

Autonomous Measurement Drone for Remote Dangerous Source Location Mapping

P. Croizé, M. Archez, J. Boisson, T. Roger, and V. Monsegu

Abstract—This paper presents research project conducted to realize a system able to measure autonomously a gas concentration in dangerous or hardly reachable places and locate the toxic source origin using commercialized micro Unmanned Aerial Vehicle (UAV), the AR Drone 2.0 from Parrot. The system relies upon the use of this UAV, an Arduino Nano board interfacing the gas sensor, and a GPS chip. A computer software also helps interfacing the hardware with user data.

Index Terms—Autonomous drone, automatic localization, measurement, gas, green algae, hydrogen sulfur, toxic material.

I. INTRODUCTION

Green algae are a real plague of pollution on French coasts and start to be a serious problem as well on Chinese coasts. They accumulate along the beaches and rot slowly, releasing a toxic and lethal hydrogen sulfide gas. This pollution has caused casualties and continues to do damages on the shore including the issue of nitrates. Botanic name of targeted algae is *Caulerpa Taxifolia*. Various researches and investigations along the past fifteen years have shown their continuous spread which tend to reduce biodiversity existence on polluted sites. As advised in [1] and [2], the concerned French department of Bretagne devoted a significant effort to beaches recovery and an important part of which is to map areas of gas emission for determination of riskiest places in order to prevent and efficiently target anti-pollution operations.

The purpose of built-up system is to send a drone for measuring gas concentration able to move autonomously towards a user-defined area. The samples associated with GPS (Global Positioning System) coordinates can then be analyzed to determine a mapping of polluted areas and the highest concentrated source location. The drone can be programmed to move in an obstacle free area without barriers by using internal sensor data as in [3]. And the sampling system would be attached to the drone.

Selected electronic hardware is based on Arduino Nano v3.1 board as central processor unit to interact with the other components: computer, gas sensor, GPS chip, AR Drone and level converter. The global hardware system should make the drone able to move autonomously up to defined destination

for performing the measurements and get back to origin point.

The paper is organized as follows. In Section II, the global system organization is detailed with the system requirements. In Section III, the autonomous orientation algorithm is displayed. In Section IV, the AR Drone 2.0 platform is presented with the script used to interpret received data and necessary pilot commands. In Section V, the gas sensor experimentation is detailed with its limits for a use in open environment. Section VI presents the Graphical User Interface (GUI) implemented for mapping gas distribution. Last Section VII shows the calculation used to locate the closest and most dangerous area in the surroundings. Finally conclusions are given and future possibilities are evoked.

II. GLOBAL SYSTEM ARCHITECTURE

Every component needs to be both tested individually then together with other components. The system should meet the following requirements (see Fig. 1):

- 1) Autonomous flight between two spots
- 2) Direct GPS data acquirement
- 3) Gas measurement with automatic data save
- 4) Gas distribution mapping
- 5) GUI to communicate with onboard hardware

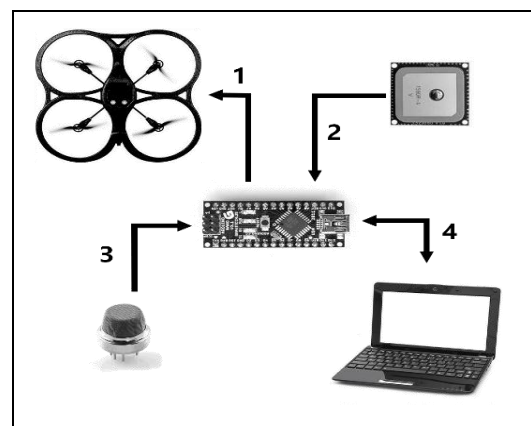


Fig. 1. System architecture.

The hardware part is represented by the embedded system modules which should be integrated on AR Drone platform. Indeed the added hardware modules should communicate with AR Drone motherboard to get the user instructions running. The software part is represented by GUI module which is running on a computer. Interface 1 uses a serial communication between Arduino board and AR Drone chip to transfer data necessary to make the correct maneuver. Interface 2 rules GPS data reception from the external LS20030 chip to Arduino Nano using a serial

Manuscript received June 2, 2014; revised August 11, 2014. This work was supported in part by the ECE Paris School of Engineering. All the authors are with the ECE Paris, School of Engineering.

Paul Croizé, Jérémy Boisson, and Matthieu Archez are with Embedded Systems at ECE Paris (e-mail: croizepaul@gmail.com, jer.boisson@gmail.com, archezm@gmail.com).

Vincent Monsegu and Thomas Roger are with Environment Sciences at ECE Paris (e-mail: vincent.monsegu@hotmail.fr, troger@ece.fr).

communication. Interface 3 is used to get the measures from the gas sensor using an analog connection. Finally interface 4 uses an USB connection with the GUI running on the computer.

III. AUTONOMOUS ORIENTATION

The drone should be able to localize its destination around itself in order to know where to go and how far it needs to go. In order to get this information, several processes operate on the coordinates received from GPS (x_{drone} , y_{drone}) and destination coordinates given by the user (x_{dest} , y_{dest}).

A. GPS Data Interpretation

The GPS chip LS20030 can transceiver NMEA 0183 using a simple ASCII, serial communications protocol defines how data are transmitted in a "sentence" from one "talker" to multiple "listeners" at a time. The GPRMC sentence is giving enough data for further processes. For example:

\$GPRMC,200350.800,A,4851.5515,N,00221.4276,E,1.64,353.11,220114,,A*6A

Then the string can be analyzed to extract the important data: current time, latitude and longitude. The coordinates are assigned to (x_{drone} , y_{drone}) after conversion from angular {degrees, minute, seconds} format into decimal degree (DD) values for calculations.

B. Drone Positioning

Once the drone coordinates are settled, one can estimate precisely where the destination point is around it.

Calculation of distance d : The use of haversine method is useful to determine distance between two coordinate points on Earth. According to following expression (2), one gets d in km with r being the Earth radius.

$$D = (2r) \arcsin\{\text{haversin}(y_{\text{dest}} - y_{\text{drone}}) + \cos(y_{\text{dest}})\cos(y_{\text{drone}}) \text{haversin}(x_{\text{dest}} - x_{\text{drone}})\}^{1/2} \quad (1)$$

Afterwards, the angle around which the destination is located is determined, see Fig. 2. This is a three step method. First a new landmark is created such that the drone becomes the origin. Consequently the virtual coordinates change for the drone to (0, 0) and for the destination point to ($x_{\text{dest}} - x_{\text{drone}}$, $y_{\text{dest}} - y_{\text{drone}}$). Secondly the landmark is rotated in order to get the drone heading to 0°.

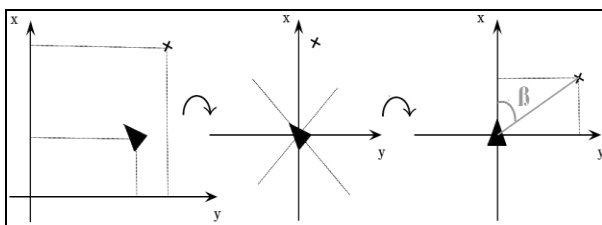


Fig. 2. Orientation diagram.

The last destination coordinates are calculated using following expressions:

Finally one can calculate the angle β between x-axis and the axis between the origin and the destination using the arctan function. Depending on coordinates sign combination, one may need to add 90, 180 or 270 additional degrees to

calculated value, see equation (1).

$$\beta = \left\| \arctan\left(\frac{x'_{\text{dest}}}{y'_{\text{dest}}}\right) \right\| + \begin{matrix} 0 \\ 90 \\ 180 \\ 270 \end{matrix} \quad (2)$$

As a result the following elements are obtained: distance to destination, angular location around the drone and difference between the drone heading and the heading it should have when facing the destination.

The heading calculation is done within a five seconds interval from two different drone positions at two different times. Once these two positions are known, it is possible to calculate the drone heading direction angle by using the same previous method.

IV. AR DRONE 2.0 PLATFORM

The drone needs to move autonomously from a point to another one in 3D space. To this aim, a C program which can be launched directly on Drone processor has been developed. This code can be uploaded via USB key reading. Then a cross-compilation on ARM processor of this code can be performed directly on the drone. It is necessary to be aware of the whole inside technology and sensor that can be accessed ([4] and [5]) to use the correct data exchange.

A. Serial Communication with Arduino Board

For communication with the Arduino board, the Parrot proprietary plug located below the drone has been used, but it only allows data reception. The plug also has some specific characteristics such as a 1.8V TTL data definition which can be handled by a level converter adjusting voltage between Arduino Nano and the drone. Once this detail being settled down, a frame template based has been designed to transfer data correctly:

$$\text{\$}<\alpha>,<\text{heading}>,<\text{arrived}>* \quad (3)$$

- The '\$' is the start character defining the frame beginning.
- The ',' represents a delimiter used to separate different types of data. For example : <alpha>,<heading>
- The '*' is the end character defining the end of the frame.

With the frame definition, the software only needs three variables to find its way. The <alpha> value is an angle between 0 and 359 calculated by the Arduino board and represents the vector angle between drone current point and destination. The <heading> value is also an angle between 0 and 359 degrees defining the instantaneous drone heading. The <arrived> value is a Boolean which is worth false until the drone reaches its destination.

Finally, according to these received data, pilot commands can be operated from AR Drone processor such as rotating, moving forward or landing.

B. Multi-Threading

The software uses two different threads, one handling the communication with Arduino board and another one controlling the drone. The *Arduino thread* is basically a loop

reading through the serial port waiting for the data frame. It will analyze, divide and save the new data in order to keep the navigation data update.

The *Control thread* is split into two main functions. The orientation function keeping the drone in the right direction by rotating it if <alpha> and <heading> are not close enough, and the forward function which has to make the drone take off, move forward and land when arrived. In concrete terms, it will send moving forward command to the drone until the destination is reached or the drone needs to change its orientation.

C. Maneuver Algorithm

The function in charge of drone maneuver is based on a simple algorithm. Indeed, as the drone will always go forward when it moves, it is possible to calculate its heading thanks to the GPS. As destination angle vector <alpha> is known, by comparing these two values, the algorithm can easily know if the drone is going to the right direction. Otherwise, the algorithm will make the drone turn on itself until it is in the right direction.

V. GAS SENSOR UNIT

Before making any sampling measurement, it is compelled to have an understanding of gas composition and physical behavior with information provided in [6]. Used gas sensor MQ-136 is a semiconductor sensor for toxic gas Sulfur Hydrogen H_2S . It uses the 5V output from Arduino Nano as a small heater with an electro-chemical sensor. The sensor needs to be self-calibrated during 24 hours before being used for test. This preheat time is necessary to get as precise as possible measurements. The measure itself is an analog signal sent from the sensor to Arduino card. It operates as follows: when the gas is actively present in the air, gas sensor conductivity rises with gas concentration. It is characterized by a good sensitivity, a long life estimation, and it can be used on a simple drive circuit. It can be used for multiple purposes such as domestic, industrial or portable gas detection. It can handle a 220ppm maximum gas concentration (see Fig. 3).

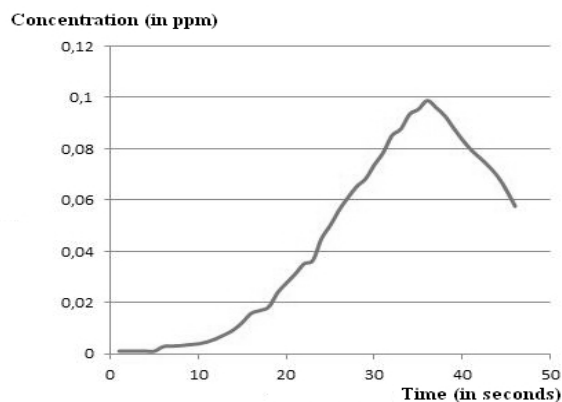


Fig. 3. Gas concentration evolution vs time.

To calibrate the measurement it is important to know sensor resistance R_0 at 10ppm of H_2S in the air. This value goes along with the sensing value R_s calculated from the input tension received from the sensor. The following formulas (4) show how to calculate R_s depending on input value received

from the sensor and the ratio between R_s and R_0 giving gas concentration in ppm:

$$R_s = \frac{\text{input} \cdot 10^3}{5.0 - \text{input}} \quad \frac{R_s}{R_0} = [H_2S] \quad (4)$$

After reaching 0.1ppm concentration during practical tests, it has been possible to determine the value R_0 at 1M Ω . From the calculated results, it is now possible to know whether the gas concentration is dangerous or not (see [7] for H_2S exposure and environmental effects).

VI. GRAPHICAL USER INTERFACE

The user interface has three main goals to fulfill:

- It must allow the user intuitively search, navigate on a map of a desired location and pinpoint the locations where he wants the drone to take its gas samples, see Fig. 4.
- It must be able to send information to the Arduino board, in the form of GPS coordinates (latitude and longitude), so it can be used for navigation.
- Because the user may want to visualize the result of a drone sampling mission; the GUI must display the information concerning the gas quantity associated to the chosen locations.



Fig. 4. Display of local area and specific points of measurement.

The user interface itself is implemented through a Windows Form coded in C# using the Visual Studio 2013 IDE, for libraries to read and write through a USB serial port are available and it is quite possible to integrate HTML and JavaScript to a Windows Form. The mapping process is performed in a second time frame once the measurements are performed. Other processes based on image scan (see [8]) can be used to realize samples mapping if the environment constraints do not remove that possibility.

Indeed to offer a complete and intuitive interface to the user, the Google map API accessed through a JavaScript seemed an acceptable alternative. The JavaScript contained in a HTML page is called from the Windows form by the user to launch and display the Google Map interface at a given location. A few options have been implemented to permit the user to place or to remove markers and access their GPS coordinates, as well as to transmit them to the windows form once a set of markers is defined.

Once a set of markers is entered by the user on the map, he can transmit the coordinates of these markers to a board after entering the COM port to which the board is connected. An encoded string containing all the information will be transmitted to the Arduino Nano board by serial transmission.

Retrieving data from the Arduino is using a similar process, a serial transmission is initiated and the GUI waits for an encoded string from the Arduino: data concerning the time, longitude, latitude and gas concentration of each sample location is transmitted and put into a text file, to be displayed on the map afterwards.

The start location can be determined by the user by Geocoding, requiring only entering an address for the API to plot the map around the specified location.

Moreover, if a session was saved by the user, it will automatically be launched as the program start, giving a quick access permitting the user to quickly check or modify the drone route and visualize the gas concentration of each pinpoint.

VII. DANGEROUS SOURCE LOCATION

From studied example, the goal is to locate the area with highest concentration in H_2S in drone vicinity. But this system can be used for other purposes depending on the embedded sensor and the characterization that needs to be drawn out of it: high or low gas concentration, dense populated area during demonstrations, and fire detection in forest...

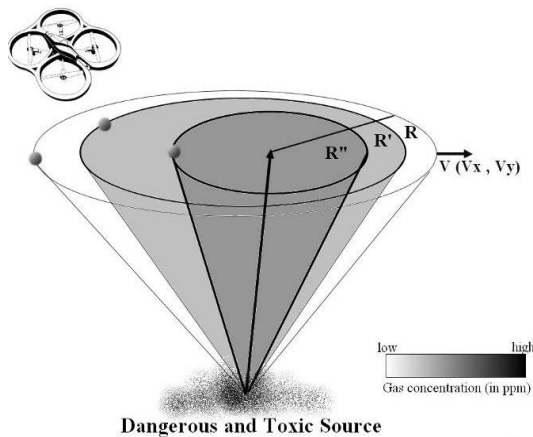


Fig. 5. Drone measurement of three different gas concentration at three different spots.

Once the user decided to send the drone on a specific location, this one may start looking for the most dangerous location around itself. To do so, a new set of destination coordinates needs to be defined. A unique sampling of gas concentration does not seem therefore sufficient to give an accurate representation of gas distribution. After first gas sampling, the drone can randomly go towards a new nearby location (no more than 10 meters away) and take a new sampling. Afterwards, another new sampling can be performed on a location in order to draw a triangle shaped with points made of the three previous coordinates. The calculation of last and third sampling needs to be done in order to make an equilateral triangle out of the three points.

From the three realized samples, the one with strongest characterization factor is considered as the main point of the next set of destination coordinates points, see Fig. 5. The higher is gas concentration, the closer is the drone to the dangerous source. This approach is different from the Kernel algorithm (used in [9] and [10]) as it requires less samples but also reduces precision in the detection final result.

After a specific number of measurement iterations, or when the characterization factor tends to stay static, the system can stop scanning the area. One can then study the place following a 2D flat plan at the level the measurements have been made. From last measurement point coordinates, here $A(x_A, y_A)$, a detection circle can be drawn with a radius R depending on researched precision to localize emitting source. A small radius would give an incorrect source location and a large radius would give a too large area if the source needs to be located precisely. The value of this "search radius" should be modified by the user depending on the research area and adapted to the type of studied toxic source (volatility or dispersion of the material).

The gas dispersion is based on an elliptical fluid mechanics. The resources [10] and [11] describe gas distribution modeling as an ideal application for mobile sensors like drone based for example: regarding environmental conditions, the optimal sensor location may vary and sampling/mapping results can be altered. Gas modeling purpose is to give a more accurate representation of gas distribution by going through a set of spatially and temporally distributed measurements of relevant variables such as : gas concentration (by use of electrochemical sensor for example), wind, temperature... Hydrogen sulfur emissions and distribution can be affected by factors like temperature and wind, which can largely vary given drone location. A unique sampling of gas concentration does not seem therefore sufficient to give an accurate representation of gas distribution. In [9] (the model of which is based only on gas measurement) it describes gas behavior as quite challenging to predict: gas tending to get into "packet" following erratic trajectories and creating intermittent patches of high concentration. The fact that there exists a technical limitation (impossible to sample a whole zone at the same time as done in [12]) makes this kind of mapping even harder... Two versions of distribution study can be considered: 2D and 3D. As the second one is demanding much more data measurements and is not presenting any real advantages in present case, the 2D version has been preferred. Even though gas distribution would be more accurate if represented on an elliptical area, a circle can be chosen to cover a larger area. This circle obeys the Cartesian equation:

$$(x - x_{S'})^2 + (y - y_{S'})^2 = R^2 \quad (5)$$

Once R has been chosen, the circle of detection center S' ($x_{S'}$, $y_{S'}$) can be set. The localization of original source coordinates S (x_S , y_S) can be retrieved from S' coordinates with correct wind vector deviation correction applied over time t . This vector is represented by $V(x_V, y_V)$ on Fig. 6.

$$d^2 = (x_{S'} - x_S)^2 + (y_{S'} - y_S)^2 \quad (6)$$

with $x_{S'} = x_S + v_x \cdot t$ and $y_{S'} = y_S + v_y \cdot t$,

$$d = ((v_x \cdot t)^2 + (v_y \cdot t)^2)^{1/2} \quad (7)$$

From this result, estimated distance to original toxic source can be calculated. Although this method seems to give acceptably accurate results, it can only be implemented under the condition of being able to get the necessary numerous angles and information, which cannot be easily achieved with AR drone since the data are only used internally and cannot be used otherwise. A simple use of dedicated anemometer could also be a good option to obtain this vector.

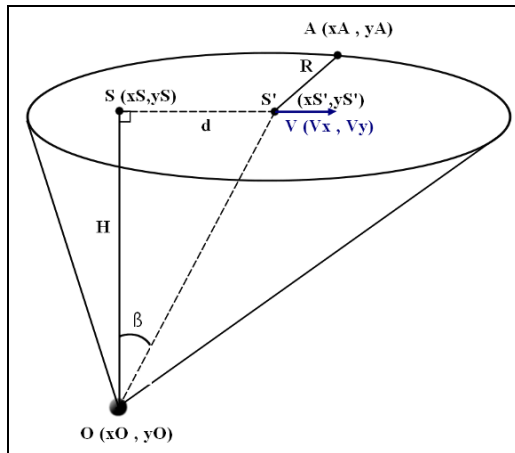


Fig. 6. Geometrical diagram for source localization.

A deeper study and/or result validation can be established by making a new sample on a lower level below S point position. If needed, a further human intervention on site can be executed, or local area can be forbidden to human activity until further notice.

VIII. CONCLUSION

To access the measurements of specific element at a (sometimes instantaneously unreachable for human operator) prescribed location or in prescribed area, an autonomous drone system carrying the relevant sensor has been developed with very accessible technical involvement without looseness on measurements quality. System set up has been described and functionalities have been organized to give access to information in easy-to-use interfaced way. Finally nearby toxic source detection can be implemented to localize the original source spot and prevent any danger for human interaction or correct any immediate danger such as a gas leak or a fire start. Advanced research and improvements can be conducted to higher the detection precision depending on the danger targeted. Also building a resistant hardware shell with embedded sensors can improve data acquisition possibilities.

ACKNOWLEDGMENT

The authors are very much indebted to ECE Paris School of Engineering for having provided the environment where the project has been developed, to M. Ataman for his guidance and Pr M. Cotsaftis for help in preparation of the

manuscript.

REFERENCES

- [1] P. Bartoli and C. F. Boudouresqu, "Transmission failure of parasites (Digenea) in sites colonized by the recently introduced invasive algae *Caulerpa taxifolia*," *MEPS*, vol. 154, 1997.
- [2] G. Newcombe and J. House, "Management strategies for cyanobacteria (Blue-Green Algae): A guide for water utilities," Research Report 74, Cooperative Research Centre Water Quality Research Australia, 2010.
- [3] A. Visser, N. Dijkshoorn, M. van der Veen, and R. Jurriaans, "Closing the gap between simulation and reality in the sensor and motion Models of an autonomous AR drone," in *Proc. the International Micro Air Vehicles Conference*, Amsterdam University, Digital Academic Repository, 2011.
- [4] P.-J. Bristeau, F. Callou, D. Vissère, and N. Petit, "The navigation and control technology inside the AR Drone Micro UAV," in *Proc. 18th IFAC World Congress*, Milano, Italy, August 28-September 2, 2011.
- [5] T. Dean, "An autonomous flight controller for the parrot AR Drone," MSc Robotics Dissertation, Plymouth University, 2013.
- [6] N. Bonnard, T. Claver, and M. Falcy, *Fiche Toxicologique FT32 Sulfure d'Hydrogène*, National Institute for Research and Security, 2009.
- [7] H. E. Kalusche, *Hydrogen Sulfide-Health Effects, Detection and Exposure Prevention*, Industrial Hygienist, Safety and Environmental Management Office, Environmental Laboratory.
- [8] R. Steffen and W. Förstner, "On real time mapping for unmanned aerial vehicles," Report of Dept of Photogrammetry, Bonn Univ. Germany, International Society of Photogrammetry and Remote Detection, 2008.
- [9] P. P. Neumann, S. Asadi, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, "Micro-Drone for Wind Vector Estimation and Gas Distribution Mapping," *IEEE Robotics and Automation Magazine*, vol. 6, no. 1, January 2011.
- [10] A. J. Lilienthal, M. Reggente, M. Trincavelli, J. L. Blanco, and J. Gonzalez, "A statistical approach to gas distribution modelling with mobile robots — The kernel DM+V algorithm," in *Proc. IEEE/RSSJ International Conference on Intelligent Robots and Systems*, 2009.
- [11] M. R. Beychok, *Fundamentals of Stack Gas Dispersion*, 4th ed. 2005.
- [12] C. Stachniss, C. Plagemann, A. Lilienthal, and W. Burgard, "Gas distribution modeling using sparse gaussian process mixture models," *Robotics: Science and Systems IV*, 2008.



P. Croizé was born in Paris, France in 1992. He is a student at ECE Paris School of Engineering, France since 2010 and at Aalborg University, Denmark in 2013. He got his high school diploma at Les Franches Bourgeois School, Paris in 2010; bachelor degree of signal processing obtained at Aalborg University, Aalborg Denmark in 2013. His field of study specialized in embedded systems.

He worked at Ubisoft International in Paris as an intern on international maintenance servers. Later he was a developer at EcoCo2 and worked on an Android application for EcoCo2. Then he developed with Archos and EcoCo2 a system to monitor power consumption at home using an Android Tablet and multiple Bluetooth Low Energy technology sensors. During his studies in Denmark, he could improve and deepen his knowledge about signal processing and communication systems in order to build systems able to fulfill requirements with accuracy.



M. Archez was born in Paris in 1992. He is a student at ECE Paris School of Engineering, France since 2010 and at Aalborg University, Denmark in 2013. He got his high school diploma at Lycée Condorcet, Paris in 2010; bachelor degree of signal processing obtained at Aalborg University, Aalborg Denmark in 2013. His field of study specialized in embedded systems and project management.

He worked at the I3F asset management in Paris. Later he worked at Amundi IT Services, Paris in the IT network/development team. Currently working as an internship student within Konkuk University's Networking development laboratory, Seoul Korea, where he is studying innovative communication architectures (eXpressive Internet Architecture, Overlay multicasting Relay). Throughout his studies in Denmark and Korea, he hopes to extend his area of knowledge and expertise while improving his ability to work and learn in contact with various work cultures and approaches.



J. Boisson was born in 1992 in Metz, France. As a student at ECE Paris School of Engineering, he specialized in Embedded Systems since 2010. He got his high school diploma at Henri Moissan, Meaux in 2010, bachelor degree of science specialized in signal processing, Aalborg University, Denmark in 2013.

His experience as a commercial assistant and qualified handyman gave him the possibility to develop his skills during summer internships. He worked as an intern on Graphical User Interface development for the Defense department of Thales French Company.



T. Roger was born in Clermont-Ferrand, France in 1991. As a student at ECE Paris School of Engineering, France since 2012, he specialized in energy and environment. He got his high school diploma with mention at the High School of Chamalières, France in 2009.

He worked at Bio Rad Verdot in Riom, France, as an intern on obtaining environmental standards of quality.

Then, during an internship at ECHALIER, he highlighted the crucial points about energy and helped to geotag the company's fleet.



V. Monségu was born in Mont-De-Marsan in 1991. He is a student at ECE Paris School of Engineering, France since 2012 and in preparatory class for "Grandes Ecoles" from 2009 to 2012 at the high school Louis Barthou, Pau. In 2009, he got his high school diploma at the high school Gaston Crampe, Aire-sur-l'Adour.

He worked as a handyman at a construction site of Spie Fondations in Paris. Later, he worked at the Hydroelectric SHEM Company in South of France as an intern on Telecom Technology. He had to design a wireless communication solution to link sensors to a very distant data recorder.