



D8.1 - INITIAL SWARM OF DRONES DEMONSTRATION

Deliverable ID	D8.1
Deliverable Title	Initial Swarm of drones demonstration
Work Package	WP8 - Use Cases Implementation
Dissemination Level	PUBLIC
Version	1.0
Date	2019-03-26
Status	Final
Lead Editor	DIGISKY
Main Contributors	Omar Morando (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	28-12-2018	Omar Morando (DIGISKY)	First draft
0.2	11-03-2019	Omar Morando (DIGISKY)	Updated draft
1.0	26-03-2019	Omar Morando (DIGISKY)	Fixed minor issues, updated content

Internal Review History

Review Date	Reviewer	Summary of Comments
26-03-2019	Ákos Milánkovich (SLAB)	Fixed a number of minor issues
26-03-2019	Melanie Schranz (LAKE)	Fixed a number of minor issues, added some suggestions about content

1 Executive summary

This deliverable, namely “**D8.1 - Initial Swarm of drones demonstration**”, describes the work carried out in “Task 8.1 Swarm of drones use case”. The consecutive results until M36 will be reported in “D8.2 - Final Swarm of drones demonstration”.

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

Document History	2
1 Executive summary	3
2 Introduction	5
2.1 Document organization	5
2.2 Related documents	6
3 Scenario overview	7
4 Hardware specification for the demonstration	9
5 Software architecture	13
6 Description of demonstration at month 18 th	14
7 The added value of the swarm drone scenario	17
8 Next steps	18

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.

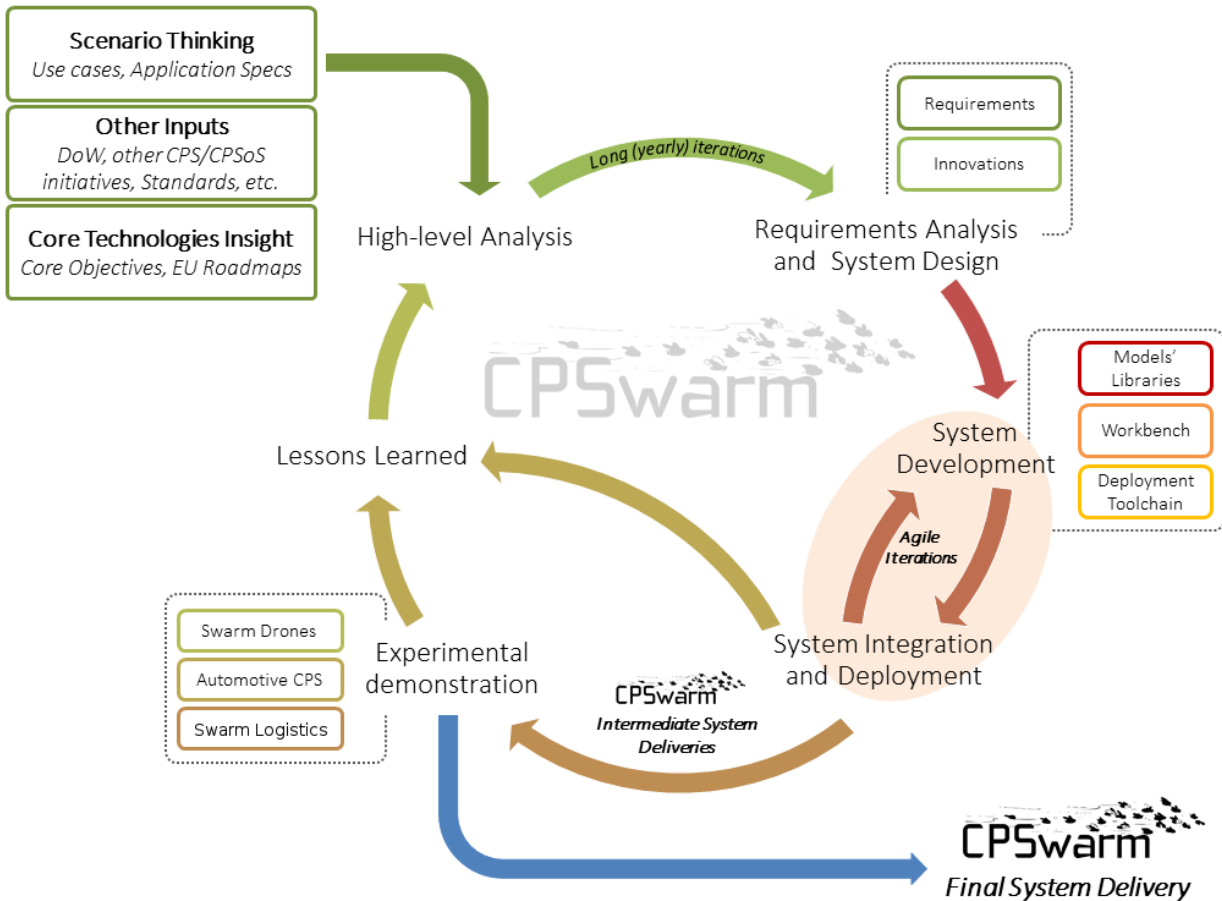


Figure 1 – CPSwarm project lifecycle

This “**D8.1 - Initial Swarm of drones demonstration**” is a public deliverable focused on the results of “*Task 8.1 Swarm of drones and ground robots scenario*” at M18 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 demonstration of M18 of the project.
- Sections 4 and 5 illustrate the hardware and software architectures of the devices used.
- Section 6 explains the performance of the M18 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
[RD1]	Final Vision Scenarios and Use Case Definition	D2.2	1.0	M16
[D4.5]	Updated Swarm Modeling Library	D4.5	1.0	M22

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 UAVs (quadcopter) that, in a collaborative and completely autonomous way, scan a large outdoor area searching for human victims or people trapped in the disaster area. 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 (VIS, IR), 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 which 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 agents (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.

In order to offer a use case that could meet these expectations, we have chosen to simulate a SAR scenario in a limited area of 30x25 m, at the partner Digisky site located inside the Turin Aeritalia Airport. Digisky has an arena specifically designed for the safe flight of drones, consisting of a segregated area of 30x25 m for sides and 8 m of height, bounded with protection nets so as to be able to fly in compliance with the actual Italian regulations.

Mission setting

- The swarm for the demo is composed of two quadcopters and two ground rovers.
- 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 (real case: 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. For the demo only two quadcopters and two rovers were used, due to the preliminary state of the algorithms and of the control system. 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 5 illustrates the main components of each quadcopter and rover, except for the camera which is dedicated only to UAVs.

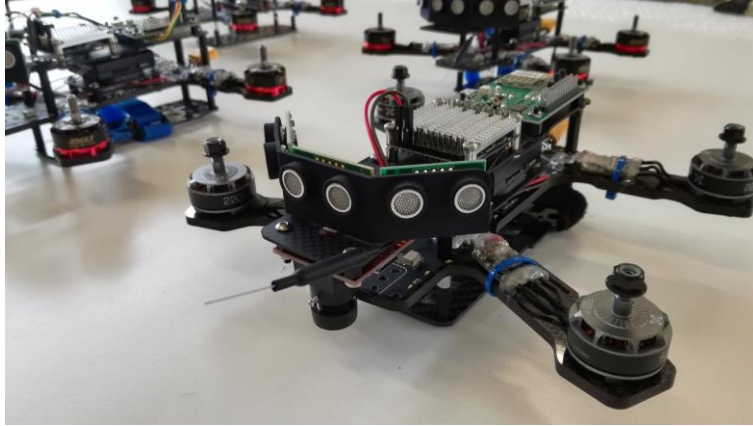


Figure 3 – One of the quadcopters

For the rovers, it was decided to adopt the mechanical structure of a radio-controlled car, modifying the control system and adding the necessary hardware to make it suitable for the swarm's needs (e.g. autopilot, companion computer, UWB, ultrasound etc.).

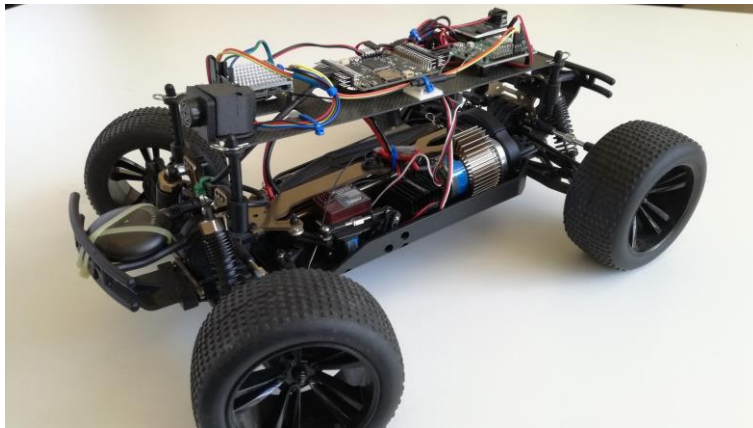


Figure 4 – One of the rovers

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 almost completely unaltered and changing only parameters depending from the hardware.

HARDWARE ARCHITECTURE

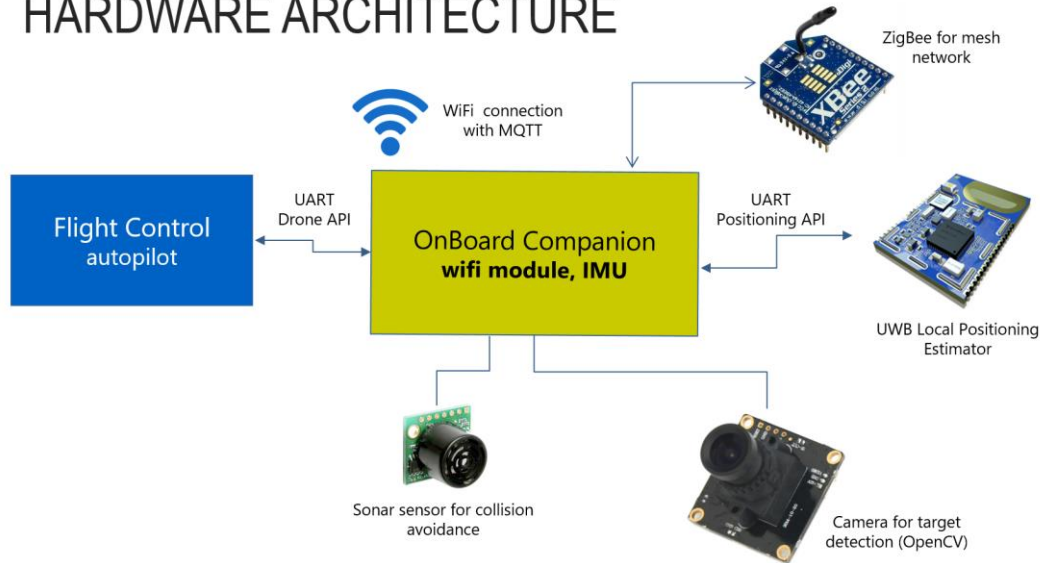


Figure 5 – Main components of quadcopter/rover

As you can see, 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.

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 monitoring station.

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 UWB solution that was implemented allowed to simulate a GPS device, providing a positioning

accuracy of about 15 cm. In this way, by treating the coordinates as based on a normal latitude-longitude reference system, it was possible to perform the demonstration 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.

HARDWARE ARCHITECTURE

Single Board Computer (companion)

- Ubuntu 16.04, kernel Linux 4.11Real-Time extension
- CPU Quad-core Cortex-A7 1.2 GHz
- Storage 32GB eMMC Storage
- Wifi 802.11b/g/n, Bluetooth 4.0 dual mode
- DVP Camera Interface
- GPIO, UART, PWM, I2C, SPI

Proximity sensors

- ultrasonic range finder (sonar)
- 0 to 6.5 meters, accuracy: 2.5 cm

Ultra-Wideband Local Positioning

- DecaWave DWM1001
- Ranging precision of 10 cm
- Up to 100 m Line Of Sight

Camera

- CMOS OV5640 chipset
- 5M-pixel (2.592 x 1.944)
- Transfer rate: 1080P@30fps, 720P@60fps



Figure 6 – Hardware features

A careful engineering work was carried out on the various components in order to make a quadcopter with very small dimensions. For the project, two different frames were created: one with a 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.

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, represented by an AprilTag marker; 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, VTOL aircraft or any ground robot.
- PX4 Middleware: a highly efficient, lightweight and blazing fast robotics communication toolkit.
- QGroundControl: 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, 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.

SOFTWARE ARCHITECTURE

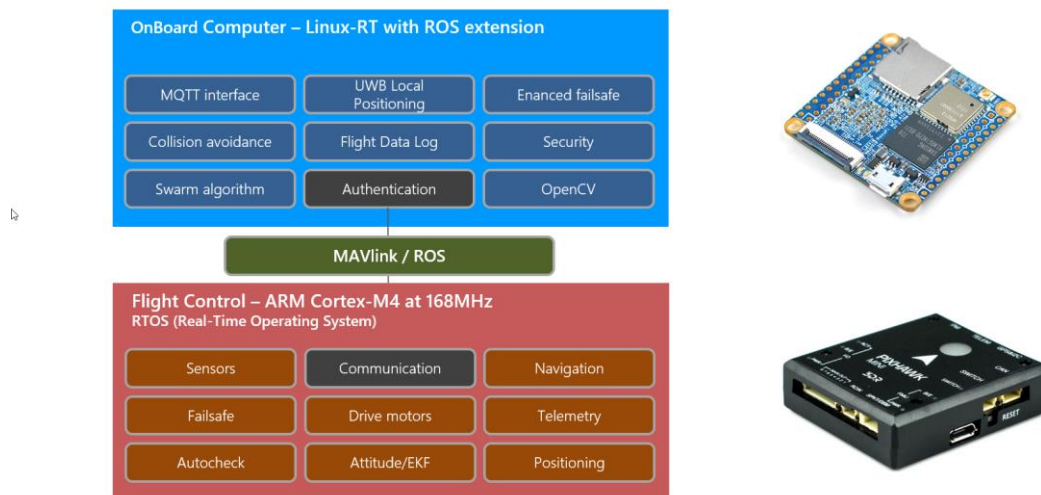


Figure 7 – Software architecture

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

They include:

1. swarm algorithms
2. state machine module
3. UWB based positioning system
4. collision avoidance
5. marker detection
6. communication library

6 Description of demonstration at month 18th

Starting from the following description of work reported in the official CPSwarm project documentation (D2.1):

“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 Demo explanation

As described in paragraph 3.1, the demonstration took place inside the Digisky arena. As mentioned before, for the demo only two quadcopters and two rovers were used due to the preliminary state of the algorithms and of the control system.

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.



Figure 9 – Some tests inside the arena

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 D4.5. – Updated Swarm Modeling Library for more details on the state machine and the algorithms).

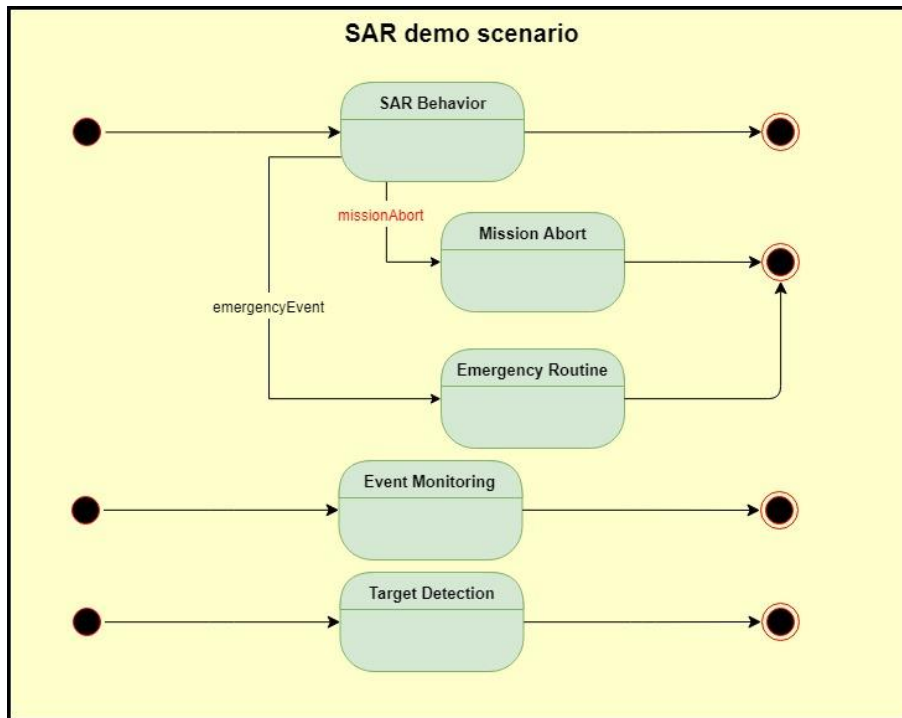


Figure 10 – State Machine for SAR scenario

6.3 Advantages of using swarm of drones and ground robots

Especially in SAR a single minute can decide between death and life. 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 with in 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 an «unmanned systems».
- Increased security for supervisory staff.
- Real-time interaction with Operations Center, better coordination during the inspection/SAR operations.
- Not only visual inspection by use of different sensors: audio analysis, presence of gas, thermal cameras etc.

7 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, in particular as regards 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 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.

8 Next steps

Some ideas that the CPSwarm consortium intends to carry out for the final demo M36 of the swarm SAR scenario are:

- a) At least 5 drones will start a coverage task with a new version of the coverage algorithm (to be discussed with LAKE).
- b) When a target will be found, the position will be communicated to the rovers (as in M18 demo).
- c) While one of the rovers is coming, the drone will perform a tracking task (we will have a movable target to show this functionality).

We are considering to add new features that will be demonstrated:

- a) Obstacles avoidance also for the rovers (not implemented yet).
- b) Drone simple obstacle collision avoidance (under testing).
- c) A drone tracking a target needs to come back home (battery low). A free drone (doing coverage) has to substitute it. The new drone will be selected autonomously by the swarm (under development).

9 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] Decawave DWM1001 ultra-wideband modules: <https://www.decawave.com/>
- [5] OpenMV Cam M7 camera: <https://openmv.io/>
- [6] PX4 flight stack: <https://px4.io/>

Acronyms

Acronym	Explanation
CPS	Cyber-Physical Systems
SAR	Search and Rescue
UAV	Unmanned Aerial Vehicle