D8.2 - FINAL SWARM OF DRONES DEMONSTRATION

Deliverable ID	**D8.2**
Deliverable Title	**Final Swarm of drones demonstration**
Work Package	**WP8 - Use Cases Implementation**
Dissemination Level	**PUBLIC**
Version	**1.1**
Date	**14-01-2020**
Status	**Final**
Lead Editor	**DIGISKY**
Main Contributors	**Paolo Pari (DIGISKY)**

Published by the CPSwarm Consortium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731946.

Document History

Version	Date	Author(s)	Description
0.1	31-12-2019	Paolo Pari	First draft
1.0	10-01-2020	Paolo Pari	Final release
1.1	14-01-2020	Paolo Pari	Minor updates

Internal Review History

Review Date	Reviewer	Summary of Comments
10-01-2020	Ákos Milánkovich (SLAB)	Minor corrections and suggestions
10-01-2020	Etienne Brosse (SOFT)	Minor corrections and suggestions

1 Executive summary

This deliverable, namely "**D8.2 - Final Swarm of drones demonstration**", describes the work carried out in "Task 8.1 Swarm of drones use case". The document is based on the predecessor document version "D8.1 - "Initial Swarm of drones demonstration" and contains a few updates concerning the modifications and completion of work during the last work period.

The main objective of the CPSwarm project is to develop a workbench that aims to fully design, develop, validate and deploy engineered swarm solutions. More specifically, the project revolves around three vision scenarios: Swarm of Drones, Swarm Logistics Assistant and Automotive CPS.

WP8 aims at investigating application scenarios for the complete toolchain developed in CPSwarm. The work of this WP is carried out in 4 tasks, one for each use case and an additional task dedicated to use case validation. Strongly driven by industrial needs, the work package focuses on three scenarios related to:

a) Swarms of drones and ground robots;

b) Swarm Logistics scenario;

c) Automotive use case.

Each scenario belongs to a specific task defined within the work package. The swarm of drones and ground robots scenario is defined in the Task 8.1, the swarm logistics scenario at Task 8.2 and the automotive use case at Task 8.3. The fourth and last Task 8.4 includes the use cases outcomes tested and validated under an industrial standpoint.

Table of Contents

2 Introduction

The CPSwarm project is articulated through different stages of implementation, as shown in Figure 1. The experimental demonstration step is a fundamental part of the realization of the final project, since it allows to validate the phases of the correct development of the CPSwarm workbench. Experimental demonstration is covered by the Work Package 8 and is divided into three main scenarios or use cases.

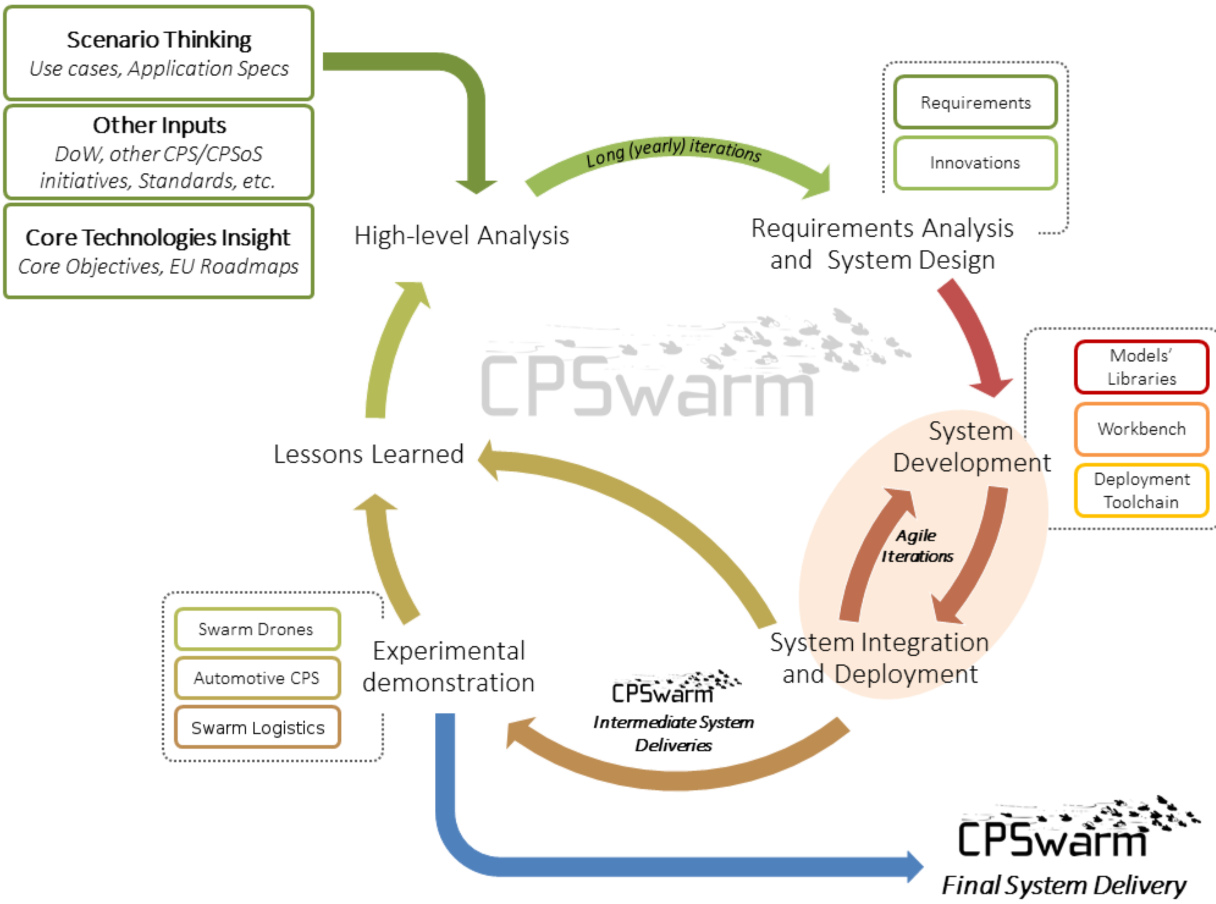

Figure 1 - CPSwarm project lifecycle

This "**D8.2 - Final Swarm of drones demonstration**" is a public deliverable focused on the results of "*Task 8.1 Swarm of drones and ground robots scenario"* at the end of the project.

2.1 Document organization

This document describes in detail the demonstration that was carried out for the drone and ground robots swarm scenario, aimed at simulating a *Search and Rescue (SAR)* operation.

- Section 3 presents an overview of the scenario realized during the final demonstration of M36 of the project.
- Sections 4 and 5 illustrate the hardware and software architectures of the devices used.

- Section 6 explains the performance of the final demonstration, as well as the advantages that the drone swarm solution has brought with respect to a similar SAR operation carried out in a stand-alone or centralized way.

2.2 Related documents

ID	Title	Reference	Version	Date
[D2.2]	Final Vision Scenarios and Use Case Definition	D2.2	1.0	M16
[D8.1]	Initial Swarm of drones demonstration	D8.1	1.0	M24
[D4.6]	Final Swarm Modeling Library	D4.6	1.0	M35
[D7.2]	Final CPSwarm Abstraction Library	D7.2	1.0	M35
[D7.6]	Final Monitoring and configuration framework	D7.6	1.2	M35

3 Scenario overview

For this scenario, the swarm of drones concept is addressed combined with swarm behaviors and evolutionary algorithms.

In CPSwarm we considered heterogeneous swarms of ground robots and Unmanned Aerial Vehicles (UAVs, in this scenario quadcopters) that, in a collaborative and completely autonomous way, scan a large outdoor area searching for human victims or people trapped in the disaster area (see Figure 2). In a Search And Rescue (SAR) scenario, swarms can be exploited for:

a) generating a situation overview of the disaster scene in case of an industrial plant accident including real-time images (Visible Infrared), toxic and explosive gas leakage detection;
b) finding human casualties or people trapped in the disaster area.

The gathered information is used to help security personnel, first responders as well as rescue teams to conduct their mission efficiently. This application scenario has some fundamental requirements: a vast spatial area has to be inspected and information has to be provided to the stakeholders (security personnel, rescue teams, etc.) in real-time, especially in case of an incident. Swarms can reduce the inspection/detection times compared to, e.g., single UAV/rover applications due to their super-linear characteristics (the effect of the overall system is more than the sum of the effects of its individual parts).

Figure 2 – Use case scenario (e.g. industrial plant)

Every single Cyber Physical System (CPS) that takes part in the swarm, either a quadcopter or a rover, is connected to the others in order to continuously exchange information on the current position, the state, and the task that is currently in progress. Quadcopters are in charge of flying over the area and covering it (partially using optimized paths), looking for a missing target that is automatically recognized by analyzing the images of the camera on the drone. Once the target has been identified, the rovers have the task of reaching it and leading it to the nearest collection point via optimized trajectories and avoiding obstacles along the way. All these operations are carried out autonomously, thanks to swarm algorithms that take into account the local behavior of the individual CPS and the global context.

3.1 Swarm of drones and ground robots description

The purpose of the demonstration is to validate the swarm algorithms using heterogeneous devices, each of them has a specific task depending on the features and sensors it is equipped with. One of the important features of the swarm algorithm is the "dynamic" response, namely the ability to adapt the behavior of the swarm members (CPS) in function of an unplanned change of the situation, such as a failure of one of the members of the swarm or the presence of moving obstacles.

3.2 Mission setting

We consider a scenario in which a swarm of drones and ground robots is used in order to optimize the time of the entire operation.

- The scenario story will take place in an industrial or power plant where the swarm will conduct a SAR task, e.g. while a gas leakage is happening (demonstration at Digisky arena).
- A control station will be used to configure some parameters of the mission (e.g., the size of the area to monitor) and to collect data coming from the sensors placed on the CPSs.
- During the mission, drones and rovers will collaborate in order to find people trapped in the industrial area (represented by graphical markers, e.g., QR codes) and help them to reach the exits of the plant.
- Unexpected events will be introduced in the scenario in order to simulate a dynamic situation (e.g., one of the exits will be blocked by some obstacles).
- We would like to demonstrate that the swarm can reduce the inspection/detection times compared to a single drone/rover application.

4 Hardware specification for the demonstration

For the entire CPSwarm project, Digisky produced seven quadcopters (see Figure 3) and three rovers (see Figure 4) by assembling off-the-shelf components. All UAVs have been designed and built from scratch, taking into account the needs for autonomous flight/movement, communication, and execution of the swarm algorithms.

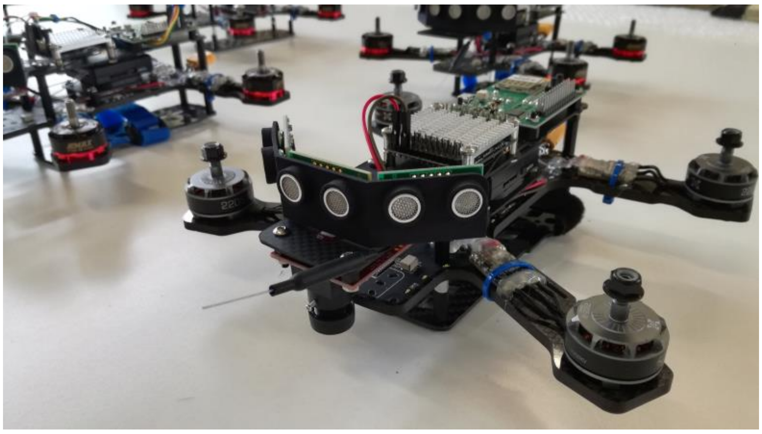

Figure 3 - One of the quadcopters

For indoor use, the rovers have been replaced by Turtlebot robots instead of the previous radio-controlled cars used during the M18 demonstration, more suitable for use on smooth flooring and with greater maneuverability (Figure 4).

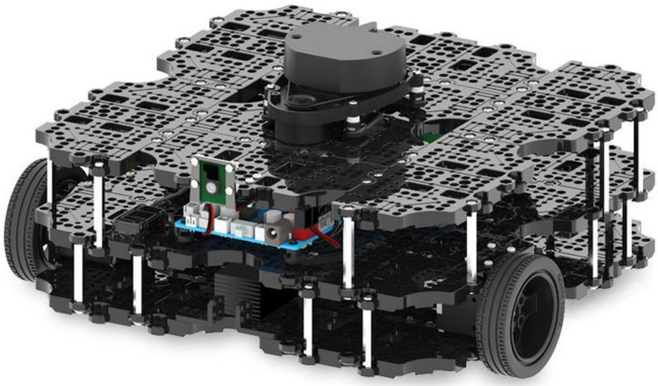

Figure 4 - One of the ground robots (Turtlebot 3 “waffle”)

Figure 5 illustrates the main components of each quadcopter and rover, except for the camera which is dedicated only to UAVs. The following system architecture has been adopted both for quadcopters and for rovers, thanks to the autopilot's flexibility and the ROS-based companion computer, which have made it possible to keep the software of both device types almost completely unaltered and changing only parameters depending from the hardware.

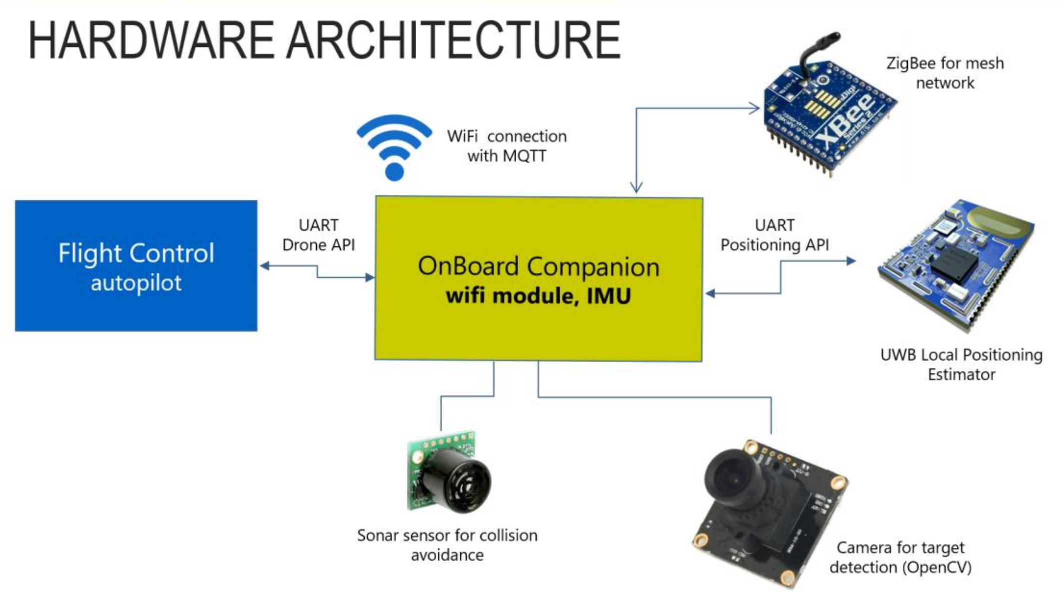

Figure 5 - Main components of quadcopter/rover

As the reader can observe, two main boards are in the drone: the autopilot (or flight controller) and the companion computer.

Autopilot or flight controller

The autopilot is an ARM® Cortex® M4 with single-precision FPU board, equipped with gyroscope, accelerometer, magnetometer and barometer. It has the task of managing the flight of the quadcopter, controlling the attitude through the sensors it is equipped with and acting on the engines to keep the flight stable. Furthermore, it has the task of executing the movement commands that are given by the companion computer to perform the swarm algorithms. As for the rovers, the autopilot has the task of managing the movement of the car in terms of speed and steering.

4.1 On-board companion computer

The companion computer is a single-board Linux-based computer whose job is to execute swarm algorithms and send motion commands to the autopilot. Moreover, thanks to the greater capacity of processing capability, it carries out important auxiliary functions such as collision avoidance, positioning, computer vision, communication with the CPSwarm Monitoring and Control Tool.

Besides all this, the drone is equipped with the following devices (all components and their capabilities summarized in Figure 6):

a) Three ultrasonic sensors: used for collision avoidance against obstacles or other drones in flight.

b) One camera: positioned downwards, necessary for the identification of the target (during the demo, AprilTag markers were used).

c) One ultra-wideband module (UWB): used to allow precision positioning of quadcopters and rovers. The implemented UWB solution allows to simulate a GPS device, providing a positioning accuracy of about 15 cm. By treating the coordinates as based on a normal latitude-longitude reference system, the demonstration was performed in both outdoor and indoor environments, without modifying anything at the software level but simply replacing the standard GPS with the UWB module depending on the environment.

d) One communication module: necessary for data sharing between CPS and ground station.

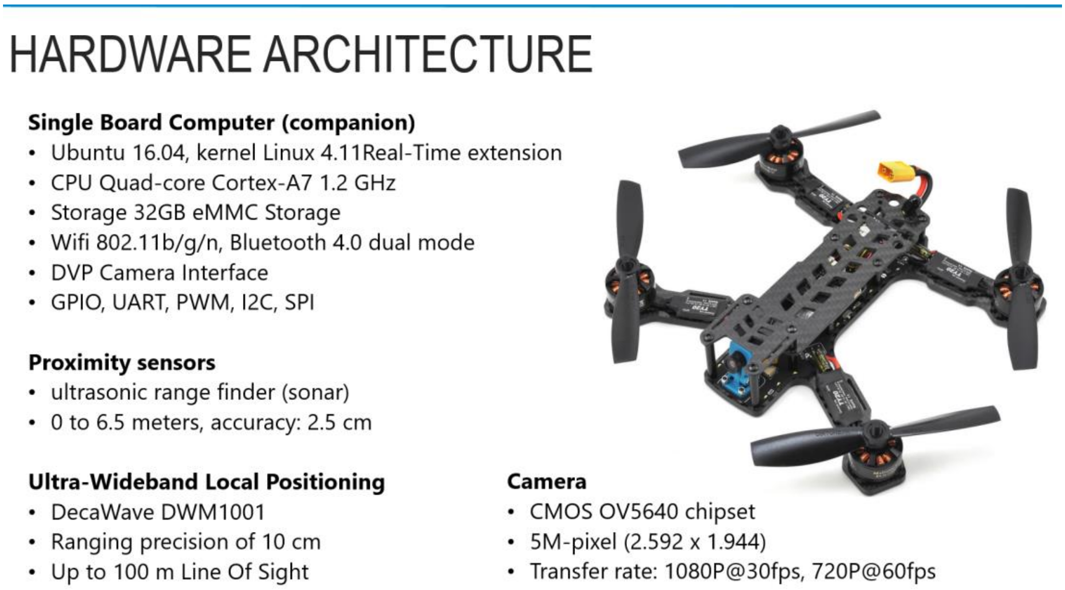

Figure 6 - Hardware features

A careful engineering work was carried out on the various components to make a quadcopter with very small dimensions. For the project, two different frames were created: one with a motor-to-motor size of 210 mm and the other of 150 mm. Both are equipped with the same devices and are fully interchangeable in terms of functionality. The very small dimensions made it possible to fly more quadcopters in small spaces, maintaining an excellent level of stability and maneuverability.

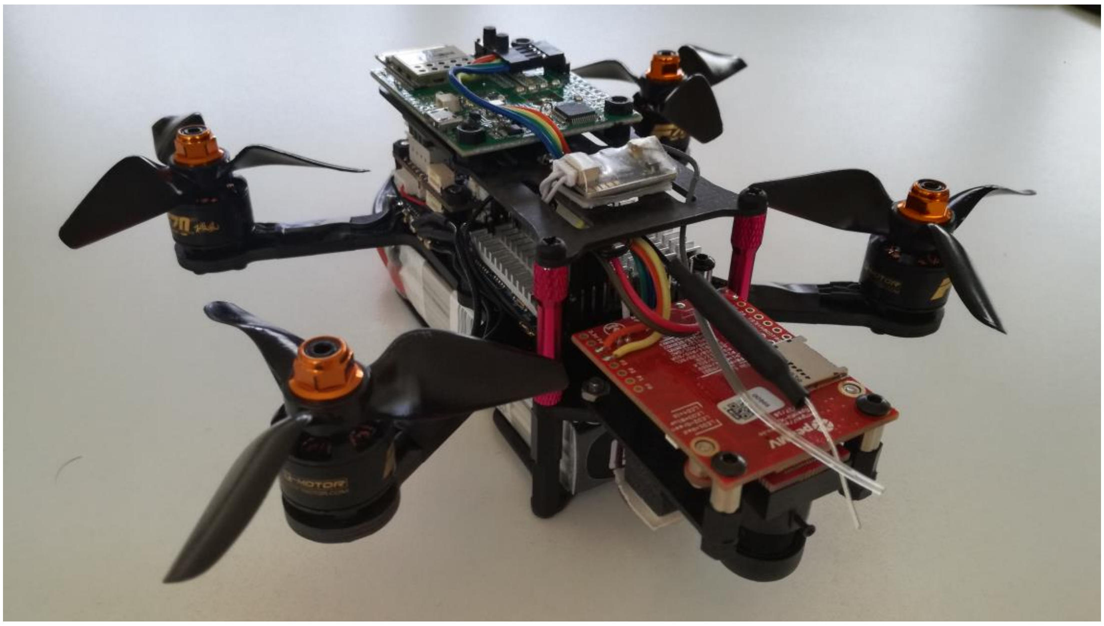

Figure 7 - The smaller version of quadcopter

4.2 List of components

The following components were selected for the construction of the quadcopters and rovers (see Figure 5):

- Companion computer: NanoPi NEO-Air[1], single board computer with Quad-core ARM Cortex-A7 1.2 GHz, 32 GB eMMC storage, Ubuntu 16.04 and kernel Linux 4.11 with Real-Time extension, Wifi 802.11 b/g/n, Bluetooth 4.0, GPIO, 3 UART, I2C, SPI and GPIO.

- Autopilot: PIXFALCON[2], a PX4 fully-compatible flight controller board based on ARM Cortex-M4F at 168 MHz, 16 bit gyroscope, 14 bit accelerometer and magnetometer, barometer, PPM sum signal input, PWM 400 Hz output, UART.

- Ultrasonic sensors: n. 3 MaxBotix I2CXL-MaxSonar-EZ[3] (MB1242), sonar sensor able to detect object from 20 cm to 765 cm with 1 cm of resolution. The sonars are connected to the companion computer via I2C bus and provide a coverage angle of 270° for obstacle detection and collision avoidance.
- Precise positioning: ultra-wideband Decawave DWM1001[4], a plug-and-play board to easily assemble a fully wireless RTLS (Real-Time Localization System) system, including anchors, tags and gateways.

- Camera: OpenMV Cam M7[5] with OV7725 optical sensor and ARM Cortex-M7 on-board microcontroller. This board was programmed with microPython to provide the following features: automatic tag recognition (demonstrated with AprilTag markers); optical flow for the improvement of the position during the flight of the quadcopter; altitude measurement (valid only for indoor environments).

5 Software architecture

UAVs and rovers autopilot systems are based on PX4[6] flight stack and flight controller. The PX4 flight stack is a complete, open source, autopilot solution and it consists of several customizable software packages.

- PX4 Flight Stack: a complete flight control solution for multicopters, planes, Vertical Take-Off and Landing (VTOL) aircraft or any ground robot.
- PX4 Middleware: a highly efficient, lightweight and blazing fast robotics communication toolkit.
- QGroundControl[7]: modern, mobile and desktop user interface to configure the system and execute flights.

PX4 is easy to use and easy to configure and offers a consistent user interface experience across mobile and desktop. Software architecture is based on Real-Time Operating System NuttX[8], which is hardware independent. New modules, libraries and core code can be developed in C++ language. One of the most important point is that ROS (Robot Operating System) is fully supported by PX4.

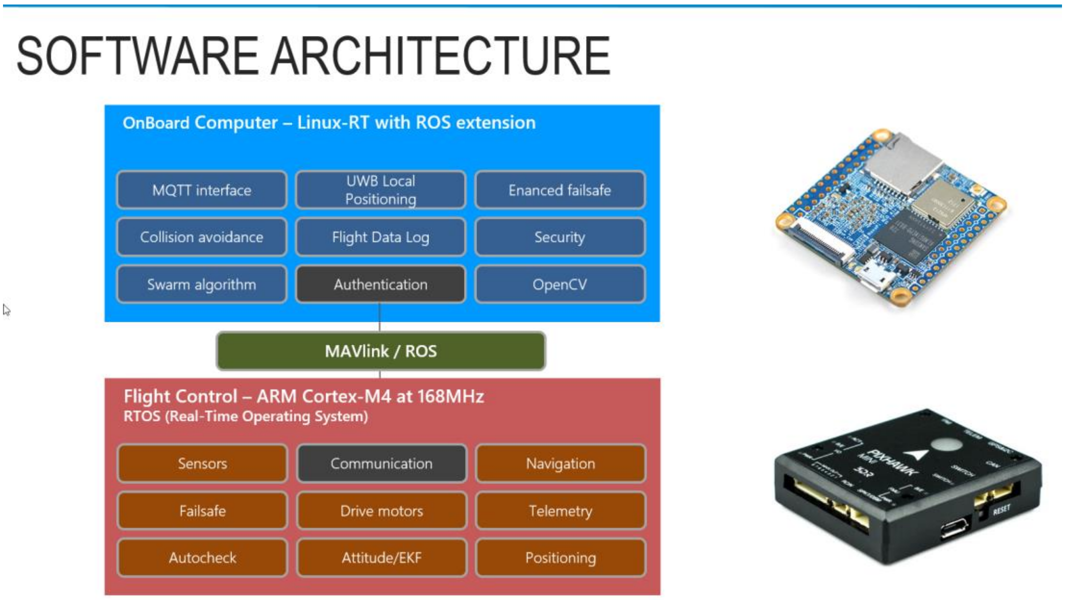

Figure 8 - Software architecture

The modules of the companion computer were created in C++ programming language and are based on ROS, in order to ensure easy integration of the different parts created by the project partners.

They include (with reference to documents for more details):

1. Swarm algorithms (ref. D4.6 – Final Swarm Modeling Library)
2. State machine module (ref. D4.6 – Final Swarm Modeling Library)
3. UWB based positioning system (ref. D7.2 - Final CPSwarm Abstraction Library)
4. Collision avoidance (ref. D7.2 - Final CPSwarm Abstraction Library)
5. Marker detection (ref. D7.2 - Final CPSwarm Abstraction Library)
6. Communication Library (ref. D7.2 - Final CPSwarm Abstraction Library)
7. Monitoring and Control tool (ref. D7.6 - Final Monitoring and configuration framework)

6 Description of the final demonstration

Starting from the following description of work reported in the official CPSwarm project documentation (D2.2):

> *"We will consider heterogeneous swarms of ground robots/ rovers and UAVs to conduct certain missions in the surveillance of critical infrastructure e.g., industrial or power plants as well as in Search and Rescue (SAR) tasks."*

The swarms can be exploited for generating a situation overview of the disaster scene in case of an industrial plant accident including real-time images (VIS, IR), toxic and explosive gas leakage detection and finding human casualties or people trapped in the disaster area.

6.1 Story

An explosion at the power plant in Pripyat has caused an electric blackout in the entire town. There is fire everywhere at the plant. Workers are stuck in the middle of fire. Fire fighter vehicles have arrived. They are trying to control the fire. It is difficult to estimate how many casualties are still in danger. The captain of the firefighter's team decides to survey the whole area for casualties.

He goes to the control station and feeds in the details of the power plant and fire coverage into the CPSwarm surveillance framework. The framework analyses that there is a need to send in a swarm of 15 drones and 8 rovers to provide full coverage. The swarm goes into the fire, collaborating with each other to identify the casualties as well as the fire-free path for the members of the swarm. The drone flying over the shop floor identifies that there are 2 casualties stuck in the shop floor. It locates a rover nearest to the shop floor and sends a request to have a closer look at the location where the casualties are identified. Upon confirmation of presence of casualties, the rover sends the signal back to the control station and help is sent to the identified location.

6.2 Demonstration explanation

As described in paragraph 3.1, the demonstration took place inside the Digisky arena. The final demo will consist of an improved version of the previous one done on M18.

During the last M18 demonstration we used the arena that Digisky has specially designed for the safe flight of drones, consisting of a segregated area of 30x25 m for the sides and 8 m in height, bounded with protection nets so as to be able to fly in compliance with the actual Italian regulations. Since the final demonstration is held in February, it was decided to use an equivalent space in an indoor environment, in order to have more shelter from cold temperatures and bad weather conditions (Figure 9).

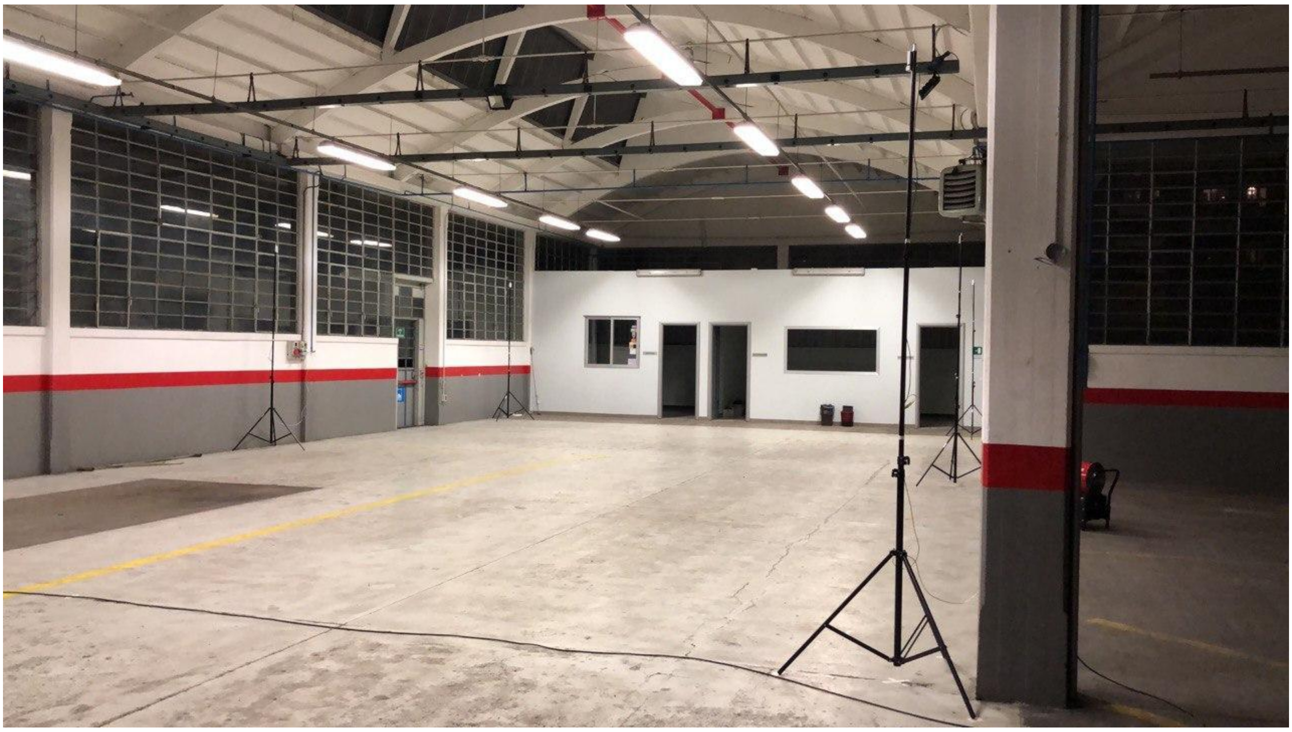

Figure 9 - The indoor arena

Furthermore, during all this last period we have had the opportunity to perform tests, in particular on the swarming algorithms, inside a cage built at the Links headquarters (Figure 10).

Figure 10 - The cage

New features implemented in the final demo

- A swarm of 5 drones starts a coverage task (instead of 2 drones in M18).
- New version of the coverage algorithm, which has been optimized to minimize time.

- A movable target it used to show tracking task.
- Rover used in M18 are replaced by Turtlebots robot that offers better maneuverability in small area.
- Given the period (February), the demo takes place in an indoor environment.
- Due to the previous reason, the UWB localization system is used instead of GPS.
- New algorithm of path planning with obstacle avoidance for the rovers.
- A better collision avoidance algorithm for drones.

At the beginning of the mission:

1. The drones patrol the selected area using an optimized swarm strategy.
2. The rovers wait for a call for intervention.
3. Every fixed amount of time (e.g. 500ms) all the members of the swarm exchange their position in a broadcast channel (among each other and to the control station).

When a drone discovers one of the casualties (markers):

1. The drone communicates the position of the marker to the rovers (and the control station) and starts to hover above the marker communicating possible changes of position.
2. In case of the target moves, the drone follows it by tracking.
3. The drone decides which rover is more suitable to reach the casualty (e.g., the closest rover is selected).
4. The selected rover reaches the marker using an emergency exit strategy, exploiting its current knowledge of the area and information coming from other CPSs.
5. After reaching the selected point, the rover will reach the closest exit, the drone starts doing coverage again.
6. In the event that a drone has to land (e.g. low battery), it is replaced by another drone that resumes running the task from where it left off.

6.3 State Machines

Based on the final vision scenario described above, the following state machine and the corresponding (swarm) behaviors have been designed in collaboration with LAKE (see D4.6. – Final Swarm Modeling Library for more details on the state machine and the algorithms).

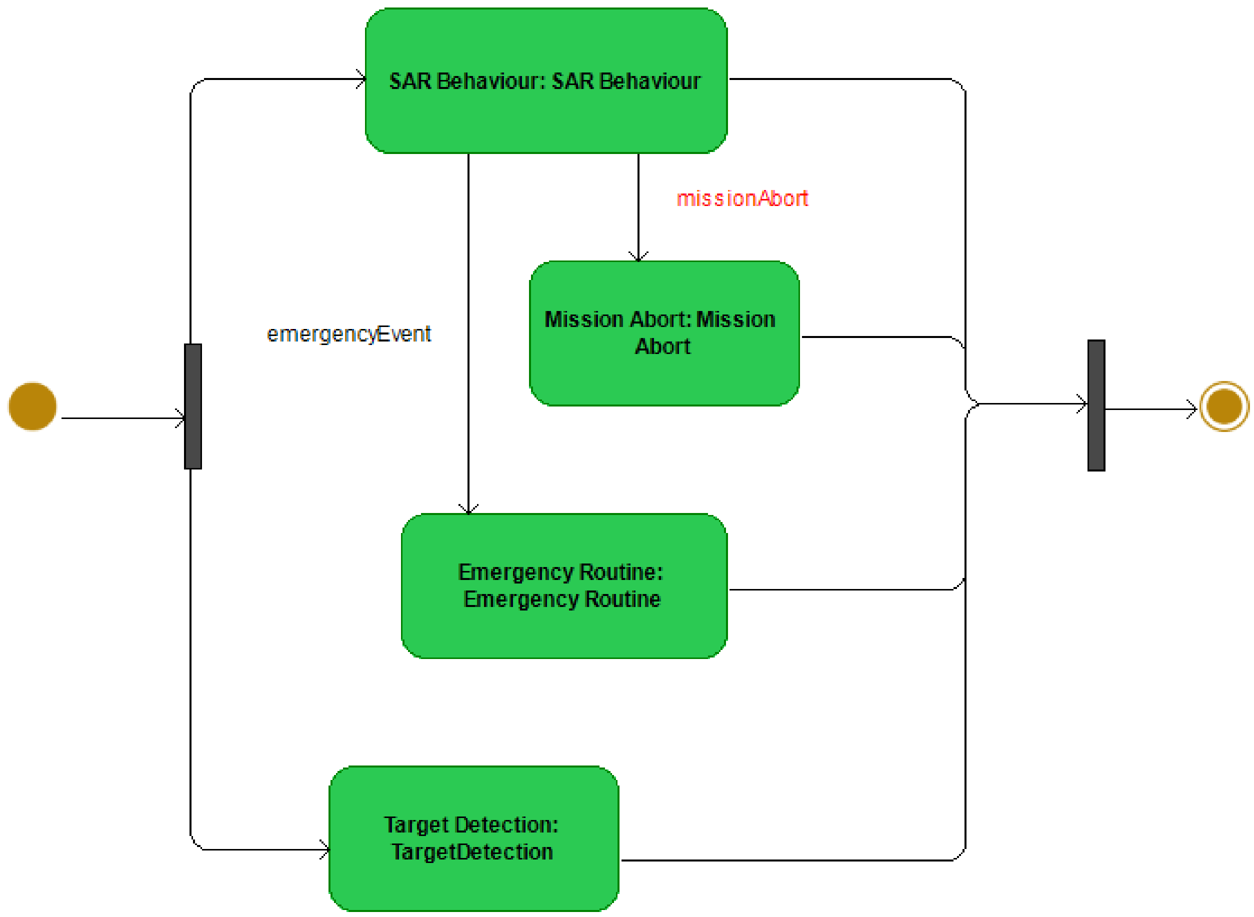

Figure 11 - 1st level State Machine for SAR scenario

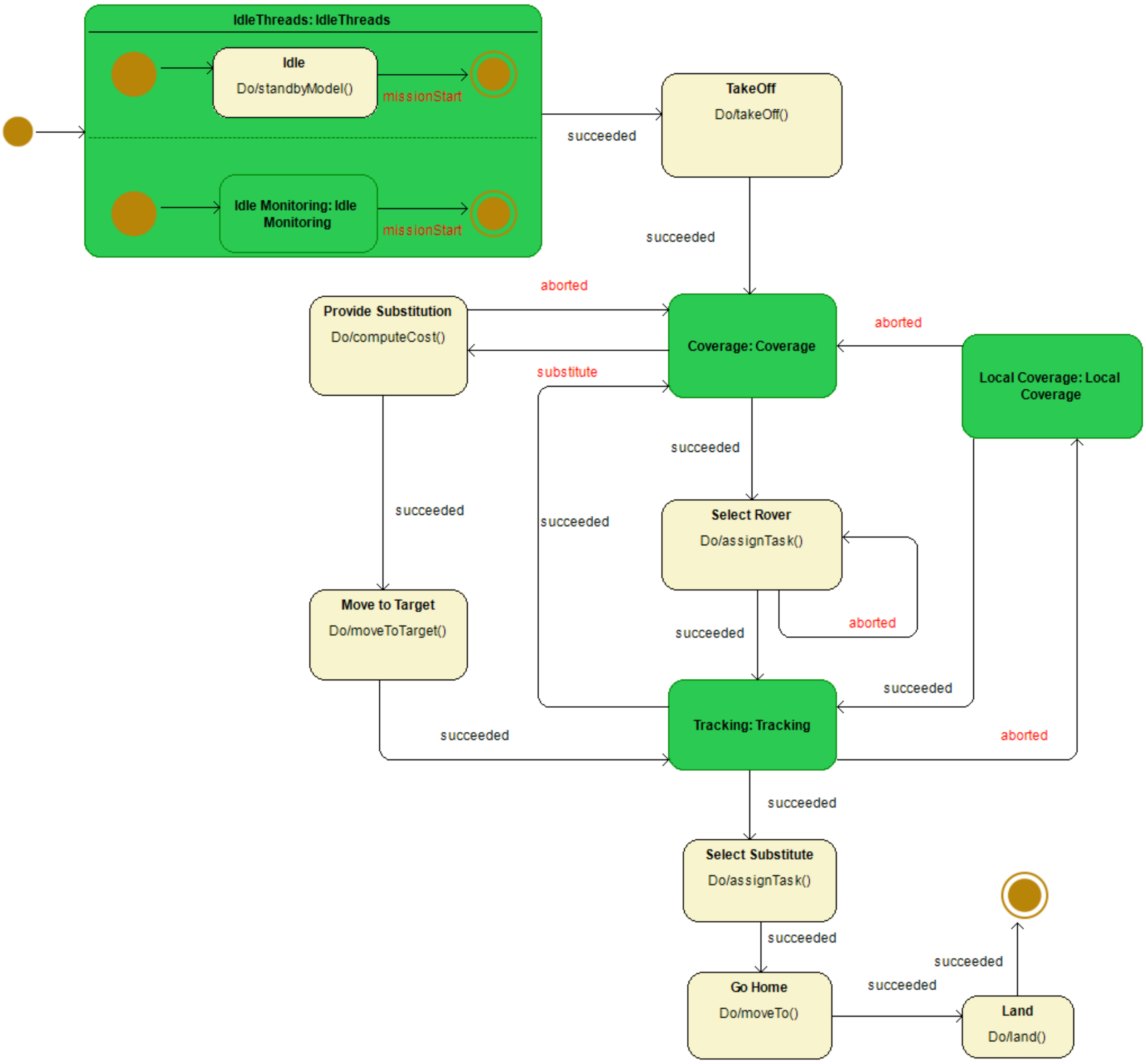

Figure 12 - 2nd level State Machine for SAR scenario

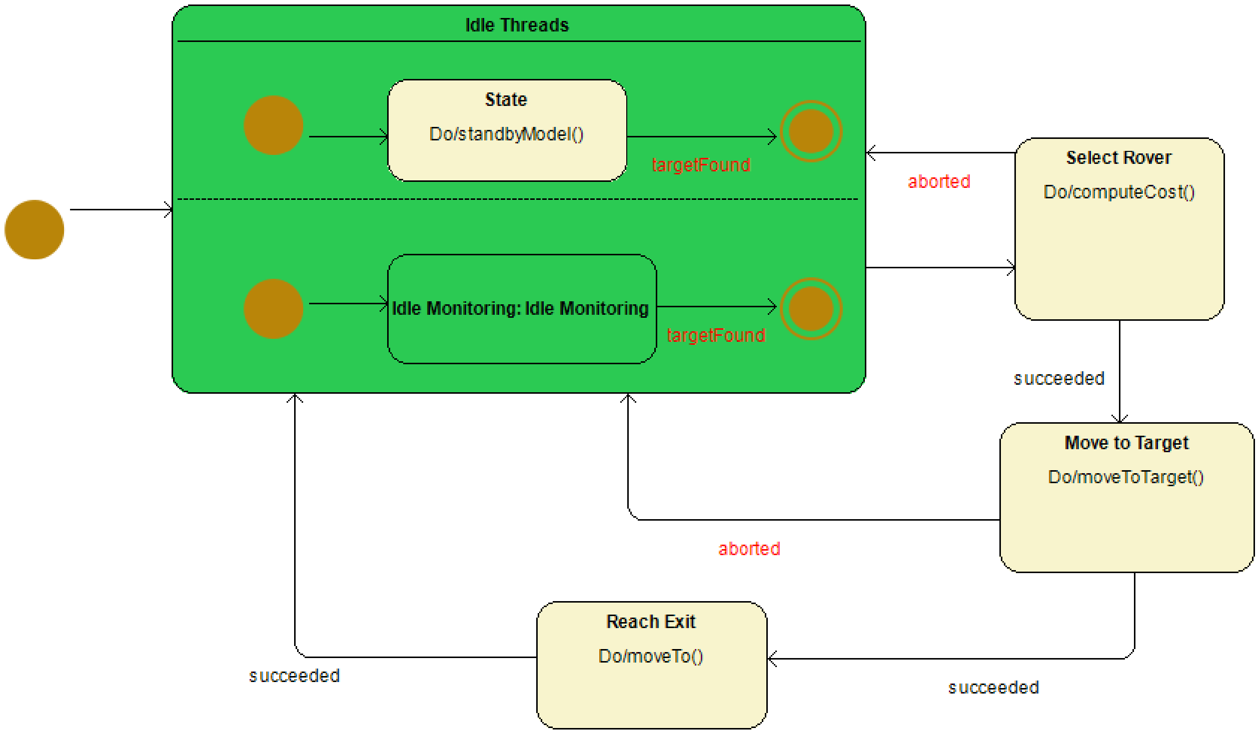

Figure 13 - Rover State Machine for SAR scenario

7 Advantages of using swarm of drones and ground robots

Especially in SAR a single minute can decide between life and death. The inspection cycle time for surveillance can be reduced considerably enabling denser inspection. At the core, there are multiple UAVs and multiple rovers that can act autonomously. They carry different sensors (VIS or IR cameras, microphones, gas sensors, etc.). Either each vehicle carries all sensor modalities or the sensor modalities are distributed among the vehicles. The vehicles carry intelligence via on-board processing capabilities and can communicate among each other (via WiFi, 4G or others). They operate as a self-organizing mixed team where particular tasks for each vehicle are not predefined at mission start but negotiated during mission execution. Such a swarm is highly adaptive to changes in the environment and can act dynamically.

For example a ground rover may order a camera UAV to look for the best path, or a UAV finds something strange and orders additional UAVs with other sensors to check or asks if one of the rovers can move there to perform some action.

Moreover, in contrast to fully centralized control, such a swarm can still operate even if the connectivity among vehicles or with a base station is sparse. The swarm mission itself is defined in a central operation center in the beginning of the mission with a dedicated swarm definition tool (mission planner) that defines the goals and behavior of the swarm, thanks to the CPSwarm Deployment Toolchain. The mission planner will take a number of fixed parameters, e.g., area to cover, swarm size and purpose of mission i.e. look for casualties, etc. In addition to this information, the swarm is fed with the details of the covered area and all possible routes within that area. During a mission, the decision-making process with respect to routes is adaptive based on the latest information opportunistically acquired by the swarm while performing its tasks in the rescue mission.
The central station can additionally collect the sensor data and perform sensor fusion and additional analysis in real time. The vehicles can also form a meshed ad-hoc communication network to improve communication performance. The central station is equipped with suitable user interfaces to enable the operator to influence the swarm (e.g., tell that he/she wants to see a certain scene then the swarm automatically sends the closest vehicle to the scene, camera remote control) or to document the swarm mission including all GPS and sensor data. On top of this, members of intervention teams may access the swarm and its data directly via wearable devices and, possibly influence/modify the swarm tasks. They may be sent out to map a (disaster) area, look for casualties or perform intrusion detection with automated tracking of the intruder.

For using a heterogeneous swarm, the following benefits using can be summarized:

- Drones/rovers start from the same starting point and reach their own positions autonomously.
- Strategic real-time viewpoints during inspections/SAR operations.
- A single mission for all devices: area planning, boundaries, paths and viewpoints.
- Huge reduction in the time taken for inspection/SAR in large sites/areas.
- Human error reduction: repetitiveness of the operations is carried out by unmanned systems.
- Increased security for supervisory staff.
- Real-time interaction with Monitoring and Control Tool, better coordination during the inspection/SAR operations.
- Not only visual inspection by use of different sensors: audio analysis, presence of gas, thermal cameras etc.

8 The added value of the swarm drone scenario

The use of swarm technology in recent years is finding more and more interest in the market, particularly regarding mobile robots and autonomous drones. The realization of this use case has required the overcoming of very complex aspects that concern the interaction of several different devices, the dynamic division of tasks and the adaptation of behavior according to different situations.

In many real-life situations the time factor is a fundamental aspect, not only in a SAR operation but also in less critical contexts for the safety of people, but equally important for economic and efficiency aspects. Having tools at hand to speed up the phases of planning a mission and implement it quickly, represents a key element for Digisky to consider for the near future. We are already actively working to identify new application sectors where the CPSwarm workbench can be of real advantage over the competition.

9 Final steps conducted to completion

Concerning the swarm of drones use case, the final stages carried out during the last period of work were:

a) the final set-up of the hardware configuration of the drones, optimizing the components layout and the wiring;
b) updating the software stack, implementing the latest versions of the algorithms and all the necessary ROS nodes;
c) these operations were repeated for all the 7 drones that Digisky made for the project, thus completing the swarm fleet;
d) finally, the entire set-up was subjected to testing and to the verification and validation process.

10 Conclusion

This deliverable presents the work done in Task 8.1. It provides an overview on the finally conducted activities in the CPSwarm Project with respect to the swarm of drones use case obligations.

All features presented into the document have been integrated into the M36 demo to improve the demonstration. The number of drones has been increased from two to five, as well as the number of ground robots. New algorithms have been implemented for swarm flight control, collision avoidance and target tracking. Optimizations have also been made to the drone hardware in order to further improve flight performance.

The whole CPSwarm Workbench was integrated into the use case, helping and facilitating the development of the demonstration.

Acronyms

Acronym	Explanation
CPS	Cyber-Physical Systems
SAR	Search and Rescue
UAV	Unmanned Aerial Vehicle
GPS	Global Positioning System
UART	Universal Asynchronous Receiver-Transmitter
SPI	Serial Peripheral Interface
RTLS	Real-Time Locating Systems
UWB	Ultra Wide-Band

Table of Figures

References

[1] NanoPi NEO-Air companion computer: http://nanopi.org/NanoPi-NEO-air_Feature.html

[2] PIXFALCON autopilot: http://www.holybro.com/manual/pixfalcon11.pdf

[3] MaxBotix I2CXL-MaxSonar-EZ sonar sensors: https://www.maxbotix.com/

[4] Decawawe DWM1001 ultra-wideband modules: https://www.decawave.com/

[5] OpenMV Cam M7 camera: https://openmv.io/

[6] PX4 flight stack: https://px4.io/

[7] QGroundControl: http://qgroundcontrol.com

[8] NuttX: https://nuttx.org