# Speed and Cadence Sensor Summary

## Introduction

The Smart Bike project uses a Wahoo Speed and Cadence Sensor to capture both speed and cadence data. This document will summarize how the relevant code operates, some examples of the payload and how the sensor fits into the architecture of the Smart Bike as a whole.

More information on the sensor can be found through the following link: <https://au.wahoofitness.com/devices/bike-sensors/speed-and-cadence-sensors-bundle>

## Code

This sensor requires two scripts to operate which include ‘mqtt\_client.py’ and ‘cadence\_speed\_sensor.py’. The ‘mqtt\_client.py’ is a MQTT client that enables publishing to and subscribing from MQTT topics in HiveMQ cloud and, in this instance, is used for speed and cadence data. This script ultimately provides many of the functionalities needed to connect with the necessary components such as the HiveMQ MQTT Broker, establishing a connection between Raspberry Pi and Arduino and much more. The ‘cadence\_speed\_sensor.py’ script uses the functionalities of the ‘mqtt\_client.py’ script to set up a number of different connections and also to publish data from the sensor to the MQTT broker.

The ‘cadence\_speed\_sensor.py’ script first establishes a connection between Raspberry Pi and Arduino, sets up the HiveMQ connection, sets up Bluetooth Low Energy (adafruit\_ble), gets a MQTT client connection and finally connects to the Smart Bike’s speed and cadence sensor. It will connect to multiple speed and cadence sensors if there are more than one Smart Bike. Thereafter, the script will continuously get the speed and cadence values from the sensor and publish said data to the MQTT broker.

## Example Payloads

Below is a table showcasing examples of the payload that would be published to the MQTT broker. In this case, speed indicates how fast an actual bike would move and is measured in metres per second and has a range of 0 to 30. Cadence indicates how fast the pedals of the bike move and is measured in revolutions per minute and has a range of 0 to 200.

|  |  |  |
| --- | --- | --- |
| **Topic** | **Payload** | **Description** |
| Bike/00001/speed | Ts: 11662625808, speed: 15 | Bike One is traveling at 15 metres per second |
| Bike/00001/speed | Ts: 11662625808, speed: 27 | Bike One is traveling at 27 metres per second |
| Bike/00002/speed | Ts: 11662625808, speed: 9 | Bike Two is traveling at 9 metres per second |
| Bike/00001/cadence | Ts: 11662625808, cadence: 173 | Bike One is travelling at 173 revolutions per minute |
| Bike/00002/cadence | Ts: 11662625808, cadence: 89 | Bike Two is travelling at 89 revolutions per minute |
| Bike/00002/cadence | Ts: 11662625808, cadence: 27 | Bike Two is travelling at 27 revolutions per minute |

## Architecture

How the Wahoo Speed and Cadence Sensor fits into the architecture of the Smart Bike as a whole is showcased in figure 1 below. As you can see, the speed and cadence sensor communicates with a Raspberry Pi device and makes its way to the HiveMQ MQTT Broker and eventually ends up in the MongoDB database. The stored data is then able to be used by the end user systems such as the mobile application and Unity software.

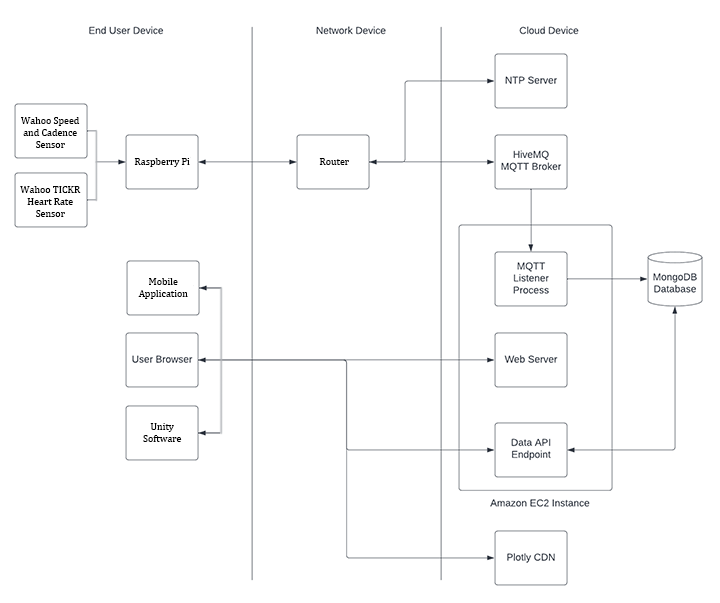


Figure - Suitable Architecture based on the architecture provided by Adrian Grigo and modified by myself