# Low-cost solution in international robotic challenge: lessons learned by Tuscany Robotics Team at ERL Emergency Robots 2017

**Raffaele Limosani**
The BioRobotics Institute
Scuola Superiore Sant'Anna
Pisa, Italy
raffaele.limosani@santannapisa.it

**Alessandro Manzi**
The BioRobotics Institute
Scuola Superiore Sant'Anna
Pisa, Italy

**Alessandro Faggiani**
The BioRobotics Institute
Scuola Superiore Sant'Anna
Pisa, Italy

**Matteo Bianchi**
Dept. of Industrial Engineering
University of Florence
Firenze, Italy

**Marco Pagliai**
Dept. of Industrial Engineering
University of Florence
Firenze, Italy

**Alessandro Ridolfi**
Dept. of Industrial Engineering
University of Florence
Firenze, Italy

**Benedetto Allotta**
Dept. of Industrial Engineering
University of Florence
Firenze, Italy

**Paolo Dario**
The BioRobotics Institute
Scuola Superiore Sant'Anna
Pisa, Italy

**Filippo Cavallo**
The BioRobotics Institute
Scuola Superiore Sant'Anna
Pisa, Italy

## Abstract

European Robotics League (ERL) Emergency Robots is an outdoor robotics contest focusing on multi-domain emergency response scenarios. In this context, the deployed robots are expected to fulfill land, underwater, and flying cooperative tasks which emulate real-world situations inspired by the 2011 Fukushima accident.

Participation of the Tuscany Robotics Team at ERL Emergency Robots 2017 was deeply affected by damage to (and resulting unavailability of) the Unmanned Ground Vehicle (UGV) to be used in the competition. This damage occurred three days before the challenge. An entirely new working mobile platform was built from scratch using a low-cost chassis. It was able to compete with the other participants in the challenge and achieved various tasks ($\sim 83\%$ of the tasks completed by the best performer).

The paper provides a complete description of the work carried out and the system used during the competition. The development was based on low-cost solutions, in particular:

- Adoption of *open-hardware* and *open-software* technology, providing cheap and fast-prototyping solutions
- Use of *smartphones*, allowing sophisticated sensors to be exploited in an affordable way
- Employment of *Cloud* infrastructures, to reduce the computational burden of offloading heavy processes

Results of this experience represent a concrete demonstration, showing that even low-cost solutions can achieve complex tasks, such as those required for the European robotic competition. The presented lessons learned aim to provide guidelines useful for reacting to unexpected events. Finally, the team was honored with the Creativity Award *"for building a land robot from scratch in less than two days when their ground platform broke"*.

# 1 Introduction

Robotic competitions have been a successful means to draw attention to robotic research and to stimulate new topics and fields of research. They provide the opportunity to develop and demonstrate new technologies in various scenarios.

In 1992, Tom Dean and Pete Bonasso convinced the American Association for Artificial Intelligence (AAAI) to host a robot competition at the National Conference on AI. From this point, the AAAI Mobile Robot Competition and Exhibition was born. The event has endured and has become the oldest AI-centric robotic competition in the world (Balch and Yanco, 2002).

In 1997, the first RoboCup was held during the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97) in Nagoya, Japan, as part of IJCAI-97 special program. The Robot World-Cup Soccer event was an attempt to foster AI and intelligent robotics research by providing a standard problem in which a wide range of technologies can be integrated and examined (Kitano et al., 1997). Today, RoboCup is still present, and is encompassed by different leagues (RoboCup Soccer, RoboCup Rescue, RoboCup@Home, RoboCup Industrial, RoboCup Junior).

In 2004, the first competition of the DARPA Grand Challenge was held in the Mojave Desert region of the United States. The DARPA Grand Challenge was a prize-based competition for autonomous vehicles, funded by the Defense Advanced Research Projects Agency (Buehler et al., 2007). The second competition of the DARPA Grand Challenge was held in 2005, while the third competition, known as the "Urban Challenge", took place in 2007. From 2012 the DARPA Robotics Challenge focused on humanoid robotics[1].

The European Robotics League (ERL)[2] is funded by the European Union Horizon 2020 Programme. It is an ongoing framework for two indoor robotics competitions, ERL Industrial Robots and ERL Service Robots, and one outdoor robotics competition, ERL Emergency Robots.

One of the main purposes of each of those competitions is to be one step closer to a reliable deployment of robots in realistic conditions, particularly for disaster response. In fact, the real-world usage of unmanned vehicles in past events, such as the World Trade Center and the Fukushima Daiichi disasters, emphasized the deficiencies and pervasive challenges presented by such complex and dangerous task conditions (Murphy and Burke, 2005)(Nagatani et al., 2013).

As a result, among search and rescue robotics, the current number of active events and challenges is becoming respectable and involves participants ranging from young students to academic researchers. For instance, current competitions include the *UAV challenge*[3]; the *Urban Search and Rescue*[4], part of the West Virginia Robotics Alliance; the *RoboCup Rescue Robot League*[5]; the *Mohamed Bin Zayed International Robotic Challenge*[6]; and the aforementioned ERL Emergency Robots.

This paper describes the experience of the Tuscany Robotics Team during the participation at the ERL Emergency Robots 2017 in Piombino (Italy). In particular, it details the findings and the "lessons learned" obtained after the unexpected and irreparable breakage of the Unmanned Ground Vehicle (UGV) robot. The Team had to face this issue, building a low-cost prototype in less than two days to have the chance to compete in the challenge. In the end, the team ranked 4th among 7[7], using a significantly less expensive system compared to the opponents, without even performing preliminary on-field tests due to lack of time. Moreover, the team was honored with the Creativity Award by the referee committee.

[1] `http://archive.darpa.mil/roboticschallenge`, visited November 2017.

[2] `https://www.eu-robotics.net`, visited November 2017.

[3] `https://uavchallenge.org`, visited November 2017.

[4] `https://wvroboticsalliance.org/urban-search-and-rescue`, visited November 2017.

[5] `https://www.robocup2017.org/eng/leagues_rescue.html`, visited November 2017.

[6] `http://www.mbzirc.com`, visited November 2017

[7] `https://www.eu-robotics.net/robotics_league/erl-emergency/scoring-and-standings`, visited November 2017

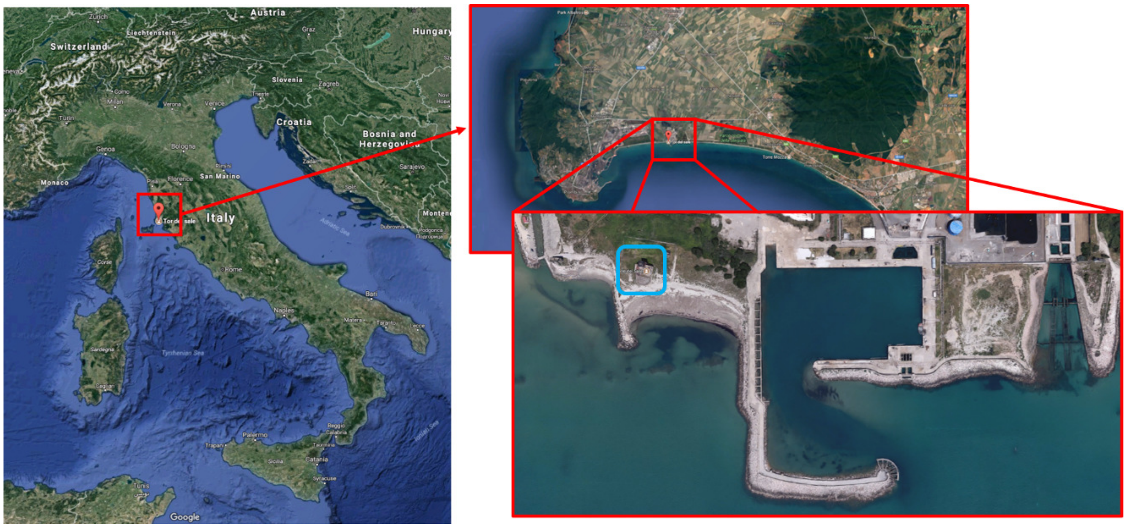


Figure 1: Google Earth view of ERL Emergency 2017 competition area (source: Google Maps).

### 1.1 ERL Emergency Robots

ERL Emergency Robots (Röning et al., 2017) (Winfield et al., 2017) is a civilian, outdoor robotics competition, with focus on realistic, multi-domain emergency response scenarios. Inspired by the 2011 Fukushima accident, the ERL Emergency Grand Challenge can only be overcome when land, underwater, and flying robots successfully cooperate. It is a continuation of the previously organized euRathlon challenge (Winfield et al., 2016).

The ERL emergency competition was held in the proximity of the Torre del Sale building in Piombino, Italy. The building lies on the shoreline, as shown in Figure 1 (highlighted with a cyan rectangle).

Unmanned Ground Robots (UGVs) operated near the building. In this area, the robots were faced with beach sand and static obstacles such as debris, stones, holes, and vegetation, with the possibility of rain, wind, and muddy terrain.

According to the competition rules, the expected UGV behavior can be divided into three missions.

**Mission-A: Search for missing workers.** UGVs had to search for two workers (mannequins) that were missing in the area inside and around the "nuclear plant" (represented by Torre del Sale). The workers that were missing had to be found as quickly as possible. One worker was missing inside the building, another one outdoors. Workers found during the first 30 minutes had a good chance to be rescued alive (which corresponded to achieving more points in the challenge). In the disaster setting, once found, the (two) missing workers would have required immediate first-aid assistance. For this reason, they had to be provided with a first-aid kit that must have been deployed near the workers.

**Mission-B: Reconnaissance and environmental survey.** The robot had to inspect the reactor building to evaluate the damage and find the safe path to the "machine room", in which the valves were located. This required robots to reconnoitre the area to provide situational awareness to the emergency response team. That was possible by transmitting live position (when GPS signal was available) and images/video to the control station during the run. Then, robots had to enter the building and find a safe path (a path that a ground robot can follow) to the machine room. An unobstructed entrance (at least 70 cm wide) had to be found first as well as a safe and unblocked path from the starting position of the ground robot. Robots had to create a floor (2D) or 3D map of the indoor part of the building.

**Mission-C: Pipe inspection and stemming the leak.** The cooling system was to use pipes that connected the reactor to the sea. After the staged earthquake and tsunami, these pipes might have been damaged, and substances might have been leaking from them. The valves that close and open the pipes were located inside the building in the machine room. If any leak or damaged was detected, the robots had to stem it by closing the correct valves.

The paper focuses on the development of a ground robot, but the missions are part of a broader context, where also air robots and underwater robots are involved to achieve tasks cooperating in a multi-domain scenario.

### 1.2 Aim of the paper

The participation of the Tuscany Robotics Team was significantly influenced by the irreparable damage (and resulting unavailability) of the Clearpath Husky UGV planned for use in the competition. This occurred three days before the challenge, during the first test session. Since a minimum of one land, one aerial, and one underwater robot was required to participate in the Grand Challenge, the team moved toward building a new mobile platform from scratch in less than two days to compete in the challenge.

The unexpected damage of the Husky UGV involved not only the building of a new ground robot, which demanded the planning of another design process in a rather short time, but required also a change in approach to the competition itself.

The purpose of this paper is to provide a complete technical description of the work conducted to create a fresh system on the competition field, in a short amount of time. From a technical point of view, the adopted solution, described in detail in the remainder of the paper, is based on the same pillars characterizing most of the related works for a low-cost platform, namely:

- Adoption of *open-hardware* and *open-software* technology, providing cheap and fast-prototyping solutions
- Use of *smartphones*, allowing sophisticated sensors to be exploited in an affordable way
- Employment of *Cloud* infrastructures, to reduce the computational burden of offloading heavy processes

The achieved results allowed the team to obtain fourth place in the overall standings and to receive the Creativity Award from the competition committee.

Finally, the results of this experience represent a concrete demonstration that quality of robotic technologies is growing fast, and low-cost solutions that can achieve favorable complex task scoring in the European robotic competition are available.

The paper is organized as follows:

- Additional low-cost solutions for robotic applications are presented in Section 2.
- Overall developed system is detailed in Section 3.
- Results, in terms of successes in the competition tasks, are reported in Section 4.
- *A posteriori* discussion with lessons learned is presented in Section 5, and the conclusion is given in Section 6.

## 2 Related work: low-cost robotic solutions in literature

Analyzing the current state of the art, several research papers focus on the interaction between economic matters and robotics. These papers cover heterogeneous fields of application, ranging from experimentation to teaching, and including ground and marine areas.

This section provides an overview of literature research that presents working robotic solutions under economic constraints.

In 2013, McLurkin et al. presented a low-cost robot design that enables large-scale multi-robot research, innovative new curriculum, and multi-robotics outreach to younger students of K-12 schools (McLurkin et al., 2013). The aim was to create an inexpensive and sophisticated robot which could be affordable for everyone to enable its usage among students and other robotics enthusiasts. The system was based on four main parts: the *r-one* robot, a Python development environment, a camera tracking system for ground-truth localization, and server software to connect all the pieces together. The *r-one* robot had a cost of approximately $220 per robot and contained a full sensor suite including a gyro, accelerometer, wheel encoders, and light sensors.

The educational target was also the focus of the work presented in (López-Rodríguez and Cuesta, 2016). Their work describes an open, low-cost (35 euro), modular and extendable mobile robot based on Android and Arduino to be used as an educational tool in labs and classrooms of information and communications technology (ICT). The final aim of the authors was to introduce the "BYOR: Bring your own robot" education policy equivalent to "BYOD: Bring your own devices" in the computer world.

This research was based on two popular trends: the use of *open-hardware* and *open-software* technology, represented in this case by the Arduino project; and the tremendous growth of the *smartphone* market, which provides sophisticated devices that are affordable to everyone.

The resulting platform, called "Andruino-A1" (see Figure 2), was designed to be used in introductory robotics courses in a vocational education school, with 20- to 42-year-old students, as a part of their curricular activities.

Concerning the marine environment, in (Valada et al., 2014) a low-cost multi-robot autonomous platform (CRW: Cooperative Robotic Watercraft) for a broad set of applications, including water quality monitoring, flood disaster mitigation, and depth buoy verification, was outlined. The concept was based on a combination of Wi-Fi and 3G, to improve communication reliability while maintaining a low development cost for each craft, below $800. On the other hand, concerning the air domain, performances of low cost Micro Aerial Vehicle platform AR-drone ($\sim$ $150) was investigated in (Saska et al., 2012), while in (Gageik et al., 2015) authors focused on low-cost sensors to achieve obstacle detection and collision avoidance for an Unmanned Aerial Vehicle.

A key feature of low-cost solutions is undoubtedly the use of remote or Cloud resources. In (Mohanarajah et al., 2015b) the authors presented architecture, protocol, and parallel algorithms for collaborative 3D mapping in the Cloud with affordable robots. The low-cost (< $600) platform consisted primarily of a differential drive base (iRobot Create), an RGB-D sensor (PrimeSense), an ARM-based single-board computer (ODROID-U2), and a dual-band USB wireless device. All processes were managed by Rapyuta (Mohanarajah et al., 2015a), a Cloud robotics framework that runs in a commercial data center.

## 3 System

The new system was built upon the low-cost chassis lent by the ENSTA Bretagne team, with adaptation of software modules already developed.

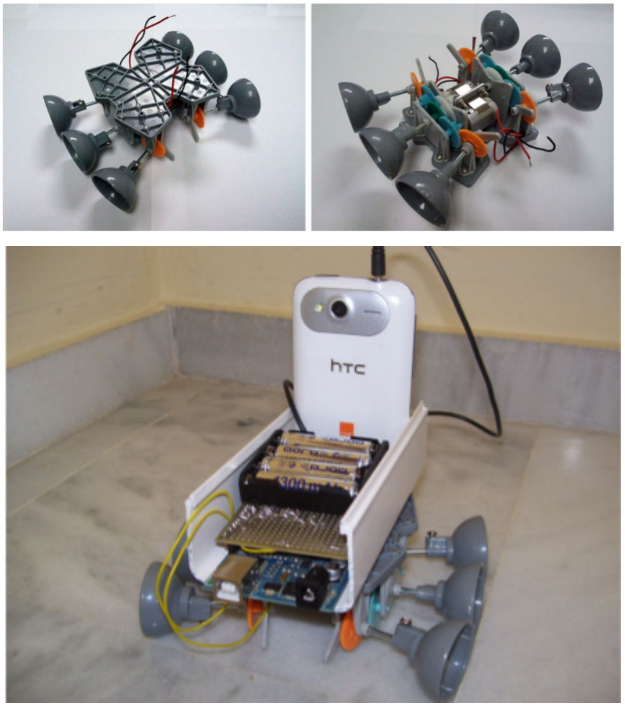


Figure 2: Andruino-A1 robotic base (top) and Andruino-A1 mobile robot (bottom), source: (López-Rodríguez and Cuesta, 2016).

The resulting platform QUOKKA (QUick Operative Kiddie KArt, name inspired by the homonym animal[8]) is a remotely-controlled mobile platform, depicted in Figure 3. It is composed of several modules: chassis, Intel NUC, Arduino board, webcam, smartphone, and Li-On batteries.

The remainder of this section details the hardware, firmware, and the entire software architecture of the system.

### 3.1 Hardware and firmware

The developed low-cost prototype was built on the GM 4WD Crawler Punisher chassis, along with some assembled components of the original ground vehicle.

Figure 4 reports an overview of the connection scheme adopted to build the platform. In the following sections, a detailed description is given for each component.

#### 3.1.1 GM 4WD Crawler Punisher platform

The low-cost chassis, provide by the ENSTA Bretagne team, is a commercial product from Graupner/SJ GmbH. It is typically used as a mobile base for off-road remote-controlled vehicles. Composed of a metal chassis, four steering wheels, and a suspension system, the base proved suitable for the different terrain types encountered during the competition. The structure (center-right in Figure 4) is equipped with two RC 390 brushed motors to provide a four-wheel-drive system, a regulator with cooling fan, and two GM S3660 servos for steering functions.

[8] **Quokka** (Setonix brachyurus) is a small macropod about the size of a domestic cat.

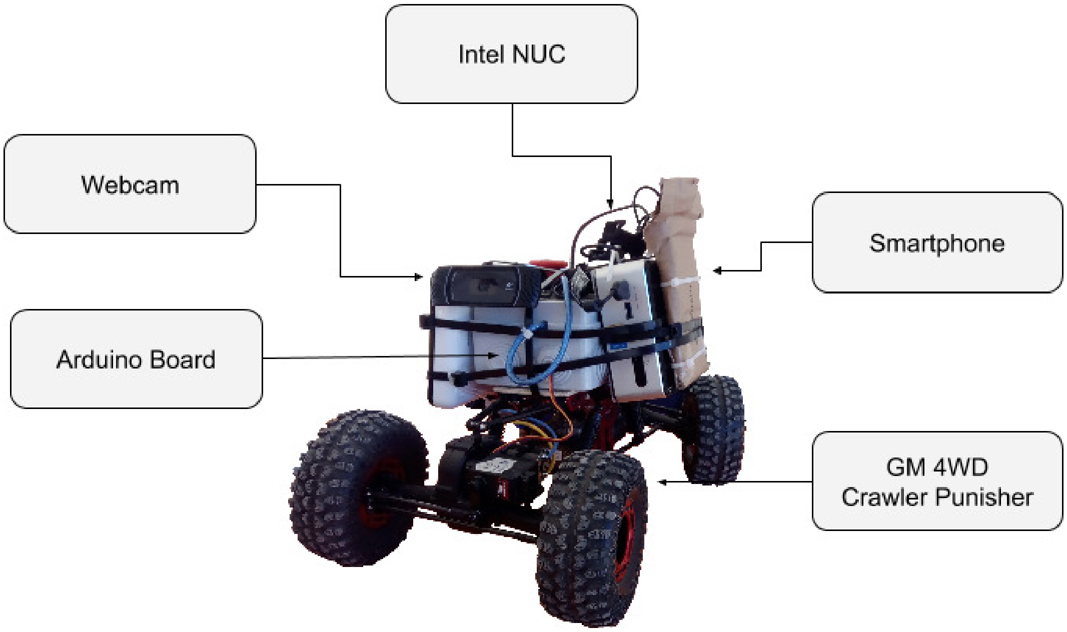


Figure 3: The QUOKKA robot, the low-cost platform built in two days during the ERL competition.

### 3.1.2 Arduino firmware

The low-level motor control is implemented on an Arduino MEGA, which processes incoming commands through a serial communication to obtain basic movements. The firmware was developed with the aim to have a reliable working module in a short amount of time. Thus, it was developed under strict constraints of reduced development and testing time. For this reason, its architecture was intentionally kept as simple as possible. The firmware continuously listens to the serial port, executing one of eight distinct functions. Specifically, *forward* and *backward* control the brushed motors, *left* and *right* use the servos to steer the wheels, *increment* and *decrement*, and *stop* control the speed, and the last function acts to *straighten the wheels up*. Additionally, as a low-level safety mechanism, the commands must be sent at least at 1 Hz; otherwise, the motors are stopped.

### 3.1.3 Vision and communication

The LOGITECH Webcam HD PRO C920, previously used on Husky UGV, is mounted in front of the chassis to provide the real-time video streaming used for teleoperation feedback and image analysis. The position is fixed and orientated to give the best field of view regarding both the floor and the building wall. A smartphone (Lenovo Vibe P1m) is used to communicate with the ground station through the 4G protocol. This technology allowed operation of the robot all along the competition field, both outside and inside the building. It is worth noting that all the other teams[9] used expensive WiFi antennas to cover the entire area, and they experience frequent connection problems and interference. This solution was already discarded by our team after the first day of on-site testing. The use of the 4G technology allowed us to deploy a communication infrastructure, which was reliable, fast, and cheap and avoided the need to modify the environment with the installation of additional networking devices. Thus, we prevented the connection problems experienced by the other teams. In addition, the smartphone was used not only to connect the platform to the system but also to provide and send the current GPS location of the robot to the ground station.

[9] The only exception was the Telerob team that used an expensive proprietary system based.

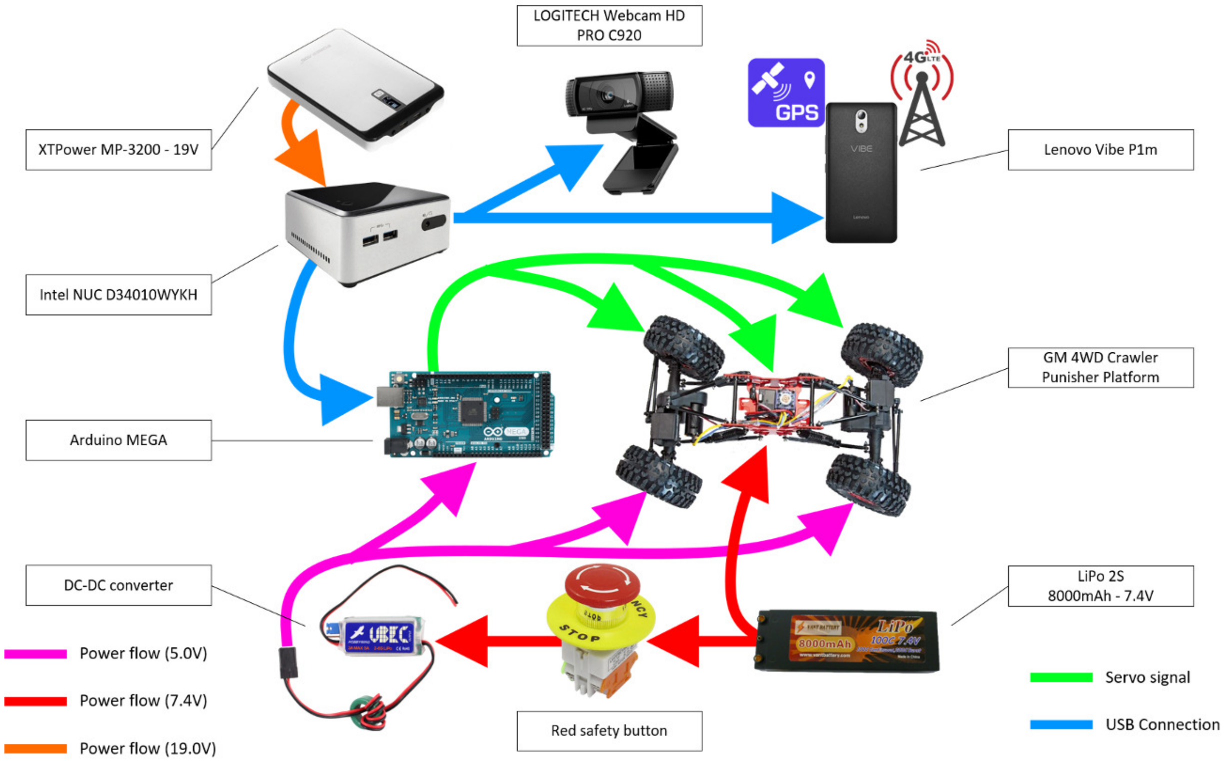


Figure 4: Connection scheme for QUOKKA components.

### 3.1.4 Computing system

The computing system is represented by an Intel NUC D34010WYKH, equipped with an Intel Core i3-4010U processor, dual rear-panel USB 3.0 ports, a 19 V(65 W) DC power connector, and a mechanical chassis size of 116.6 mm x 112.0 mm x 49.5 mm. Ubuntu 16.04 was chosen as the operating system, along with the ROS (Robot Operating System) framework (Quigley et al., 2009) due to its flexibility and easy communication module interface, including a large collection of tools and libraries. Thanks to its portability features, the adoption of ROS allowed us to reuse part of the software already developed for the damaged UGV. The USB 3.0 ports were used to connect the Arduino MEGA and manage the serial connection, webcam, and mini USB adapter for the smartphone.

### 3.1.5 Battery

The entire system is powered by two portable batteries. A LiPo 2S, 8000 mAh 7.4 V battery is used for the brushed motors and servos, while an XTPower MP-32000 with an output of 19 V/4.5 A is connected to the NUC, which also powers the Arduino and the webcam through the USB 3.0.

## 3.2 Software architecture

The software architecture of the entire system includes the QUOKKA robot, ground station, and remote Cloud resource (see Figure 5). The architecture, already presented in (Manzi et al., 2016), is designed to have real-time performance and interoperability features. In fact, it relies on the WebSocket protocol and on the ROS framework. The first provides a full-duplex communication channel over the HTTP suitable for real-time data transfer and is widely supported by modern browsers that can be easily used as graphical

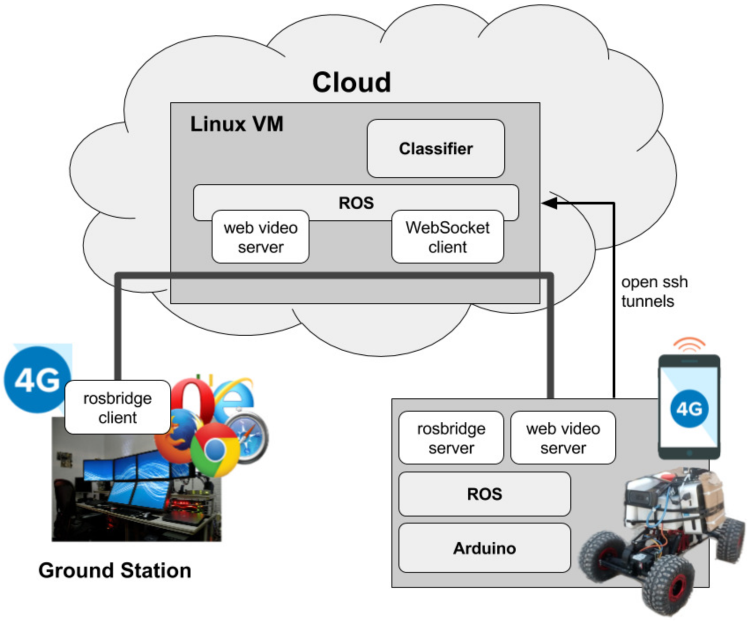


Figure 5: Software architecture of the complete system. It includes the QUOKKA prototype, ground station, and remote Cloud resource.

front-end. The latter represents the *de facto* standard robotic middleware.

In this system, each of the three components operates on a different network. While the Cloud virtual machine has a public IP address, both the ground station and the robot are not directly accessible. Generally, a common solution to solve this issue is to set up a virtual private network (VPN), which extends a private network across the public network. However, establishing a VPN is not a straightforward operation, and all the agents that want to access the system must be set up with it. To overcome this limitation, this architecture uses the "ssh tunneling" mechanism, which allows forwarding of a TCP connection only (Manzi et al., 2016). It has to be underlined that the solution described in this paper is still based on configuration methods, as in the case of VPNs configuration, rather than on implementation of new software. The rationale beyond this choice is based on the aim to achieve a reliable system starting from mature technologies already available.

### 3.2.1 QUOKKA software

The mobile platform runs a minimal set of ROS nodes to keep the system light and fast. Details of these modules are provided below.

**Arduino controller:** As anticipated in Section 5.4 and summarized in Table A1, the autonomous navigation software was not installed on the new prototype, due to the limited time and lack of sensors (i.e. encoders and laser scanner). Therefore, according to ERL Emergency Robots 2017 rulebook, the teleoperated mode was selected as the only viable solution.

As described in Section 3.1, Arduino firmware was implemented to manage information received on the serial port, which interfaced with the motors. The connection between the ROS framework and Arduino was implemented upon ROS *serial*[10] package. Basically, serial commands (namely chars) were constantly sent at a fixed 5 Hz frequency. The code adheres to the ROS *geometry_msgs Twist* message standard to integrate the platform with the ROS ecosystem.

[10] http://wiki.ros.org/serial, visited November 2017.

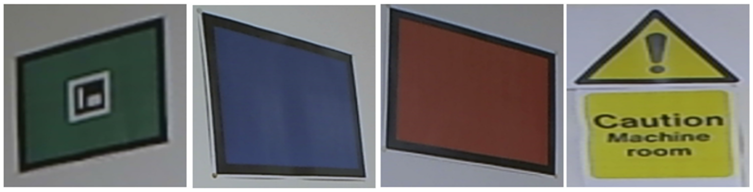


Figure 6: Dataset samples used to train the classifier during the ERL competition. From left to right: *free door*, *blocked door*, *damage*, *machine room.*

**Rosbridge server:** Provides the WebSocket transport layer, allowing web-pages to communicate with ROS through JSON messages.

**Web video server:** Provides a video stream of ROS image transport topic that can be accessed via HTTP.

**Camera:** Interfaces with standard USB cameras and publishes ROS images, allowing compressed images as well to reduce the bandwidth usage.

To solve the aforementioned visibility issue between different networks, a dedicated program executes a "reverse ssh tunneling" on the Cloud virtual machine. This solution allows the robot simply to connect to the Cloud, without becoming part of its network. Hence, the rosbridge and web video servers have their own "tunnels", which are accessible from other networks with Internet access.

### 3.2.2 Cloud resource

The remote resource was implemented on the Microsoft Azure Cloud, running a Linux Virtual Machine (VM) equipped with Ubuntu 16.04 and the ROS framework. The VM at our disposal was a low-power machine with one core and 0.75 GB RAM. The role of this Cloud Resource is twofold. First, it acts as a bridge for the different networks, namely the 4G connections of the robot and the ground station, providing the tunnels for communication. Second, it runs the OPI (Objects of Potential Interest) classifier. Hence, the VM contains two modules: the ROS node classifier and a WebSocket client written in Python to handle ROS images.

The classifier is a basic Convolutional Neural Network (CNN), which implements an architecture similar to the LeNet-5 (LeCun et al., 1998). The choice of such a simple architecture is driven by the fact that the competition requires recognition of only four classes, which also enables use of a relatively small dataset for the training phase. The CNN is implemented using Keras (Chollet et al., 2015) with Tensorflow (Abadi et al., 2015) as a backend. The network takes as input a 180x180 RGB image and has a total of four layers: two convolutional layers and two feed-forward fully connected layers (see Table 1 for more details). The training of the network was conducted by exploiting a dataset containing 160 images (40 for each class) which used a $k$-fold validation strategy. The result on the test set, containing 32 samples, produced an accuracy of 0.96. The classifier is encapsulated in a ROS node, which takes as input the compressed image from the ROS topic, resizes it with a proper resolution, and sends it as input to the network. Then, the classifier node publishes the classification result on two topics, both the resulting string and the processed image with the recognized class on top of it. Figure 7 shows two classification examples that occurred during the final competition.

### 3.2.3 Ground station

To improve the usability of the operator charged to monitor and manage the robot during the challenge, a ground station laptop was equipped with a web-based GUI that was used to control the platform remotely.

The ground station GUI was implemented on the ROS package *rosbridge_suite*[11] and shows both the video

[11] http://wiki.ros.org/rosbridge_suite, visited November 2017.

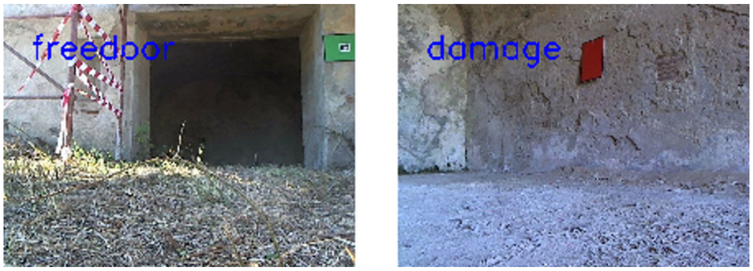


Figure 7: Classification examples during the final competition. The *free door* sign (left) outside the building and the indoor *damage* sign (right).

Table 1: The convolutional neural network architecture used for the OPI classification.

| Layer | Description |
|---|---|
| Input | 180x180x3 RGB image |
| Conv | 32 dim, 3x3 kernel |
| ReLU | |
| Conv | 64 dim, 3x3 kernel |
| ReLU | |
| MaxPool | 2x2 kernel, 2 stride, 0 pad |
| Dropout | 0.25 rate |
| Fully-Connected | 128 dim |
| ReLU | |
| Dropout | 0.5 rate |
| Fully-Connected | 4 classes |
| Softmax | |

streaming from the robot camera and data concerning robot operation, including GPS localization provided by the smartphone. The adoption of web protocols allows the ground station GUI to run on a wide range of devices, and it uses a set of JavaScript libraries providing a convenient abstraction to the core ROS functionality.

In the current implementation, the user can move the robot using a computer keyboard, sending velocity commands; other types of interfaces can be further implemented. In addition, the user can dynamically adjust the quality and dimension of video streaming, improving the feedback according to the current quality of the connection. During the challenge, for most of the time, the video streaming was gathered from the ROS topic `compressed` with a 5% quality and image size of 320 width and 240 height. Figure 8 reports a screen-shot of the implemented web-based GUI.

## 4 Results

This section details the results obtained by the land robot during the ERL Emergency Robots 2017 Grand Challenge. It is worth noting that the QUOKKA platform was never tested on the competition field during the dedicated test session days. Thus, the final challenge was its first trial. Video of the robot in the competition is available at `https://youtu.be/59eoJMRC2Sg`.

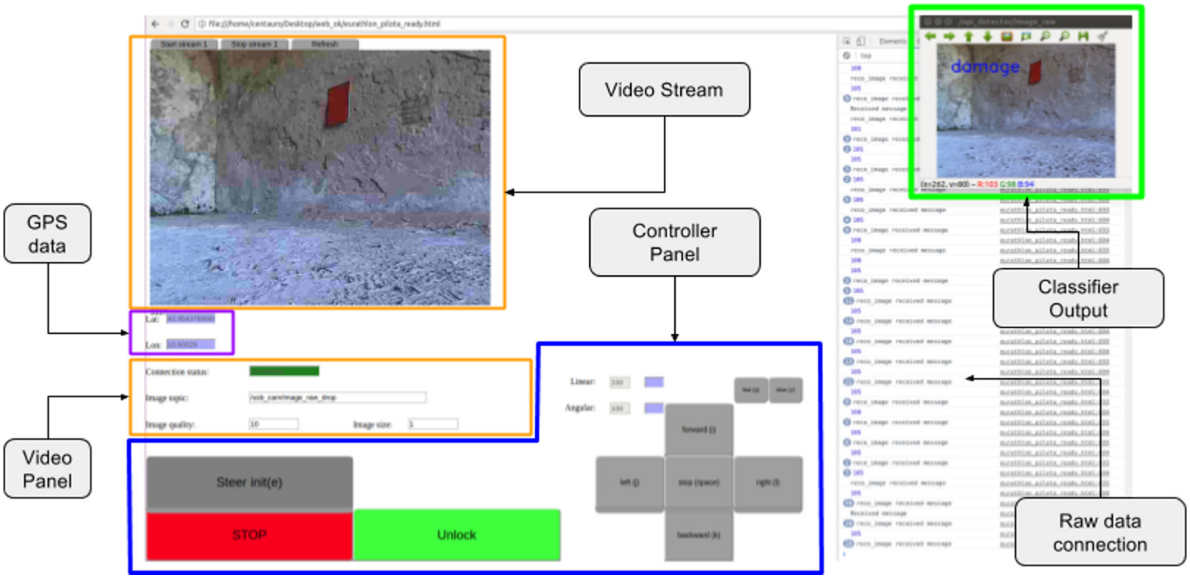


Figure 8: The ground station web-based GUI implemented for the ERL competition.

The tasks, limited to the land robot, evaluated by the referees are reported in their entirety in Table A2[12]. In particular on the aspects of cooperation among the different domains involved in the competition, this was performed as Human-Human communication (operators of different domains shared information through walkie-talkies). The competition rules allowed for direct robot-robot communication, however this was not used due to the lack of time for tests before the competition.

Even if the communication described infrastructure allowed a robot-robot communication, this was not used due to the lack of tests before the competition.

Analyzing the Referee Sheet, for 28 available tasks, the QUOKKA platform successfully completed 14 tasks (success rate: 50%). The best performance for a land robot in the competition was 23 of 28 (success rate: $\sim$ 82%).

Furthermore, the implemented low-cost solution was one of the four teams (among seven) that were able to:

- Enter the structure (*robots find a safe and unobstructed path to the unblocked entry of the building for a ground robot*; *from the starting point, a ground robot follows a safe path to the unobstructed building entrance*; *a ground robot enters the building through the unobstructed door*)
- Enter the machine room (*from the building entrance, a ground robot follows a safe path to the machine room*; *a ground robot enters the machine room*)
- Perform real-time recognition automatically (*a robot localizes the unobstructed entrance in real-time in automatic way*)
- Localize all four OPIs related to obstructed entrances and damages on indoor walls (*robots localize the obstructed entrances*; *the ground robot(s) recognize the damages on the wall of the building*)

In a deeper analysis, it can be highlighted that, considering the actual hardware configuration without laser sensors and manipulator, only 19 tasks can be accomplished concretely (success rate $\sim$ 74%), namely. In particular, the tasks that cannot be achieved were:

[12] Official results are available at `https://www.eu-robotics.net/robotics_league/upload/documents-2017/TBM1-ERL-ER-2017-results.pdf`, visited November 2017.

- *A ground robot reaches the WayPoints within a precision of 3 m in autonomous navigation*; not possible due to lack of navigation sensors (laser, odometer, etc.)

- *A ground robot deploys the first-aid kit (within radius 1 m) from the worker inside the building*; carriage of first-aid kit not possible due to the reduced size of QUOKKA platform

- *The ground robot(s) builds a geometric indoor map of the building*; not possible due to lack of mapping sensors (laser)

- *A ground robot closes the correct valve. The robot must close one valve of the set autonomously and the other one manually*; not possible due to lack of manipulator

- *The ground robot and the underwater robot close the correct valves in a synchronized process*; not possible due to lack of manipulator

The best performance on the same available tasks was 18 of 19 (success rate: $\sim$ 94%). In other words, compared with the best available results, the solution developed here was not able to:

- Find the missing worker inside the building (*within 30 minutes of start of the run, a ground robot reports the correct location of the missing worker inside the building*), accomplished by only one of the other ERL competitors

- Autonomously recognize the missing worker inside the building (*the missing worker is detected in real-time in an automatic way*), accomplished by only one of the other ERL competitors

- Autonomously recognize the machine room sign (*a ground robot recognizes the machine room sign in real-time and in automatic way*), accomplished by only one of the other ERL competitors

- Recognize the ID of the correct set of valves (*a ground robot recognizes the ID of the correct set of valves in the machine room*), accomplished by only two of the other ERL competitors

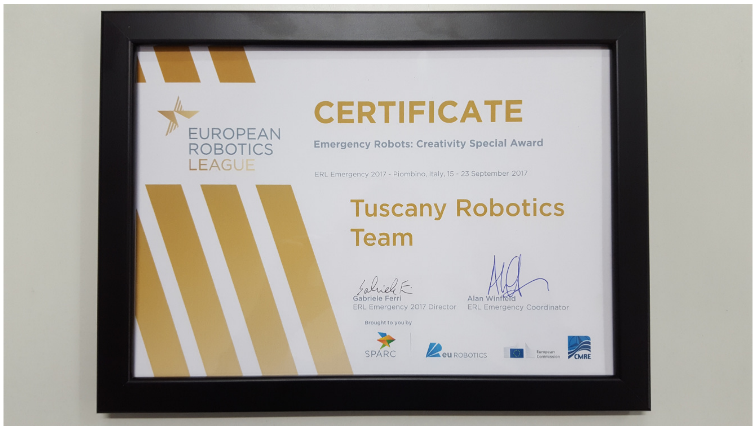
- Return to the starting point (*the ground robots return to the landing area once all the tasks have been done*), accomplished by all the other ERL competitors

As a final result, the team ranked fourth among seven teams, with a difference of two points with respect to the third place. The ranking is based on the overall performance in the three domains (Sea, Land, Air).

Furthermore, the Tuscany Robotics Team was honored with the Creativity Award, given by Prof. Bernd Brüggermann from Fraunhofer FKIE, "*for building a land robot from scratch in less than two days when their ground platform broke*" (see Figure 9).

## 5 Lessons learned

The lessons learned range, from technical solutions to planning methodologies and human experiences. In the following sections, some guidelines useful when reacting to unexpected events are provided. In particular, it is evident that tests conducted in a laboratory environment are not always enough to face the issues that arise in the real world. Hence, thinking from a new perspective, outlining a strategy according to the available resources, and designing a system with adaptability features are fundamental aspects to achieve of sound results.



EUROPEAN
ROBOTICS
LEAGUE

CERTIFICATE

Emergency Robots: Creativity Special Award

ERL Emergency 2017 - Piombino, Italy, 15 - 23 September 2017

Tuscany Robotics
Team

Gabriele E.
Gabriele Ferri
ERL Emergency 2017 Director

Alan Winfield
ERL Emergency Coordinator

Brought to you by

SPARC | eu ROBOTICS | European Commission | CMRE

Figure 9: The Tuscany Robotics Team was honored with the Creativity Award "*for building a land robot from scratch in less than two days when their ground platform broke*".

### 5.1 "Thinking outside the box"

During the competition, the Tuscany Robotics Team had to face irreparable damage to the UGV platform, and thus had to find a way to address the situation by thinking from a new perspective. The unexpected condition forced out team to look for solutions that were not part of the initial master plan.

The concept of "thinking outside the box" proposes a restatement of the solution strategy: the heart of the matter is the unspecified barrier that people typically perceive. Specifically, it concerned designing a new platform in a short amount of time, trying to optimize the available means, "reshuffling" of resources, and searching for and inserting new elements:

- Low-cost chassis lent by another team participating in the competition
- Replacement of GPS and connectivity hardware with a smartphone, a device available in every team member's pocket
- Shift of computational power on a remote infrastructure, reducing the computational costs of the limited onboard device

The lessons learned in approaching a realistic robotic competition, and, more generally, deploying a system in real conditions, can be summarized in "expect the unexpected" and always be ready to face adversity, and avoid discouragement and giving up by "thinking outside the box".

### 5.2 Platform-based strategy

The methodology behind the participation in a robotic competition should always include the definition of a detailed master plan that the human operator of the robot must follow. In this strategy, the order of the tasks to accomplish must be defined. Particularly, for each task, it is important to consider the weight of the system (points from the rulebook) and the reliability of all the platform features involved in that particular task.

The use of a new platform with different and limited features meant that the tactics had to be replanned. This analysis, which occurred before the competition, allowed the operator to clarify the strategy to adopt and to optimize the successful tasks. It is worth noting that the QUOKKA platform was never tested on the competition field prior to the competition, while the other teams had three testing days.

### 5.3 System adaptability

In addition to the resiliency demonstrated by the team to manage the unexpected events, the obtained results were achieved thanks to the adaptability of the developed system. Specifically, it was conceived by focusing on three main features that made the system more flexible to dynamic changes:

**Dynamic settings of video streaming:** The ground station GUI permits the operator to adapt the quality and size of the video streaming to manage possible network instabilities introduced by a moving platform.

**Machine learning classifier:** The use of a CNN allows use of a classifier can generalize having good performance despite the unknown samples. In fact, even with the unavailability of a dataset built with the competition samples due to the missed test days, the classifier was able to recognize the OPIs (see Figure 7).

**Software interoperability:** The use of the ROS middleware led to hardware abstraction, which was a key element for quickly reusing the developed software on a different platform with fewer efforts. Furthermore, the adoption of the *rosbridge* library for the communication between the machines of the system (see Section 3.2), facilitated the migration of the classifier from the original robot to the remote resource. As a matter of fact, the abstraction performed through the use of ROS framework was the key to reuse most of the software already implemented, as detailed in the next section.

**Communication infrastructure:** The use of communication architecture presented in (Manzi et al., 2016) based on mature third party technologies, such as mobile 4G communication and WebSocket, simplified the change of mobile platform. Other solutions, such as WiFi meshes or networks, would have been significantly compromised or would have requested a significant effort on reconfiguration, while the abstraction performed through 4G communication resulted in a substantial advantage when dealing with hardware changes: basically, the only requirement to reach and control the mobile platform was a working Internet connection. It has to be highlighted that the communication infrastructure was investigated and implemented before the breakage of the UGV prepared for the competition: actually, the abstraction performed through the use of reverse tunnelling and WebSocket technologies was one of the central reasons beyond the chance to participate with the new ground system.

The lesson learned is that “moving from labs to the real world” always introduces aspects that are difficult to foresee. Because of this, the development of a system which is easily reconfigurable *on-the-fly* with adaptability and interoperability features represents a fundamental aspect which should not be underestimated.

### 5.4 Reshuffling of resources

After the breakage of the main UGV, the team investigated the reason for the fault, and found an electrical failure that was impossible to repair during the competition (CAN motor board damage). Consequently, a brainstorming activity to analyze the situation was conducted. Subsequently, the team decided to build a totally new platform from scratch to continue in the competition. Due to the limited amount of time and resources, the only feasible solution was to make a low-cost prototype. In a display of good sportsmanship, a lightweight chassis was lent by the ENSTA Bretagne team. Thus, an analysis of the resources still available and reusable was performed. The result of this reshuffling process is summed up in Table A1 in the Appendix.

It is worth noting that the damage of a fundamental UGV component compromised the use of most of the original hardware (i.e. motor board and motors). Furthermore, the use of a lightweight and low-cost chassis made reshuffling of several devices, such as the UR5 robotic arm, SICK and Velodyne laser scanner, and Novatel GPS, impossible, mainly due to their weight and power consumption.

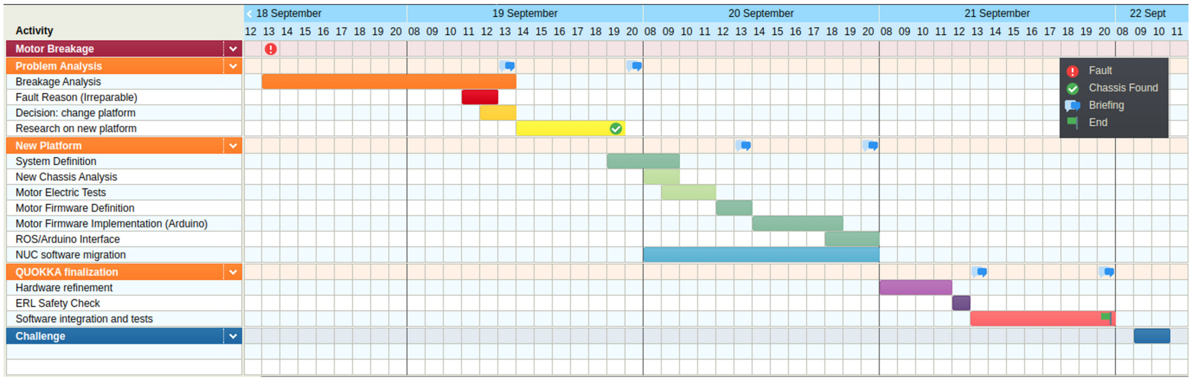


Figure 10: Gantt chart used to represent the activity schedule to perform as a reaction to the unexpected breakage.

Nevertheless, lighter hardware components became the basis for the new low-cost platform. For instance, the Intel NUC, planned to be used as support to the main computational device, turned into the new control unit, powered by the Li-polymer 32000 mAh battery, originally used as complementary power supply for additional devices (e.g. amber emergency light).

On the other hand, most of the developed software (i.e. low-level teleoperation control, web-based GUI, object classifier, video streaming module, and communication infrastructure) was completely reused without (or with few) modifications and with minimal effort. Exceptions involved the software modules for unusable hardware, such as mapping and autonomous navigation based on laser measurements.

## 5.5 Scheduled bottom-up development

The decision to build a new prototype from scratch with limited time and resources required careful definition and scheduling of the work plan to closely monitor the entire process. A Gantt chart is shown in Figure 10. The chart details all the steps that occurred during the competition days. Note that meal breaks were conceived as "milestones" for internal briefings and fast brainstorming activity.

Basically, the work plan was organized as a pipeline, where the activities were serially performed, except for the high-level software migration done in parallel with the hardware/firmware development. This was primarily due to the dearth of human resources and to the prerequisites between activities (e.g. motor firmware implementation could not start before the end of necessary electrical tests). Furthermore, it is worth noting that the development process followed a bottom-up approach. This is driven by the fact that the approach is particularly suitable because the new system starts from existing modules. In fact, the re-usability of code is one of the main benefits of the bottom-up approach, which emphasizes also the coding and early testing (Gamma et al., 1995).

After two days, the new platform passed the competition safety check, and the entire system, including ground station and Cloud infrastructure, was further refined and tested. Unfortunately, no time remained to exploit early competition test sessions. Therefore, the prototype was directly used in the final challenge. It is important to emphasize this aspect in light of the obtained results.

## 5.6 Low cost intervention

The QUOKKA platform was developed in response to an unexpected event in a short amount of time (less than two days), and was based on the reshuffling of available resources. As a direct consequence, the resulting solution was characterized by the use of (relative) low cost elements. This section outlines a cost evaluation of the prototype, providing an optimization of the adopted hardware using currently available commercial devices.

**Chassis:** The cost of the GM 4WD Crawler Punisher is 168.99 €[13]. Cheaper solutions can be found on the market; however the employed chassis represents an acceptable trade-off between cost and semi-professional performances. Similar solutions provided by the same manufacturer can offer more powerful devices but at higher costs[14].

**Motor control board:** An Arduino MEGA (cost ∼18.99€) was used for the competition. A lee expensive microcontroller can be used, but this Arduino device offers a reliable open-source solution at a reasonable price.

**Computing system:** The Intel NUC equipped with an i3 processor, available from the original UGV, was used (cost ∼200€) during the competition. Considering that the new software architecture (see Section 3.2) was rebuilt to be light and fast, low-power and less-expensive devices can be used for this purpose. Particularly, only a few ROS nodes must run on the main computing system; it requires a

1. Node that subscribes to `cmd_vel` topic, sending commands (i.e. chars) on the serial port
2. Node to gather images from a web-cam
3. *Rosbridge server* for communication
4. *Web video server* to stream the video

The use of a *Raspberry Pi 3*[15] (cost ∼35€) is adequate for this aim. An additional advantage of using this device compared to the NUC is the lower energy consumption. This setting enables removal of the XTPower MP-32000 battery, saving ∼140€.

**Battery:** Vant LiPo battery 8000 mAh 100C 7.4V [16] (cost ∼70.00€) was used to power the brushed motors and servos. Removing the Intel NUC, this battery can also be used to power the main computer system.

**4G connectivity:** During the competition, a team member's low-cost smartphone (∼150€) was used to provide 4G connectivity to the platform. Nevertheless, most telecommunications companies offer a tariff plan including a portable device at approximately 10€ per month.

**GPS device:** The sensor included in the smartphone was used during the competition to provide the current position of the robot remotely. The least expensive solution that can be found on the market is the GlobalSat USB GPS Receiver (∼24.95€[17]).

**Remote PC / Cloud infrastructure:** The Microsoft Azure Cloud was used as provider of virtual machines. The cost of a standard instance (1 core, RAM: 3.5GB, Hard Disk: 50GB) is negligible (0.125€/hour)[18].

The optimized costs for the QUOKKA platform are reported in Table 2. The overall cost of the developed prototype is approximately 320€. Obviously, the detailed costs of the other UGV participants cannot be provided. However, from a qualitative analysis, the costs of the original Husky platform, which is approximatively 50000€ , including arm and sensors, can be considered in the average of the competition opponents. In conclusion, the low-cost QUOKKA platform was able to achieve 19 of the 23 tasks achieved as the best result of the competition (∼ 83%), at a cost two orders of inferior size of magnitude less than the competition.

[13] `www.graupner.com/GM-4WD-Crawler-Punisher-XXS-Electro-1-24-Ready-to-Run-RTR/90124`, visited November 2017.

[14] `www.graupner.com/Models/Cars/Offroad`, visited November 2017.

[15] `www.raspberrypi.org/blog/raspberry-pi-3-on-sale`, visited November 2017.

[16] `www.vantbattery.com`, visited November 2017.

[17] `www.usglobalsat.com`, visited November 2017.

[18] `https://azure.microsoft.com/it-it/pricing/calculator/`, visited November 2017.

Table 2: Evaluated costs of a QUOKKA prototype

| Item | Costs During ERL (€) | Optimized Costs (€) |
|---|---|---|
| Chassis | 168.99 | 168.99 |
| Arduino MEGA | 18.99 | 18.99 |
| Intel NUC | 200 | - |
| Raspberry Pi 3 | - | 35 |
| LiPo 2S Battery (7.4 V) | 70 | 70 |
| XTPower Battery (19 V) | 140 | - |
| Smartphone | 150 | - |
| 4G connectivity | 2.5/week | 2.5/week |
| GPS | - (Smartphone included) | 24.95 |
| Cloud infrastructure | - (negligible) | - (negligible) |
| Total | ∼750 | ∼320 |

Furthermore, as demonstrated by the present ERL experience, the integration of robotics with existing information technologies, such as the Cloud and the 4G infrastructure, represents a concrete and viable solution that produces high-quality results and lowers the total cost.

# 6 Conclusion

Participation in international robotic challenges represents a valuable and unique opportunity to move academic research from the lab to real life, to compare a particular developed technical solution with results achieved by other international researchers, and to draw the attention of a non-technical audience to the latest results and issues in the robotic community.

The team developed a low-cost platform in less than two days and in a short amount of time, reshuffling the available means and replanning the overall competition strategies.

Concerning the competition results, the work conducted has demonstrated how a low-cost solution can achieve a level of performance comparable with expensive solutions (the team ranked fourth among seven opponents, achieving $\sim 83\%$ of the tasks accomplished by the best performer). In particular, the use of the remote Cloud resource and the 4G connection represented a completely different approach to the competition, and enabled drastic reduction of costs while improving the reliability of the overall system.

Finally, from a personal and subjective point of view, despite the short duration of the event, the knowledge and experience gathered during ERL Emergency Robots 2017 cannot be easily assimilated through other forms of professional growth (classroom lessons, research in the lab, academic conferences, etc.). In other words, the participation in international robotic challenges is a worthwhile activity that should be promoted and supported in any research department, whether academic or industrial, and public or private.

**Acknowledgments**

The authors would like to thank Prof. Bernd Brüggermann from Fraunhofer FKIE, for the award honored and for the moral support provided during the competition; ENSTA Bretagne team, for their good-sportsmanlike behavior during the entire event; Dr. Gabriele Ferri and Dr. Fausto Ferreira from NATO STO CMRE for the wonderful opportunity they provided to researchers by organizing the ERL Emergency competition; finally,

the team responsible for ERL Emergency within the EU H2020 project RockEU2.

## References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Balch, T. and Yanco, H. (2002). Ten years of the aaai mobile robot competition and exhibition. *AI Magazine*, 23(1):13.

Buehler, M., Iagnemma, K., and Singh, S. (2007). *The 2005 DARPA grand challenge: the great robot race*, volume 36. Springer Science & Business Media.

Chollet, F. et al. (2015). Keras. `https://github.com/fchollet/keras`.

Gageik, N., Benz, P., and Montenegro, S. (2015). Obstacle detection and collision avoidance for a uav with complementary low-cost sensors. *IEEE Access*, 3:599–609.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., and Osawa, E. (1997). Robocup: The robot world cup initiative. In *Proceedings of the first international conference on Autonomous agents*, pages 340–347. ACM.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

López-Rodríguez, F. M. and Cuesta, F. (2016). Andruino-a1: low-cost educational mobile robot based on android and arduino. *Journal of Intelligent & Robotic Systems*, 81(1):63.

Manzi, A., Fiorini, L., Limosani, R., Sincak, P., Dario, P., and Cavallo, F. (2016). Use case evaluation of a cloud robotics teleoperation system (short paper). In *Cloud Networking (Cloudnet), 2016 5th IEEE International Conference on*, pages 208–211. IEEE.

McLurkin, J., Lynch, A. J., Rixner, S., Barr, T. W., Chou, A., Foster, K., and Bilstein, S. (2013). A low-cost multi-robot system for research, teaching, and outreach. In *Distributed Autonomous Robotic Systems*, pages 597–609. Springer.

Mohanarajah, G., Hunziker, D., D'Andrea, R., and Waibel, M. (2015a). Rapyuta: A cloud robotics platform. *IEEE Transactions on Automation Science and Engineering*, 12(2):481–493.

Mohanarajah, G., Usenko, V., Singh, M., D'Andrea, R., and Waibel, M. (2015b). Cloud-based collaborative 3d mapping in real-time with low-cost robots. *IEEE Transactions on Automation Science and Engineering*, 12(2):423–431.

Murphy, R. R. and Burke, J. L. (2005). Up from the rubble: Lessons learned about hri from search and rescue. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 437–441. SAGE Publications Sage CA: Los Angeles, CA.

Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., et al. (2013). Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe.

Röning, J., Kauppinen, M., Pitkänen, V., Kemppainen, A., and Tikanmäki, A. (2017). The challenge of preparing teams for the european robotics league: Emergency. *Electronic Imaging*, 2017(9):22–30.

Saska, M., Krajník, T., Faigl, J., Vonásek, V., and Preucil, L. (2012). Low cost mav platform ar-drone in experimental verifications of methods for vision based autonomous navigation. In *IROS*, pages 4808–4809.

Valada, A., Velagapudi, P., Kannan, B., Tomaszewski, C., Kantor, G., and Scerri, P. (2014). Development of a low cost multi-robot autonomous marine surface platform. In *Field and Service Robotics*, pages 643–658. Springer.

Winfield, A. F., Franco, M. P., Brueggemann, B., Castro, A., Ferri, G., Ferreira, F., Liu, X., Petillot, Y., Roning, J., Schneider, F., et al. (2017). eurathlon and erl emergency: A multi-domain multi-robot grand challenge for search and rescue robots. In *Iberian Robotics conference*, pages 263–271. Springer.

Winfield, A. F., Franco, M. P., Brueggemann, B., Castro, A., Limon, M. C., Ferri, G., Ferreira, F., Liu, X., Petillot, Y., Roning, J., et al. (2016). eurathlon 2015: A multi-domain multi-robot grand challenge for search and rescue robots. In *Conference Towards Autonomous Robotic Systems*, pages 351–363. Springer.

# Appendix

Table A1: Analysis of resources after the UGV damage.

| Element | Re-usable | New use | Comments |
|---|---|---|---|
| Clearpath Husky UGV chassis | ✗ | | Too heavy for a lightweight chassis |
| Clearpath Husky motor board | ✗ | | CAN Bus chip broken |
| Clearpath Husky motor | ✗ | | Too heavy for a lightweight chassis; tricky control without motor board |
| Clearpath Husky battery | ✗ | | Too heavy for a lightweight chassis |
| Li-polymer 32000 mAh battery | ✓ | Main energy source | |
| Universal robot UR5 | ✗ | | Too heavy for a lightweight chassis |
| Laser SICK LMS111-10100 | ✗ | | Too heavy for a lightweight chassis |
| Novatel Smart6-L GPS | ✗ | | Too heavy for a lightweight chassis |
| Laser 3D Velodyne VLP-16 | ✗ | | Weight, power supply, and computational cost not sustainable |
| Intel NUC i3 | ✓ | Control PC | |
| Logitech 1080p Usb camera | ✓ | | Same use |
| ROS Middleware | ✓ | | Completely reusable (interoperability) |
| Mapping software | ✗ | | Lack of hardware (laser not present) |
| Autonomous navigation software | ✗ | | Lack of hardware (laser not present); limited time for developing/testing |
| 4g (LTE) communication protocol | ✓ | | Same use |
| Remote image classifier | ✓ | | Same use |
| Web-based GUI | ✓ | | Same use |

Table A2: ERL Referee Sheet, limited to the land robot: accomplished (✓), and unaccomplished (✗) tasks.

| | | | |
|---|---|---|---|
| A ground robot reaches the waypoints (WP) within a precision of 3m. | WP1 L<br>✓ | WP2 L<br>✓ | WP5 L<br>✓ |
| A ground robot reaches the WPs within a precision of 3 m in autonomous navigation. | WP3 L<br>✗ | WP4 L<br>✗ | WP6 L<br>✗ |
| A robot localises the unobstructed entrance in real-time in automatic way. | | ✓ | |
| Robots localise the obstructed entrances (E). | E1<br>✓ | | E2<br>✓ |
| Robots find a safe and unobstructed path to the unblocked entry of the building for a ground robot. | | ✓ | |
| From the starting point, a ground robot follows a safe path to the unobstructed building entrance. | | ✓ | |
| A ground robot enters the building through the unobstructed door. | | ✓ | |
| Within 30 minutes of start of the run, a ground robot reports the correct location of the missing worker inside the building. | | ✗ | |
| The missing worker is detected in real-time in an automatic way. | | ✗ | |
| A ground robot deploys the first-aid kit (within radius 1 m) from the worker inside the building. | | ✗ | |
| The ground robot(s) recognise the damages (D) on the wall of the building. | D1<br>✓ | | D2<br>✓ |
| From the building entrance, a ground robot follows a safe path to the machine room. | | ✓ | |
| A ground robot recognises the machine room sign in real-time and in automatic way. | | ✗ | |
| A ground robot enters the machine room. | | ✓ | |
| The ground robot(s) builds a geometric indoor map of the building. | Area1<br>✗ | | Area2<br>✗ |
| A ground robot recognises the ID of the correct set of valves in the machine room. | | ✗ | |
| A ground robot closes the correct valve. The robot must close one valve of the set autonomously and the other one manually. | Manual<br>✗ | | Auto<br>✗ |
| The ground robot and the underwater robot close the correct valves in a synchronised process. | | ✗ | |
| The ground robots return to the landing area once all the tasks have been done. | | ✗ | |
| The ground robot(s) transmits live position and images/video to the control station during the run. | | ✓ | |