

**FLYINGFOX**

HOVER GAMES 2019



Soheil Bahrami, NXP Hamburg

Mahmoud Sherrah, NXP Hamburg

TABLE OF CONTENTS

[INTRODUCTION 3](#_Toc31394726)

[PROBLEM 4](#_Toc31394727)

[SOLUTION 5](#_Toc31394728)

[SYSTEM COMPONENTS 7](#_Toc31394729)

[GOOGLE CORAL EDGE TPU + CAMERA 7](#_Toc31394730)

[SIGFOX 8](#_Toc31394731)

[PX4 FLIGHT MANAGEMENT UNIT 11](#_Toc31394732)

[WORKFLOW 14](#_Toc31394733)

[DRONE ASSEMBLY AND FIRST FLIGHT 14](#_Toc31394734)

[POSENET AND CORAL BOARD 14](#_Toc31394735)

[SIGFOX 15](#_Toc31394736)

[DEMO FLOW 16](#_Toc31394737)

[BILL OF MATERIALS 18](#_Toc31394738)

[OUTLOOK 19](#_Toc31394739)

[LIST OF FIGURES 20](#_Toc31394740)

# INTRODUCTION

In the age of information we currently live in, the value of collection and usage of information is growing in exponential fashion. Even large-scale organizations can cough up millions and sometimes billions of dollars to acquire a huge data set that can assist in prediction of market trends and user preferences. Moreover, Artificial Intelligence (AI) will define the next generation of solutions to everyday life. Therefore, it is high time AI and machine learning is integrated into drone technology, not only for recreational activities but rather for rescue missions and potentially saving lives.



Figure 1: FlyingFox fully AI equipped and ready to help fighting fire

Enter FlyingFox: an autonomous robust quadcopter drone equipped with an HD camera and a powerful on-board AI module – Google Edge TPU - which is able to identify human gestures and instantly sends a summary of the gathered information through the Sigfox network to the fire brigade members before they even arrive to the location of the fire outbreak.

# PROBLEM

“One of the main problems we face as firefighters is the lack of information during dispatch” – Dann Annan, Fire Brigade Lead, Hamburg, Germany.



Figure 2: FlyingFox team meeting up with Hamburg's fire brigade

The average response time of fire brigades in Europe and the United States is in the region of 6 to 8 minutes from the start of the dispatch call to the arrival of the brigade to the location of the fire outbreak. In this period of time, very minimal information – if not completely lacking - is available to the dispatched fire brigade. Having already asked a leading figure in the fire brigade of Hamburg, Germany, it was confirmed that it is a major issue worthy of a high tech. solution.

# SOLUTION

It is thus the motivation of the FlyingFox team to provide as much valuable information as possible about the fire outbreak to the fire brigade in advance. And what information is more important that human life!

The proposed solution is a robust autonomous quadcopter drone – FlyingFox- equipped with an HD camera and a powerful on-board artificial intelligence component, which is able to quickly identify human gestures and instantly sends a summary of the collected information through the internet to the fire brigade members before they even hit the road.

Each building has a FlyingFox sitting on the roof in sleep mode. Once a fire alarm is triggered in the building:

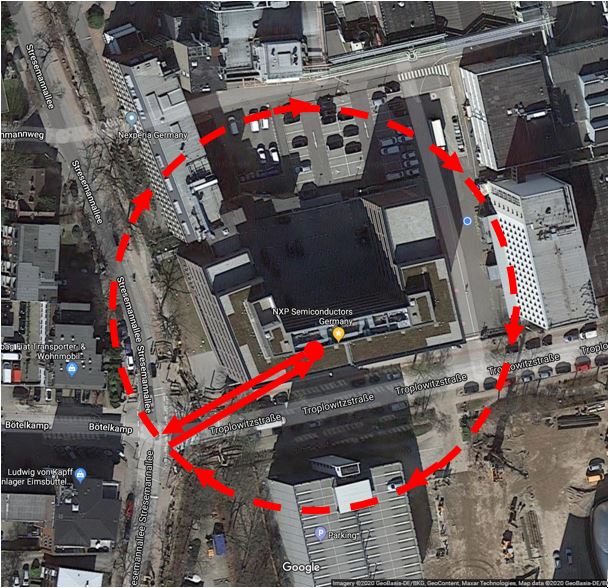


Figure 3: Aerial View of NXP Semiconductors GmbH Office building in Hamburg, Germany

* FlyingFox autonomously completes one circuit around the building in one minute, much shorter than the dispatch time.
* During the circuit, the camera feed is captured by the CPU and forwarded to an on-board real-time offline Machine Learning accelerator.
* The Machine Learning accelerator detects human poses from the camera feed, where a predefined Save Our Souls (SOS) gesture can signal humans in need of help.
* The number of detected humans is tracked.
* At the end of the circuit, the CPU pushes the AI-acquired data to the internet using a low power IoT infrastructure, where the fire brigade can have access to the data on the way to the distress location.

# SYSTEM COMPONENTS

The rich product portfolio of NXP inspired the FlyingFox team to use as much NXP technology and partner infrastructure as possible in order to create a coherent eco-system of fire-fighting drones.

Figure 4 shows the high-level system component diagram:

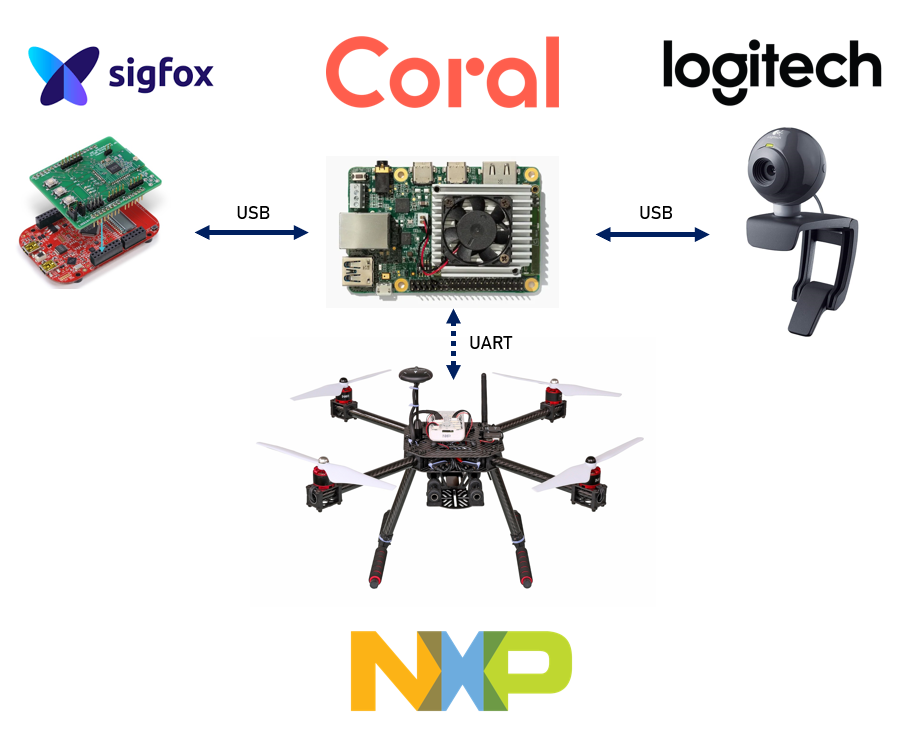


Figure 4: High-level component diagram of FlyingFox

## GOOGLE CORAL EDGE TPU + CAMERA

The Google Coral Edge TPU development is the heart and the brain of FlyingFox.

The Coral board is a fully fledged Linux-based computer powered by the NXP i.MX 8M System-on-Chip (SOC). The Machine Learning add-on board (the EDGE TPU) is a TPU coprocessor capable of performing 4 trillion operations (tera-operations) per second. (TOPS), using 0.5 watts for each TOPS (2 TOPS per watt). For example, it can execute state-of-the-art mobile vision models such as MobileNet v2 at 400 FPS, in a power efficient manner.

The Coral board is able to locally run and process TensorFlow Lite models. For the purpose of the FlyingFox application, the PoseNet demonstration project by Tensorflow is used as basis for the detection of a human pose when one or more persons come into the view of the camera.

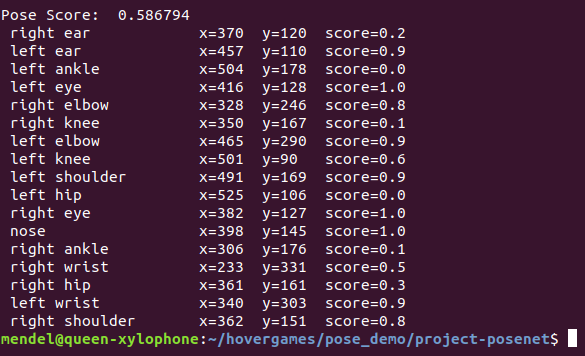


Figure 5: PoseNet overlay output from the Coral board to an HDMI screen

A python script is developed to make use of the PoseNet APIs in order to detect human poses. The algorithm analyzes each frame of the camera and outputs a score of each estimated feature point of a human skeleton (right/left eye, wrist, elbow, knee…etc.). Each feature joint of the human skeleton is located through X-Y coordinates relative to the output screen size.

Figure 5 shows a sample output of the PoseNet demo of only one frame.

## SIGFOX

Sigfox is a world-class worldwide service provider for Internet-of-Things (IoT).

Sigfox provides an ultra-low power interface to the internet for embedded devices. Therefore, a FlyingFox equipped with a Sigfox module is capable of publishing AI-collected information from the embedded Linux platform of Coral to the internet

Each Sigfox module needs to be registered to the network once or based on a subscription plan. The fire brigade and the government can negotiate a deal with Sigfox to enable the registration of millions of FlyingFox around the world.

Sigfox supports multiple network standards around the world, for example ETSI, FCC, ARIB and more. NXP is an official partner to Sigfox and already has developed a chip as well as a complete demo board to access the Sigfox network. Figure 6 shows the components of the attached Sigfox module.

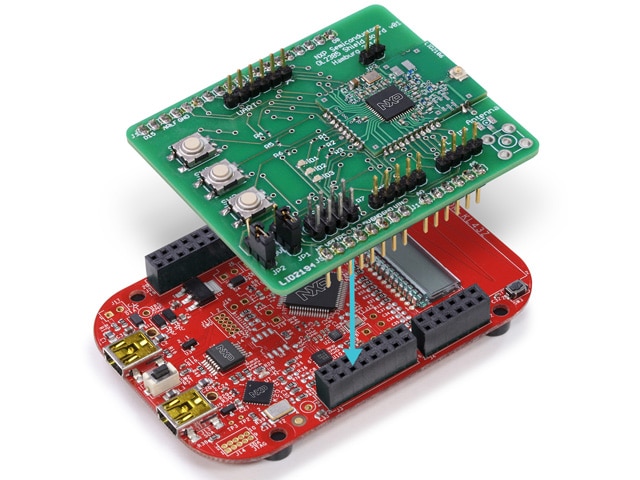


Figure 6: NXP OL2385 Sigfox board mounted on a NXP KL43 microcontroller board

The OL2385 Sigfox development board by NXP (green board in Figure 6) is the module of choice and the main product in NXP’s portfolio to access the Sigfox network. While the OL2385 board can be programmed individually, it can also be controlled externally over SPI by an external host. The SPI control concept speeds up the prototyping of a Sigfox integrated product. The NXP KL43Z microcontroller board, powered by an ARM Cortex M0 microcontroller, is the board of choice where the Sigfox module can plug directly onto the KL43Z to line up the SPI connection.

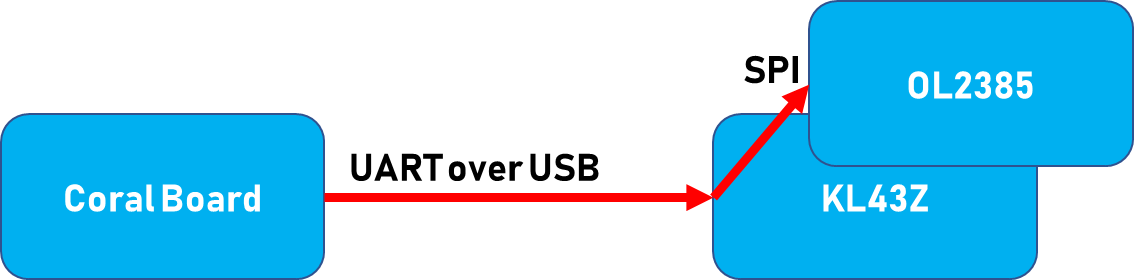


Figure 7: Connection diagram including Coral, KL43Z and OL2385 boards

A sample Sigfox SPI driver is flashed onto the KL43Z board which is connected to the Coral board through USB. Therefore, in order to transmit a Sigfox message, a python driver was developed to send the required UART commands to the KL43Z board, which in turn translates these commands to the SPI format and transmits it to the OL2385 Sigfox module.

Figure 8 shows the impressive coverage map of the Sigfox network worldwide. The coverage is expected to increase as more projects adopt IoT infrastructure where NXP and Sigfox having a leading edge already.



Figure 8: Worldwide Sigfox coverage map

Figure 9 shows the final message presentation to the fire brigade, accessible worldwide through a simple web address over the internet. Due to the limitation of the Sigfox network payload length and message count, payload size from the FlyingFox to the fire brigade is kept at a minimum. Additionally, it is very unlikely that a fire will break out in the same building twice on the same day. Therefore, Sigfox hits a sweet spot of functionality and low system costs for FlyingFox application.

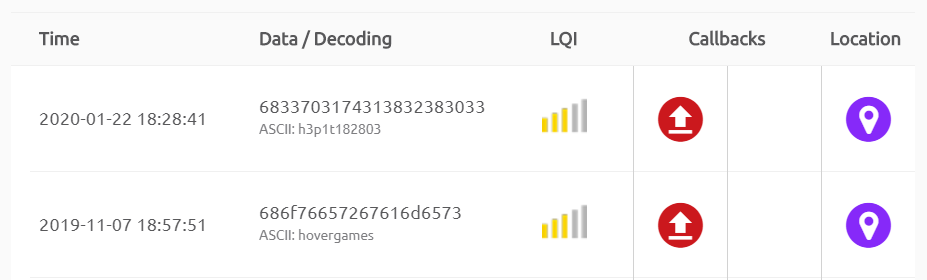


Figure 9: Sigfox message database accessible by the fire brigade through internet

## PX4 FLIGHT MANAGEMENT UNIT

The PX4 Flight Management Unit (FMU) is a pre-developed flight controller module provided by the NXP HoverGames committee. While the expected project scope is to contribute to the open source PX4 project in order to extend the functionality of the NXP drone, the focus in FlyingFox was shifted to developing on a new companion board (the Coral board) to take advantage of the powerful AI engine of Google Edge TPU. The PX4 FMU then transformed to a slave device controlled by the Coral board over a UART port. A python script is developed based on the DroneKit project, which provides python APIs to control the drone’s movement through the FMU.

Figure 10 shows the assembled and fly-ready FlyingFox drone. The Google Coral board is mounted on the drone skeleton and connected via USB to the camera and the Sigfox module, as well as being connected to the FMU over a standard UART port.



Sigfox

Google Edge TPU

FMU

CAMERA

Figure 10: FlyingFox - assembled NXP drone kit with FMU, Coral AI board and Sigfox IoT board.

Figure 11 is a symbolic electrical diagram of the whole system. There was no need for a proprietary circuit to be developed since FlyingFox integrates already developed and mature products and infrastructure.

An XT60 splitter was needed to provide power to the Coral board as well as the original PX4 FMU. An XT60 to USB converter was also needed to drop down the voltage from 12V to 5V. The converter can supply up to 2A which is an ideal supply current for the Coral board.

Since the Coral board only has one USB 3.0 port, a compact USB hub was needed to connect both the Sigfox module (KL43Z board as UART over USB interface) and the Logitech webcam to the Coral board.

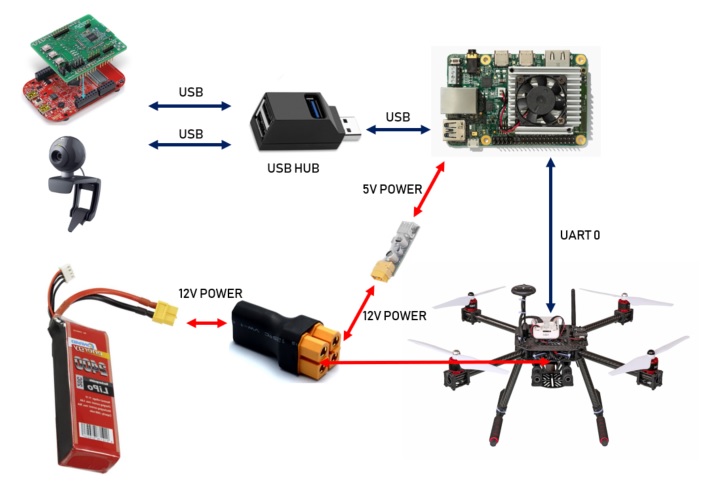


Figure 11: Symbolic diagram of the complete system with power and data lines

# WORKFLOW

## DRONE ASSEMBLY AND FIRST FLIGHT

The first task was to assemble the NXP drone kit. The drone skeleton was built up, the FMU mounted on the center stage, motors and blades were fitted to the skeleton. QGroundControl, the PC based flight control software, was used to control the drone as well as the flight remote controller, through the included telemetry module.

Task was successfully completed as shown in the following video clip:

[](https://www.youtube.com/watch?v=b3nBpN-8QPc)

Video 1: First flight after assembly of initial drone kit in Hamburg’s famous Stadtpark

## POSENET AND CORAL BOARD

The next task was integrating the AI component – the Google Coral Edge TPU - into the system. The Coral board now takes center stage, instead of the PX4 FMU. The Coral board had to be flashed and prepared for boot up into the Linux-based Mendel operating system.

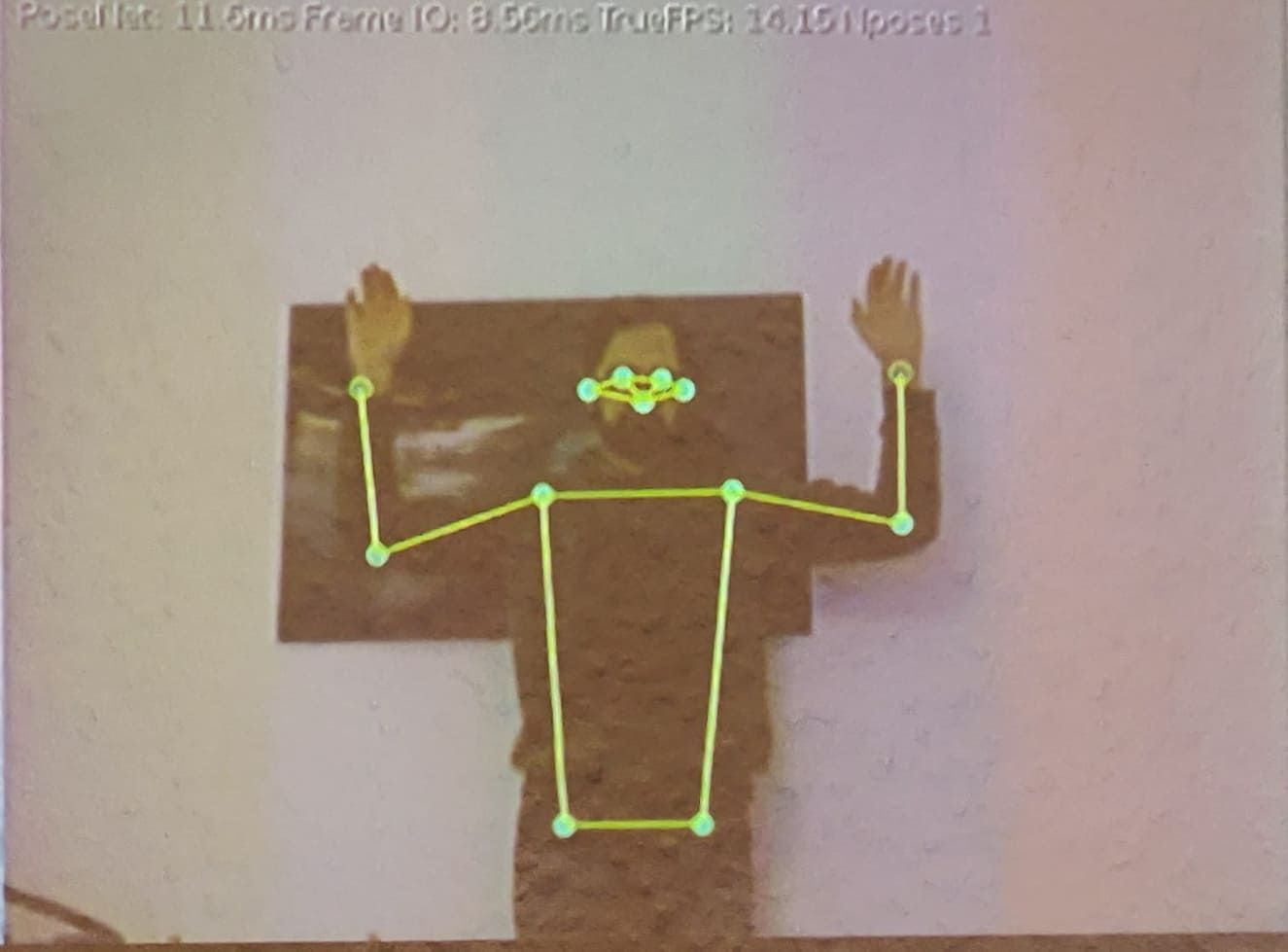


Figure 12: Video overlay of PoseNet pose detection by the Coral board to an HDMI projector

## SIGFOX

The next task was to integrate the Sigfox module to the Coral board. A python script was developed to drive the Sigfox module *sigfox.py*, which sends commands from the USB port of the Coral board to the UART interface which is routed through the OpenSDA chip on the KL43Z and converts the USB signal to the required UART signal format to be read by the ARM microcontroller on the KL43Z board.

Therefore, the Python periphery library was installed on the Coral board to access the UART peripheral, to enable UART over USB communication with the Sigfox module.

The KL43Z software reads the UART commands and transmits the input payload as desired in one Sigfox message.

Due to the limitation of the number of messages to be transmitted, it was required to create a stub file sigfox\_stub.py, which instead of transmitting over the network, it simply outputs to the terminal that a message shall be transmitted, if the stub functions were to be replaced by the original functions in sigfox.py script.

## DEMO FLOW

The demonstration software is included in one main python script: *hg\_demo.py*, which can of course be found on the BitBucket link provided in the project submission. The script uses interfaces provided by the previously mentioned components (TensorFlow PoseNet and Sigfox).

The following video shows the FlyingFox demo in action, where the Google Edge TPU is constantly scanning for human gestures, and video overlay of the detection skeleton features is output through the HDMI port of Coral board and displayed on a TV screen.

In parallel, the scores of each joint is sampled by the Python script and based on the detection quality, a pose is recognized.

[](https://www.youtube.com/watch?v=F0IvV6rk4pM)

Video 2: FlyingFox in action with Coral board video overlay, pose detection and Sigfox interface

Figure 13 shows the debug output of the complete project. A “Detection Counter” is tracking the number of the predefined detected pose. The programs assesses the quality of the pose and then increments the counter. After the circuit around the building under distress is completed, a Sigfox message is transmitted including the total number of detected persons and a timestamp of the measured for integrity.

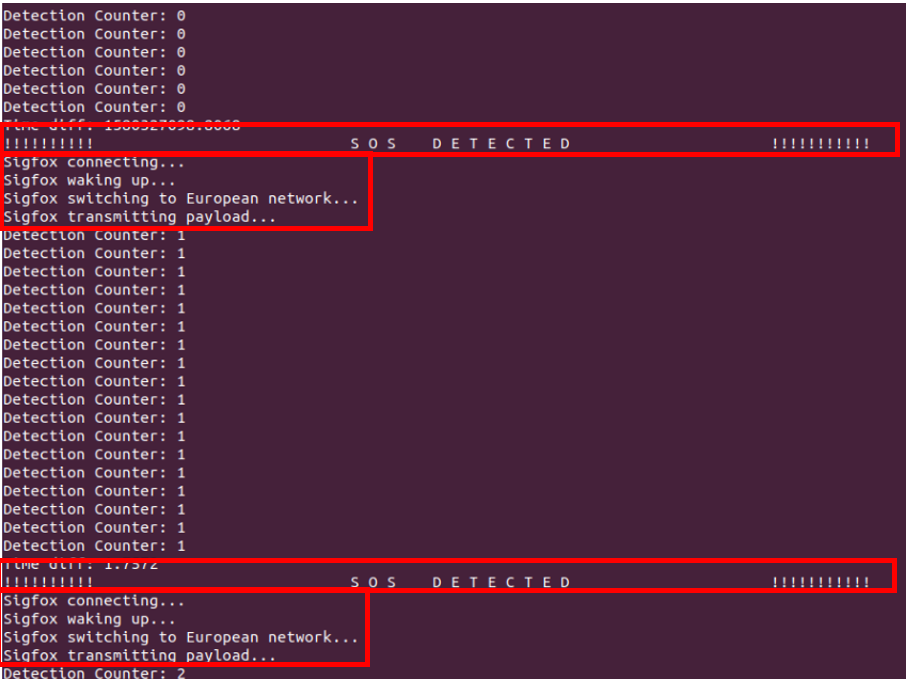


Figure 13: Coral Linux terminal output showing gesture detection and Sigfox interface

# BILL OF MATERIALS

|  |  |
| --- | --- |
| ITEM | COUNT |
| NXP KIT-HGDRONEK66 | 1 |
| NXP HGD-TELEM433 | 1 |
| LIPO BATTERY | 1 |
| GOOGLE CORAL EDGE TPU DEVELOPMENT BOARD | 1 |
| NXP OL2385 SIGFOX DEMO BOARD | 1 |
| NXP KL43Z MICROCONTROLLER BOARD | 1 |
| LOGITECH HD USB WEBCAM | 1 |
| COMPACT USB HUB | 1 |
| XT60 12V TO USB CONVERTER | 1 |
| XT60 SPLITTER 1 FEMALE TO 2 MALE | 1 |
| USBA TO USBC CABLE | 2 |
| USBA TO USBMINI CABLE | 1 |

# OUTLOOK

This proof of concept is just the spark to finally introduce real-time offline Machine learning and AI capabilities to fire-fighting drones. As any proof of concept project, there is quite some room for improvement and optimization:

* Upgrade the camera to an ultra-high definition camera with 60 FPS with advanced low-light performance.
* Train the Machine learning algorithm to detect the floor on which a human gesture was detected and add the information to the Sigfox message.
* For demo purposes, the flight is initiated and controlled through QGroundControl. Coral board shall take over the autonomous control of the drone using MAVLink protocol and UART connection to the FMU.
* Addition of detection of more gesture to communicate particular requests.
* Miniaturization of the complete platform into a single small form factor PCB.

# LIST OF FIGURES

[Figure 1: FlyingFox fully AI equipped and ready to help fighting fire 3](#_Toc31394741)

[Figure 2: FlyingFox team meeting up with Hamburg's fire brigade 4](#_Toc31394742)

[Figure 3: Aerial View of NXP Semiconductors GmbH Office building in Hamburg, Germany 6](#_Toc31394743)

[Figure 4: High-level component diagram of FlyingFox 7](#_Toc31394744)

[Figure 5: PoseNet overlay output from the Coral board to an HDMI screen 8](#_Toc31394745)

[Figure 6: NXP OL2385 Sigfox board mounted on a NXP KL43 microcontroller board 9](#_Toc31394746)

[Figure 7: Connection diagram including Coral, KL43Z and OL2385 boards 10](#_Toc31394747)

[Figure 8: Worldwide Sigfox coverage map 10](#_Toc31394748)

[Figure 9: Sigfox message database accessible by the fire brigade through internet 11](#_Toc31394749)

[Figure 10: FlyingFox - assembled NXP drone kit with FMU, Coral AI board and Sigfox IoT board. 12](#_Toc31394750)

[Figure 11: Symbolic diagram of the complete system with power and data lines 13](#_Toc31394751)

[Figure 12: Video overlay of PoseNet pose detection by the Coral board to an HDMI projector 15](#_Toc31394752)

[Figure 13: Coral Linux terminal output showing gesture detection and Sigfox interface 17](#_Toc31394753)