DuxBox™

Getting Started Guide
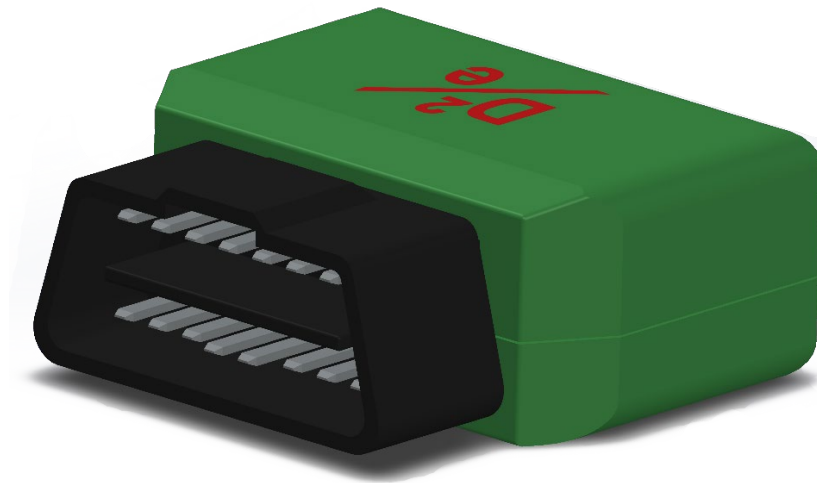
Table of Contents

Document Information

Naming Conventions

None.

Glossary

DuxBox™ – This hardware device that is pluggable into a vehicle.

DuxApp™ – Cloud application that can provide driver safety scoring either based on the information collected by DuxBox-es or from other relevant data streams.

IC – Integrated Circuit or semiconductor chip that is a part of a DuxBox™.

IMU – Inertial Measurement Unit IC containing 3-axis Accelerometer, 3-axis Gyroscope and Magnetometer.

CAN – Controller Area Network that is commonly used for communication with vehicles.

OBD2 – Common protocol to communicate with vehicles over the CAN.

GPS – Global Positioning System receiver IC that is collecting vehicle position, orientation and speed information.

Revision History (Version, Date, Description of change)

v.0.1 June 13, 2022 Initial Draft

Overview

DuxBox™ is a simple, low-cost sensor that plugs into a vehicle's CAN/OBD2 port that collects vehicle's physical movement and acceleration information along with GPS location.

Hardware Description

The ESP32-based DuxBox™ collects GPS vehicle acceleration data alongside with the GPS data. ESP32 is a dual core SoC microcontroller.

DuxBox™ contains Accelerometer, Gyroscope, GPS receiver and CAN controller. Vehicle acceleration data from Accelerometer and Gyro is fused into a single reading. It buffers data until the Internet connectivity is available.

DuxBox™ uses Wi-Fi to connect to a MQTT broker of a cloud service provider via Internet whenever the service is available. It publishes data readings with time stamps into 3 distinct MQTT topics: /GPS, /CAN, /IMU. DuxBox™ receives control and configuration commands via subscription to the /config topic.

Wi-Fi connectivity allows for the automatic upload of data when the vehicle returns to the garage or otherwise has internet connectivity. Using opportunistic data upload process makes it possible to operate DuxBox™ without on-going charges to mobile data connectivity provider. Several Wi-Fi username + password pairs can be programmed into the device.

DuxBox™ has 2 external connectors: 16-pin OBD2 vehicle connector for data collection and power from the vehicle, and a Micro-USB serial terminal connector for programming. Whenever a connection to a SAE J1939, 9-pin round connector is desired, an adapter cable can be used. In this case DuxBox™ must be mounted to the vehicle's body (using Velcro® or self-adhesive trips) to be able to track vehicle's acceleration directions correctly. It should not be left free dangling.

DuxBox's main power source is vehicle's 12VDC drawn from the OBD2 port. In addition, the device has a built-in small, rechargeable Li-Po battery with power management circuitry to maintain continuity and connectivity if the device is unplugged from the vehicle. The internal Li-Po battery can be charged/maintained either through the OBD2 connector's 12V power or through Micro-USB connector's 5V power.

Operation

DuxBox™ collects data as it becomes available. IMU and vehicle's CAN Bus data is being collected only when vehicle engine is running. GPS data is being collected as soon as a satellite fix is obtained.

Data Buffering

DuxBox™ device contains a large data buffer that can accumulate data for several days of driving without being connected to a network. Once the device is connected to a network, the data is published to the appropriate MQTT broker topics and the buffer is emptied. If the internal buffer is full, the old data will be overridden with new, hence keeping about the last 3 days' worth of driving data to be transmitted.

Regardless of when the data is published to MQTT topic, each JSON message contains a time and data stamp as to when a measurement has been taken. The time is UTC epoch time.

DuxBox™ contains a list of known Wi-Fi networks (see provisioning). Periodically it checks if any of the networks are available. Once the connection has been successfully established, the device publishes the data on the MQTT broker.

Since the amount of data being transferred is relatively small, the driver's mobile phone Wi-Fi hot spot can be used to transmit the data.

MQTT Publish – Subscribe Topics
Initially, the DuxBox™ device will publish its JSON data messages into the following topics: *DeviceSerialNumber*/GPS, *DeviceSerialNumber*/IMU, *DeviceSerialNumber*/CAN.

Once the device successfully reads the Vehicle Identification Number (VIN) upon a start of a vehicle's CAN Bus, which usually happens upon the engine start, the messages will be posted into *VIN*/GPS, *VIN*/IMU, *VIN*/CAN topics respectively.

DuxBox™ devices will respond with configuration changes to JSON messages published either on *DeviceSerialNumber*/config or *VIN*/config topics.

DuxApp™
Further data processing may be facilitated by the DuxApp™ that uses proprietary data fusion, heuristic algorithms, and AI/ML technologies for driver safety scoring. By combining the DuxBox™ data with the intersections mapping, historic NHTSA crash, weather, and other data, DuxApp™ flags adverse events and "near-miss" events and develops recommendations for driver improvement.

Datasheet
The Product Datasheet is available [here](#).

Standard Kit Contents
1. DuxBox™ device
2. Reference card with QR Codes – links to the User's Guide and other web resources

User Provided Items
Micro-USB 2.0 cable to computer running terminal emulator on Windows, Linux or macOS.

Other Purchasable Items
DuxApp™, a data analysis and driver scoring cloud application is recommended to be used in conjunction with the DuxBox™.

Additional Hardware References
Refer to the [Technical User Guide](#).

Set up your Development Environment
None. DuxBox™ contains proprietary firmware that is not open for modifications.

Set Up Your Hardware
Connect any terminal or terminal emulator to the Micro-USB connector on the back of the DuxBox™. A computer will recognize the USB device as a COMXX port. Set the baud rate to 115200, n, 1. Using

simple 1-character commands one can provision the device to work with an appropriate cloud service. Type "H" for help menu.

You should see the list of available commands:

W – Upload List of known Wi-Fi networks and passwords
A – Set AWS endpoint
R – Upload Amazon Root CA
C – Upload Device Certificate
K – Upload device's RSA Private Key
N – Print Thing Name
P – Set subtopic names to publish. Default: /GPS, /IMU, /CAN
S – Set subtopic name for Control functions. Default: /config

Setup your AWS account and Permissions

If you don't have a current AWS account, refer to the instructions at [Set up your AWS Account](). Follow the steps outlined in these sections to create your account and a user and get started:

- Sign up for an AWS account and
- Create a user and grant permissions.
- Open the AWS IoT console

Pay special attention to the Notes.

Create Resources in AWS IoT

Refer to the instructions at [Create AWS IoT Resources](). Follow the steps outlined in these sections to provision resources for your device:

- Create an AWS IoT Policy
- Create a thing object

Pay special attention to the Notes.

Provision the Device with Credentials

1. In the AWS IoT console, create a Thing, choosing Manage / Things / Create things.
   On the Create things page, choose Create a single thing, then choose Next.
   On the Specify thing properties page, enter a name for your thing, such as D2_GreenBox, the default name, unless you changed it earlier.
   When naming things, choose the name carefully, because you can't change a thing name after you create it. Choose Next.
   On the Configure device certificate - optional page, choose to Auto-generate a new certificate (recommended). Choose Next.
   On the Attach policies to certificate - optional page, select the policy you created in the previous section. Choose Create thing.
   On the Download certificates and keys page download each of the certificate and key files and save them for later. You'll need to upload these to your DuxBox™ in the next steps.
   Download the Amazon Root CA 1, RSA 2048 bit key.

Important: You must save the certificate files before you leave this page. After you leave this page in the console, you will no longer have access to the certificate files.
Choose Done. After you complete this procedure, you should see the new thing object in your list of Things.

2. In the terminal connected to the DuxBox™ use R command to upload the Amazon Root CA 1 certificate obtained in the step 1 above to the device.
3. In the terminal connected to the DuxBox™ use C command to upload the Device Certificate obtained in the step 1 above to the device.
4. In the terminal connected to the DuxBox™ use K command to upload the Device RSA Private Key obtained in the step 1 above to the device.
5. In AWS IoT console go to Settings and copy the device data endpoint address.
In the terminal connected to the DuxBox™ Set the device to use this address using A command.
6. In the terminal connected to the DuxBox™ use W command to upload the list of known Wi-Fi networks and passwords. You can upload up to 64 networks as a comma-separated pairs of username, password.

Run the Test

To check the operation of the device, In the AWS IoT console, select Test / MQTT test client.

In Subscribe to a topic tab, type # and hit Subscribe. You should see the GPS JSON messages coming through.

Additionally, you may easily create a rule in Manage / Message Routing / Rules to forward messages from GPS, CAN and IMU MQTT topics to the tables in your favorite database, like AWS Timestream or AWS Dynamo.

Debugging

If the device is indoors, it may not have a reliable fix on the GPS satellites in which case you will not see any published messages in the MQTT test client.

Please note that if your device is not plugged into a vehicle, you will only see GPS topic messages. CAN and IMU message will only be generated when the vehicle is running.

Troubleshooting

Problem	Reason / Solution
One or more of our vehicles do not have a rectangular 16-pin OBD2 connector.	1. If your vehicle has a round SAE J1939, 9-pin round connector, you can use an adapter cable to connect the DuxBox™. 2. When using an adapter cable, please make sure the DuxBox™ is firmly attached to the vehicle's body by a double-sided, self-adhesive tape. 3. If your vehicle does not have either OBD2 or J1939 connector, the DuxBox™ device can still be used to collect limited data about the vehicle. 4. When using the DuxBox™ without connection to the vehicle, the internal battery needs to be recharged nightly using the USB port. Alternatively, the device can be powered through it's USB port by a standard 5V mobile device charger plugged into the vehicle's 12V port. 5. Please note that when using the DuxBox™ without connection to the vehicle, some of the collected information, such as vehicle's VIN number, throttle position, etc. will not be available for collection and analysis limiting the quality of driver's scoring.
MQTT test client does not show any messages in GPS, CAN or IMU topics.	1. Check device's Wi-Fi connection by observing messages on the terminal that is USB-connected to the DuxBox™. 2. Make sure the device can "see the sky" and is able be able to lock on the satellites.
Only GPS topic messages are showing on the MQTT test client.	1. Make sure the DuxBox™ is securely plugged into the OBD2 port of the vehicle. 2. Make sure the vehicle's engine is running.
IMU topic messages are not showing on the MQTT test client.	1. When vehicle is running, but not moving it will not be experiencing any trackable acceleration events and IMU topic will report nothing. This is normal.

Technical Support and Customer Service

1-877-224-7708 ext. 3

www.dux.eco

info@dux.eco