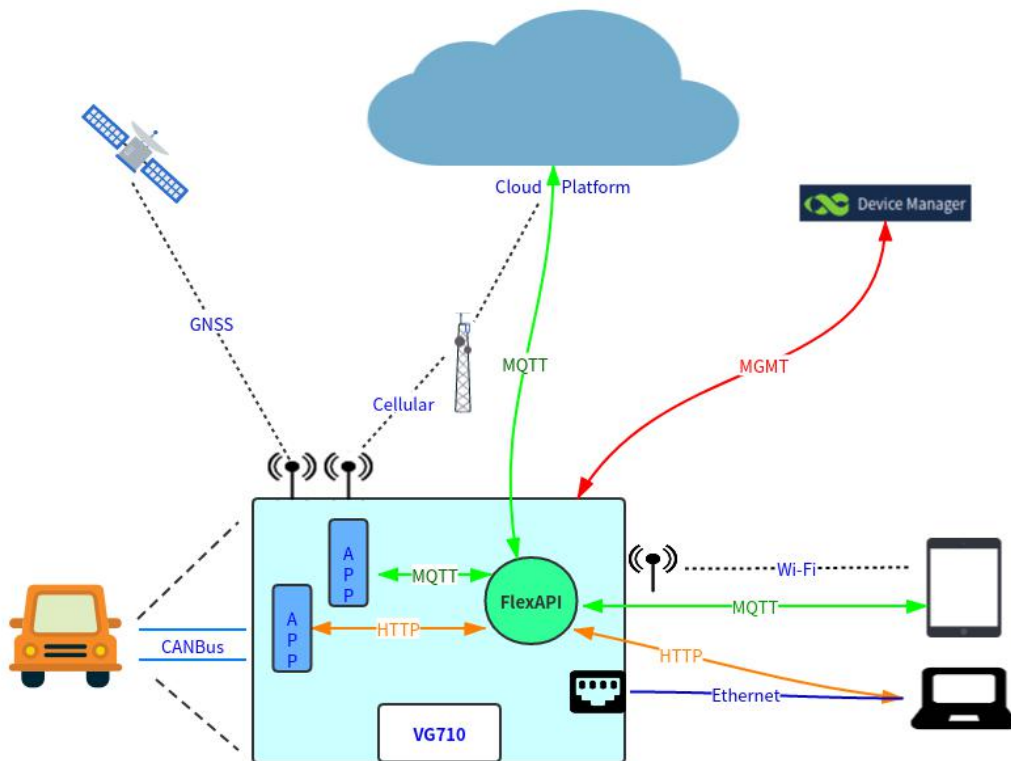
VG710 & FlexAPI Overview

Introduction

We introduced FlexAPI for the fast evolving IoT applications, which highly value easy integration, openness, flexibility, extensibility and programmability.

IoT is used intensively all over the world, smart things and even legacy devices are more and more connected to the cloud to make the world more digitalized and our lives easier. Besides, Edge computing is a rising power which features fast response, low latency and local intelligence. Cloud & Edge computing which complements each other come together could form a smarter solution and better world.

The following figure helps to understand how our powerful VG710 vehicle gateway plus its flexible FlexAPI could ease your application: **connect & serve**.



In the above figure, our VG710 at the central part physically connects to the vehicle, GNSS satellites, cellular base station, cloud platforms, Wi-Fi equipped tablet and Ethernet equipped industrial PC. Besides, service data stream flow from FlexAPI to cloud platform, local Python/Docker APPs and LAN device software.

VG710 overview

An ideal solution first needs powerful hardware features.
Our VG710 vehicle gateway has the following advantages:

1. Specially designed all in one solution

In a single box integrates:

Cellular LTE Advanced Cat6

Wi-Fi 5

Gigabit Ethernet

GNSS with Dead reckoning support

Accelerometer and gyroscope sensors

CANBus

Serial interface

IOs and Bluetooth

2. Powerful networking ability

High speed cellular WAN for always online & fast connection.

High speed WLAN for seamless wireless experience.

High speed Gigabit LAN for stable connection.

3. Edge computing ready

Processing power:

ARM Cortex A7 Quad-Core 717MHz

RAM: 512MB(optional 1GB)

eMMC storage: 8GB

Python & Docker support

4. Vehicle condition aware

Supports real-time GNSS location, motion detection, IO monitoring and OBD-II/J1939 data acquisition to know exactly what's going on.

5. Proven embedded firmware

Modular design, stable, efficient, high availability and auto fault recovery.

FlexAPI overview

An ideal solution further needs flexible software for application integration. Here comes the FlexAPI, which provides unified data and control services.

FlexAPI is designed to be Cloud and Edge friendly, you can use it for both your cloud platform integration and field Edge computing. It's network oriented and programming language independent.

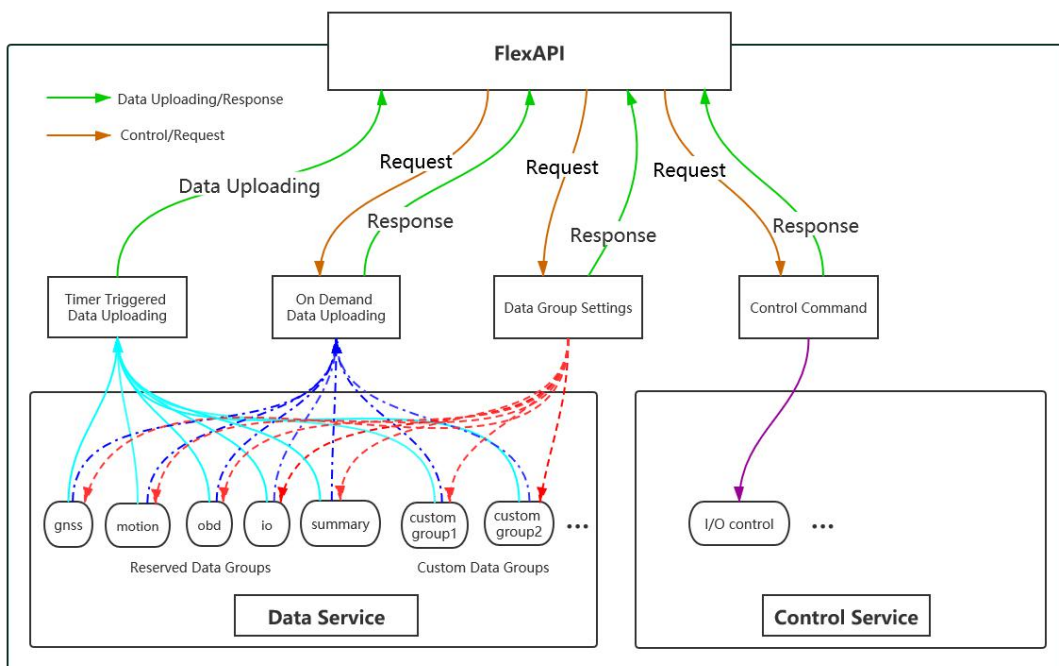
For Cloud platform integration, FlexAPI provides MQTT APIs and acts as MQTT client.

For Edge computing, you can use your Python or Docker APP to access its service, or even from your LAN device software. You also have the freedom to use MQTT or HTTP per your choice.

The following outlines the choices to use FlexAPI:

1. Cloud platform integration.
2. Run python APPs inside VG710 gateway.
3. Run docker APPs inside VG710 gateway.
4. Access FlexAPI from LAN device software via MQTT broker.
5. Access FlexAPI from LAN device software via REST API.

The following figure illustrates FlexAPI in MQTT form.



FlexAPI Use Cases

1. Cloud platform integration

You first need to setup your MQTT broker and make it accessible.

Configure your MQTT broker settings which includes its address, port and authentication.

Client ID needs to be unique so that your platform could distinguish multiple VG710 gateways.

Check WebUI: **APP >> Third-Party Platform**

APP >> Third-Party Platform

Status **Third-Party Platform Management**

Enable	<input checked="" type="checkbox"/>
Server Address	<input type="text" value="118.24.6.21"/>
Port	<input type="text" value="1883"/>
Client ID	<input type="text" value="VG7101234568888"/>
Enable Authentication	<input type="checkbox"/>
Keep Alive	<input type="text" value="60"/> s
Clean Session	<input type="checkbox"/>
TLS Encryption	<input type="text" value="None"/>
Include Invalid Data	<input type="checkbox"/>
FlexAPI Config File	<input type="button" value="Import"/> <input type="button" value="Export"/> <input type="button" value="Restore default configuration"/>

2. Run python APPs inside VG710 gateway

You first need to install python SDK on VG710.

After that you can use VS Code IDE to connect to VG710, edit and pack your APPs.

Then you can import, configure and start your APPs.

Check WebUI: APP >> APP

APP >> APP

Status APP Management

APP Manager Status Running

Python Version Python3

SDK Version 1.3.8 Upgrade

Debug Server Status Running

APP Filesystem Use% 3% of 6640 MB

Data/Log Filesystem Use% 3% of 6640 MB

Extended Memory Card Recognized

Extended Filesystem Use% 0% of 14789 MB

Extended Filesystem DIR /mnt/sd

APP Running Status

ID	APP Name	APP Version	SDK Version	State	Uptime	Action
1	HelloWorld	1.0.0	0.2.0	running	pid 2240, uptime 1 day, 1:45:52	Clear Log Show Log Download Lc

APP >> APP

Status APP Management

Enable APP Manager

Enable IDE Debug

Enable Extended Flash

Import APP Package

No file selected. Browse Upload

APP Configuration

Enable	ID	APP Name	APP Version	SDK Version	Start Parameters	Logfile Size(MB)	Operation Method
<input checked="" type="checkbox"/>	1	HelloWorld	1.0.0	0.2.0		1	Import Conf Export Conf Export App Uninstall

APP Management

Start All Stop All Restart All

ID	APP Name	Operation Method
1	HelloWorld	Start Stop Restart

Apply & Save Cancel

3. Run docker APPs inside VG710 gateway

You first need to install Docker SDK.

On docker management page, you can import or pull your Docker images to create docker APPs.

Check WebUI: APP >> Docker

APP >> Docker

Docker Management

Enable	<input checked="" type="checkbox"/>
Version	18.06.2-ce <input type="button" value="Upgrade"/>
Username	<input type="text" value="admin"/>
Password	<input type="password" value="....."/>
Port	<input type="text" value="9000"/>

[Go to the docker management page](#)

4. Access FlexAPI services via MQTT broker

Your APPs can use MQTT client to connect to MQTT broker for FlexAPI services. Examples include tablet connected to VG710 via Wi-Fi or industrial PC connected to VG710 via Ethernet.

You first need to enable the Local MQTT Broker and set your password for authentication.

Check WebUI: APP >> Local MQTT Broker

APP >> Local MQTT Broker

Local MQTT Broker Management

Enable	<input checked="" type="checkbox"/>
Enable Authentication	<input checked="" type="checkbox"/>
Username	<input type="text" value="admin"/>
Password	<input type="password" value="*****"/>
Persistence	<input type="checkbox"/>
Listening Address	10.5.16.197:1085
Include Invalid Data	<input type="checkbox"/>
FlexAPI Config File	<input type="button" value="Import"/> <input type="button" value="Export"/> <input type="button" value="Restore default configuration"/>

5. Access FlexAPI services via REST API

Your APPs can also use REST style APIs for FlexAPI services.

You first need to enable REST API for localhost or localhost & LAN access.

Check WebUI: APP >> REST API

For localhost only access, which means your APPs run on VG710, check the following settings.

In this case you will use HTTP for service access and you don't need authentication process.

APP >> REST API

REST API Management

Enable	Localhost	▼	
Localhost Listen Address	http://127.0.0.1:5432		
Include Invalid Data	<input type="checkbox"/>		
FlexAPI Config File	Import	Export	Restore default configuration

Apply & Save Cancel

For localhost & LAN access, check the following settings. In this case you will use HTTPS for service access and your APPs need to use access token for authentication.

APP >> REST API

REST API Management

Enable	Localhost & LAN	▼	
Localhost Listen Address	http://127.0.0.1:5432		
LAN Listen Address	https://10.5.16.197:60000		
LAN Access Token	iWUFB4y7720f841yLcR10dLTuo2TO4JR	Refresh Token	
Include Invalid Data	<input type="checkbox"/>		
FlexAPI Config File	Import	Export	Restore default configuration

Apply & Save Cancel

Device Manager

FlexAPI mainly focuses on business and application integration.
For VG710 system management you could use our Device Manager.

What is Device Manager

- Central management platform for InHand router & gateway
- Intuitive user interface
- Fast & Easy management at scale



Device Manager will simplify your daily tasks such as:

- Online/Offline statistics*
- Data usage statistics*
- Location tracking*
- Signal strength tracking*
- Configuration backup & restore*
- Firmware update*
- Edge Computing deployment*
- Setup tunnel for remote trouble shooting*

Effortless Setup

Easy as 1-2-3

1. Register your Device Manager account

Please go to <https://iot.inhandnetworks.com> and create your account.

2. Configure VG710 Device Manager settings

Check WebUI: **Administration >> Device Manager**

Enable Device Manager, input your registered account, then click Apply & Save. After that, VG710 will automatically connect to Device Manager.

Administration >> Device Manager

Status **Device Manager**

Device Manager Enable	<input checked="" type="checkbox"/>
Service Type	Device Manager ▾
Server Address	iot.inhandnetworks.com ▾
Secure Channel	<input type="checkbox"/>
Registered Account	<input type="text" value="liwei@inhand.com.cn"/> Sign up/Sign in
Site Name	<input type="text"/>
Asset Number	<input type="text"/>
Show Advanced Options	<input type="checkbox"/>

Once VG710 is connected to Device Manager, the status shows as below

Administration >> Device Manager

Status **Device Manager**

Device Manager

Device Manager	Connected
Description	Connection Accepted

3. Start monitoring and managing your VG710

Sign in <https://iot.inhandnetworks.com> with your registered account.

Start managing all your VG710 devices via Device Manager.

List view shows as below

Gateway Name	Signal Strength	Serial Number	IMSI	Configuration State	IP	Action
GR9021234213125	4 bars	GR9021234213125	imsi60568249	SYNC succeeded	182.150.21.232	[Edit] [Refresh] [Delete]
VT7101937000162	4 bars	VT7101937000162	imsi93915698	SYNC succeeded	182.150.21.232	[Edit] [Refresh] [Delete]
VG7109999999999	4 bars	VG7109999999999	imsi3573673	SYNC succeeded	182.150.21.232	[Edit] [Refresh] [Delete]
RW9520657934906	4 bars	RW9520657934906	imsi78734604	fetching config	182.150.21.232	[Edit] [Refresh] [Delete]
RZ3011111111112	4 bars	RZ3011111111112	imsi15966029	fetching config	182.150.21.232	[Edit] [Refresh] [Delete]
VT7101937000069	4 bars	VT7101937000069		SYNC succeeded	182.150.21.232	[Edit] [Refresh] [Delete]
RT9122016442995	4 bars	RT9122016442995	imsi71204434	SYNC succeeded	182.150.21.232	[Edit] [Refresh] [Delete]
RT9121934437729	4 bars	RT9121934437729	imsi24536417	fetching config	182.150.21.232	[Edit] [Refresh] [Delete]
IR6993012000000	4 bars	IR6993012000000	imsi10973639	fetching config	182.150.21.232	[Edit] [Refresh] [Delete]
IR7993012000000	4 bars	IR7993012000000	imsi22048374	fetching config	182.150.21.232	[Edit] [Refresh] [Delete]

Device detail shows as below

Serial Number: VG7101234568888
Device Models: VG710
Firmware Version: 1.0.0.12723
Online Duration: 0 day 0 hour 42 minutes 10 seconds

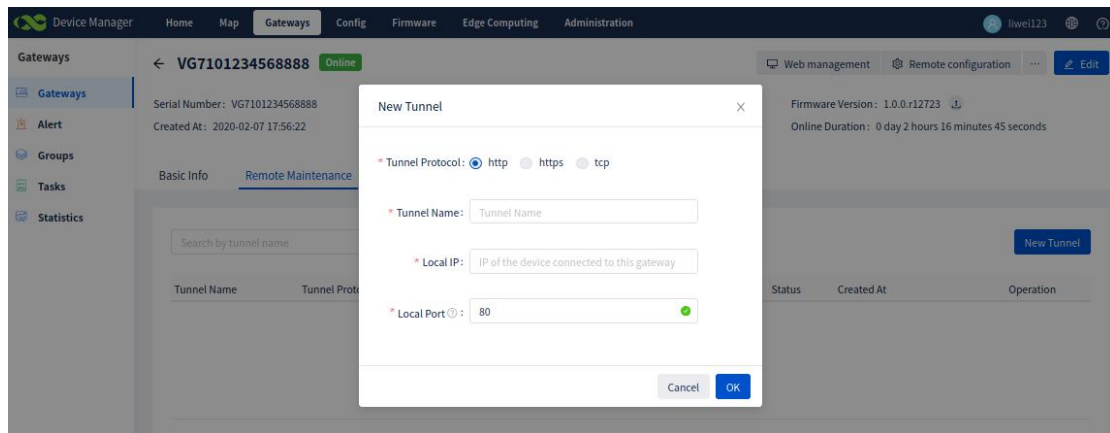
IP: 182.150.21.232
RSSI: 4 bars
IMSI:
Hardware Version:
Phone:
Configuration State: SYNC succeeded
IMEI:
Bootloader Version: 2012.07.r238
Address: 四川省成都市双流区锦华路2段
Login Protocol: mqtt
ICCID:
Reconnect Reason:

Data Usage: 2020-06
Connection Status: Chart, 2020-06-09 - 2020-06-10

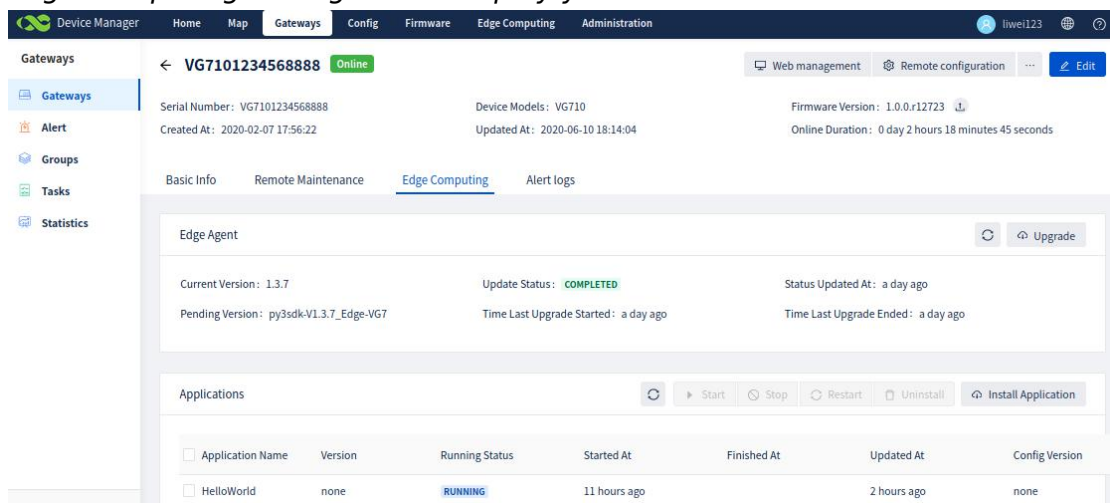
Tasks: All types, All states

Type	State	Progress	Created By	Created At	Started At	Updated At	Action
Fetch running config	COMPLETED	100%	System	18 hours ago	14 hours ago	14 hours ago	[Refresh] [Delete]
Import firmware package	CANCELED	0%	liwei@inhand.com.cn	3 months ago		2 months ago	[Refresh] [Delete]

Direct access your remote device's web services as if you were on-site.



Edge Computing management: Deploy your APPs.



Reference documents

For 3rd party platform integration:

《FlexAPI_Reference_for_3rd_party_platform_v1.0.3.pdf》

For LAN Edge computing application via MQTT broker:

《FlexAPI_Reference_for_LAN_application_v1.0.3.pdf》

For LAN Edge computing application via REST API:

《FlexAPI_Reference_for_LAN_application_REST_API_Version_v1.0.0.pdf》

For Python APP development & deployment:

《VG710_Python_APP_Development_Guide.pdf》