EPS – Article
Close Range Remote Control With Advanced Capabilities

Magda Huiculescu, Marco Fernandino Carlino, Michal Kobus, Miquel Àngel Pereira

1

Abstract— When a marine buoy with electronic equipment is deployed it is usually switched off during transport to the deployment location, which can take days to weeks. The system is then switched on just before its final installation. Electronics and batteries are stored in sealed containers that can be difficult to manipulate especially while at sea; it is preferred not to open them unless absolutely necessary. Therefore, a rugged switch or connector can be installed on the exterior of the buoy. These external connectors come with extra cost, add another path for water intrusion and might allow accidental manipulation. An alternative solution could be radio linked remote control (RC) enough rugged and safe to be used confidently in these and other critical applications.

Keywords— Android App, Bluetooth, Close range remote, Power consumption.

I. Introduction

Humanity continues to explore and reshape the world. It conquered the land and started reaching for the stars. It is also strengthening its presence on and in the seas.

One of fields of research created thanks to this third avenue of human interest is acoustic marine research.
As the name suggests it deals with underwater sounds. Those may come from artificial or natural sources. Monitoring noise is important for different reasons as sound pollution, created by ships or construction of oil platforms that may threaten the aquatic fauna and need to be controlled. A well placed set of underwater devices is a perfect tool for researching project on some species of animals.
One of the crucial tasks is deployment of buoys that gather data. This requires a lot of expertise to develop, maintain and use them and a lot of effort to deploy. Some of requirements they need to meet is perfect waterproofness and ability to operate for extended periods of time on battery power. In addition, there are logistic difficulties caused by rough sea conditions when the equipment is being delivered to its destination. Therefore, it is important to identify problems and complications and overcoming them to streamline the process.
One of entities that specialize in the previously mentioned field is Laboratory of Applied Bioacoustics (L.A.B.).
L.A.B. was established in 2003 as part of the "Universitat Politècnica de Catalunya" (UPC) and operates on its campus in Vilanova i la Geltrú. It analyses data from systems of partner co-operators but also deploys its own buoys. L.A.B. has also history of participating in European Project Semesters (EPS), a one-semester course designed to train final-year engineering students to work in international teams.

Figure 1: Laboratori d'Aplicacions Bioacústiques logo.

The proposed challenge is to eliminate the need of disassembling and reassembling the buoy to turn the electronics inside on and off on a boat just before deployment. It is inconvenient to switch it on on land as it would cost a premature power consume. As stated the sea conditions may make this difficult and time-consuming due to bad visibility, movements of the vessel or cold that forces researchers to wear gloves.

In this article there will be evaluated the design considered for this remote, the communication established between the remote or an android app for a Smartphone and an internal unit implemented inside of the buoy. At the end of this article, there will be presented a reflection and conclusion about this research.

II. Initial considerations

The first initial consideration was how to integrate a wireless technology in the pre existing buoy's hardware, as the remote needs a system to communicate with. Options are two:

1. Implement the pre existing hardware of the buoy and integrate a wireless technology.
2. Develop a separate unit that will communicate with the remote and will receive status data from the pre-existing hardware.

The selected option is to create a separated unit that takes status information from the buoy system and features as:

- A relay, to switch on and off the power of the buoy's hardware.
- Temperature and humidity sensor, to monitor eventual buoy overheating.

- Current and voltage sensor, to monitor the power consumption of the buoy's hardware.

The reasons were that implementing the pre existing hardware could have caused issue on compatibility, involved the redesign of the case, the printed circuit board (PCB) and the memory of the hardware was limited from the other functions that are already operating.

III. COMMUNICATION

One of the most important parts of this project is the communication between the designed remote control or the developed Android App and the internal unit located inside of the buoy. To establish a successful communication, different wireless technologies were analyzed and compared. The best option has been chosen according to their distance range and, specially, the needed power consumption. In the following table there are shown different wireless technologies as well as their aim characteristics.

	Near Field Communication (NFC)	**Infrared Data association (IrDA)**	**Bluetooth Class 2**	**Radio Frequencies 433Mhz (RF)**
Set-up time	< 0.1 ms	0.5 s	6 s	< 0.1 ms
Range	Up to 10 cm	Up to 2m	Up to 10m	Up to 300m
Usability	Human centric, easy, intuitive, fast	Data centric Easy	Data centric medium	Item centric, easy
Selectivity	Security	Line of sight	Who are you?	Partly given
Use cases	Pay, get access, share, initiate service, easy setup	Control and exchange data	Network for data exchange ,headset	Item tracking
Consumer experience	Touch, wave, simply connect	Easy	Configuration needed	Get information

The best suitable technology for establish communication

	RF 315/433 MHz module	**Bluetooth module**
Power consumed	40mA - Transmitter 5.5mA - Receiver	30mA – Fully working

between the remote and the buoy is the Bluetooth technology. The Bluetooth communication technology permitted to expand the features of the project developing also a mobile application in substitution of the remote. Using bluetooth allows to develop a more accessible device and it will be able to do a multiple number of functions compared to another wireless technologies. The decision of choosing Bluetooth gave the chance to implement even more the system including a mobile application that will replicate the functions of the remote and used eventually in substitution of the device. The communication between the internal unit and the buoy hardware is implemented with a wire serial communication. Tests have been already conducted on the prototype of the internal unit simulating the information that would be received from the board with Putty [1], a serial software application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection.

IV. HARDWARE DEVELOPMENT

The first consideration is to select the right components and materials for the development of the remote control and the internal unit placed inside of the buoy.

A. *Remote control*

The brain of the communication it was chosen in form of a microcontroller.

There are many microcontrollers available on the market that might be suitable for our project but the ones taken in consideration have been 3 (Raspberry Pi zero, Arduino Uno and Arduino Nano) and the one chosen was Arduino Nano due to the small dimensions that allows to be fitted in a small remote control and the compatibility with all of the components included in the remote.

Figure 2: Arduino Nano microcontroller.

For establish the wireless communication it was needed a bluetooth module that is in charge of sending and receiving information between the two different devices designed. The bluetooth module chosen is a 2.1 Bluetooth Module by Microchip. This module presents a low power consumption and it is very easy to integrate.

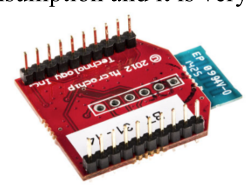

Figure 3: Bluetooth module by Microchip

A Liquid Crystal Display (LCD) has been selected to show necessary information about the status, and transmitting data of the buoy.

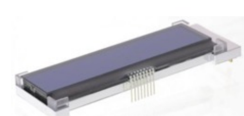

Figure 4: 4x20 LCD Midas screen

This LCD has 20 columns and 4 rows enough for display the information about the buoy. It has also a background light that allows clear visibility in dark environments.

For power up the remote control there were used two AA batteries. These AA batteries provide power to the Arduino microcontroller inside of the remote lasting at least 24 hours without disconnecting the remote control.

B. *Internal Unit*

The power source of the internal unit will be the battery that is already running the system, the internal unit will take just a little bit of power to process its function and act as a switch to give power to the buoy's system.

Switching on and off a current require a relay. A relay is a switch controlled by an electrical circuit that, across an electromagnet and a coil, defines the closing or the opening of other circuits. For the implementation of the internal unit it will be used a relay bi-stable component (ON, OFF) with two coils inside.

Figure 5: SPDT Bistable 5V Relay.

A part from controlling the state of the current, it is important to track how much current is consumed and estimate so the remaining power. To achieve that a current sensor is used, a ACS712 by Allegro.

It is a current sensor using the Hall Effect to work providing an economic and precise solution to measure AC or DC current. The current sensor send data using an analogue output.

Figure 6: ACS712 Current Sensor.

The microcontroller selected for the internal unit that is in charge of executing all the orders and connecting the the components inside of this unit is the Arduino Nano microcontroller. This microcontroller is selected taking the same considerations that were taken selecting the microcontroller for the remote control.

To detect and send information related to the condition of the buoy, such as temperature or humidity information, it will be necessary to connect to the microcontroller a digital temperature sensor. The digital temperature and humidity sensor used is the DHT11. The DHT11 sensor is compatible with the electronic systems operating between 3V-5V and it does not need of external electronic components.

Figure 7: DHT11 Temperature and Humidity sensor

V. SOFTWARE DEVELOPMENT

The software developed for this project consisted of three main parts. They were software of the remote, internal unit and the application. First two were as result of choosing Arduino boards for hardware were written in mixture of C and C++ - language used by the boards [2]. Arduino IDE was chosen because the team found it efficient, easy to use and reliable in case of programming for Arduino. The application was developed for Android as at the time it was most popular mobile OS[3]. Therefore it has rich knowledge base and many tools to use during development. The app was written in Java - language used for writing software for Android [4].Android Studio IDE was used as the team had most experience with it and was satisfied with its performance in the past. All software was tested individually and shown to company supervisors to find bugs and gather feedback. Final versions were integrated together to be once again tested and improved.

VI. REMOTE SOFTWARE

I. MENU SOFTWARE

The menu software was developed according to three goals. It was to be lightweight, easy to maintain and easy to expand. The team wanted not only to create a working prototype, but a satisfactory solution for microcontroller-based multilayered menus in general. First of those goals was result of rather limited memory of Arduino board and microcontrollers in general. It was also important to leave space for the function that the menu will allow the user to perform as they will require memory for themselfs. After some trials and two iterations final menu was developed. It made use of the three buttons of the remote and the screen. This last component proved to be a biggest challenge as libraries known to the team were not able to utilize it properly. Therefore the team had to modify a library in such a way that it rendered the screen useful again.

The menu logic itself was affected by structure of all Arduino programs. In typical C program the flow of the program could be passed from one function to another. However on Arduino the flow of the program should always return to function loop. As quickly as possible. This function is the only one that runs constantly and where the button presses can be handled. An empty sketch of a program can be seen in Figure 9.

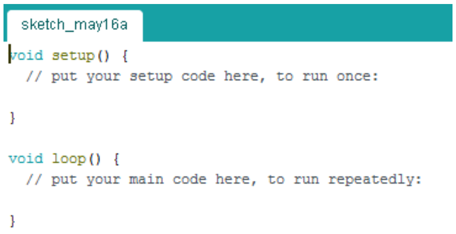

Figure 8: Structure of empty Arduino IDE

Therefore the menu program needed a way of deciding which function to run from the loop in response to button presses. This was done using pointers. The menu program keeps a pointer to current submenu function and its text. The position of the cursor is also known at all times. It that way the proper function is always called. It may mean going back or into another layer of the menu or performing another action. Separate functions for drawing text on the screen and setting the pointes during transitioning to another layer of menu where written. It resulted in reasonably short and easy to understand code that separates handling the screen, keeping track of user's position in the menu and logic of each submenu. This resulted in easier maintenance and bigger potential for further development.

A success was also achieved in case of programming the internal unit. Its software manages sensors listed previously, responds to signals from the remote or app and is capable of serial communication with computer that operates on Linux OS. Such computer is at heart of the buoy.

VII. Android App

The remote is designed for particular task and has following features: input limited to menus from which user can pick items and ability to connect to eight particular units which addresses are remembered. Drastic changes to the requirements may result in changing both software and hardware. It may be complicated and time consuming.

The application does not have mentioned limitations. Firstly a mobile device such as smartphone offers rich input possibilities with on-screen keyboard. It can also pair with virtually unlimited number of devices It is only a matter of accessing those functionalities from the application. All functionalities of the remote can be reproduced on the smartphone and adding new will not require changes only in software. Therefore, the app may work as a replacement of the remote in case of:

- Loss of malfunction of the remote;
- Appearance of new requirements, that exceed capabilities of current design and urgently need to be implemented;
- Accommodating need to control more that eight buoys.

However the application is not without drawbacks. Most mobile devices are not waterproof nor can be operated in gloves. They make it a worthwhile alternative, but prevent it from replacing the remote altogether.

The app has four main states. Each of them is a separate activity. An activity in Android is a simple screen where user can perform particular actions [5]. States are presented in Figure 10.

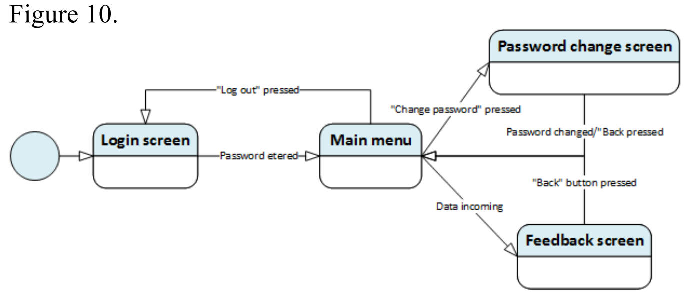

Figure 9: Android application states

First important feature of the app is associated main menu activity. It uses Android API to enable bluetooth, present user with list of paired devices and allows to perform all actions available on the remote. It also constantly listens for feedback from the internal unit. However it presented a development challenge as Android applications are defaultly run as processes with one thread of execution. [6] It means that instructions are executed one after another and one at a time. If the listening was to be done in the main, default thread it would mean that the app would be frozen until it was done. Therefore another thread of execution is launched exclusively to do the listening. Unfortunately only the main thread can interact with the user interface of the application. To do it from another thread a handler [7] is needed. This special class needs to be associated with thread where it should post information or commands. Then they can be posted there from another thread.

Second important feature is password. When the application is launched for the first time, user is presented with a possibility of inputting a password. If the password is correct main menu is shown and previous activity is finished. That way it cannot be navigated back to by pressing the mobile's back button. The user may also choose to log out. In that case all other activities are closed and user is presented with login screen. That way other activities become unavailable until right password is inputted.

This is not standard behaviour.[8] In Android one application is usually a one task. This task operates as a stack. If it starts with activity A this activity is the only item on the stack. If an activity A launches activity B, the activity B is on top. If the back button is pressed, activity on top (in this case B) is closed and A comes into focus. If user switches to another tack, stack of previous task is preserved. This is why logout opron was programmed. The team do not want unauthorised user to access activities that were left open by the user.

To store the password and change it shared preferences are used. They are preserved between the application launches and therefore allow the changed password to persist. They are also accessible only to the application created them. Usage of shared preferences [9] together with modifying the behavior of activities increased security of the app. As long as the authorised user remembers to logout, the app cannot be easily accessed.

VIII. DESIGN AND CONSTRUCTION

The design process implies following a succession of steps, from the idea to the prototype.
In the brainstorming stage of the process were set the main requirements for the design: it has to be easy to handle (even with gloves), it has to offer good protection to the electronics inside and it should have aesthetic qualities.
The main inspiration field was marine life. The chosen design, after the sketching stage, was based on the shape of the Mako shark tooth.

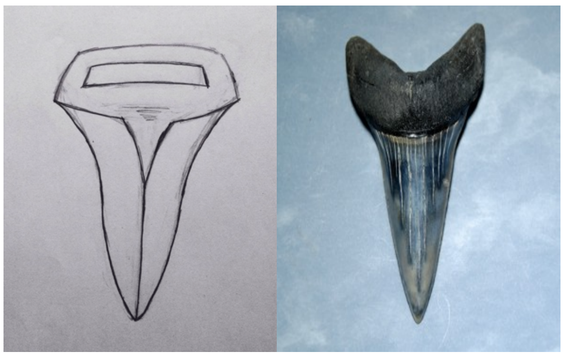

Figure 10: Sketch drawing in comparison with the Mako shark tooth

To determine the dimensions of the remote, were taken in consideration two aspects: the hand dimensions of an average user and the dimensions of the electronic components.

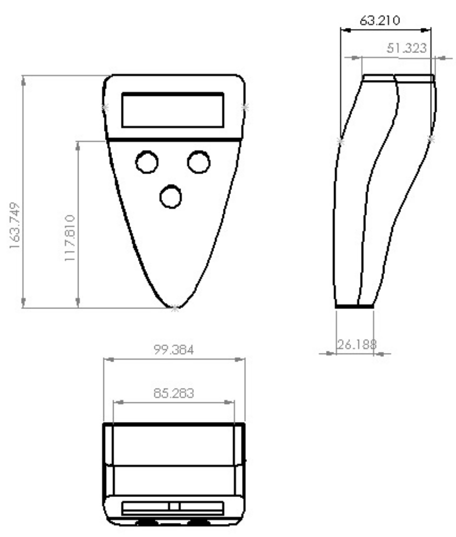

Figure 11: Dimensions of the remote

The ergonomics stage of the design was composed of: the most suitable position of the buttons and the position of the display. The position of the buttons was calculated and tested on polistyren models. Accordingly to the research, the display was positioned to an 20° tilt**.** This angle will offer the user eye comfort and also a straight position of the body.

The material chosen for the remote is ABS. The manufacturing method for the prototype is 3D Printing and the manufacturing method in case of mass production is injection molding. The research and graph analysis completed have shown that ABS is suitable for both manufacturing processes.

The color of the remote was chosen accordingly with the OSHA (Occupational Safety and Health Administration) that decrees that safety yellow, code is 13591, hax is #EAC234 is the color that should be used in the industrial field.

Regarding the ingress protection, an initial value was set to IP65. In order to obtain this value it was designed a silicone gasket and a bumper.

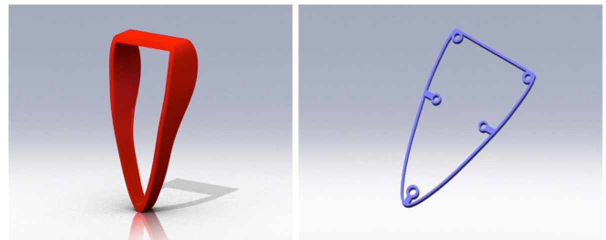

Figure 12: Ingress protection additions

To avoid the dropping of the remote, it will be secured by two accessories: one designed for the wrist and the other designed for the belt or to be hanged on the wall. The accessories can not be used together. They are designed to fulfill the user's desire accordingly to they way he prefers to secure the remote.

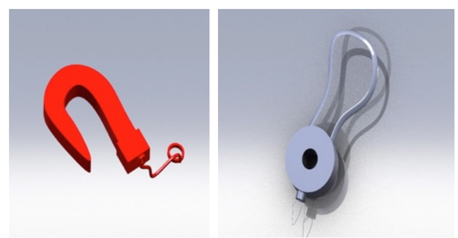

Figure 13: Safety accessories

The final design is illustrated in the 3D modeling below, together with the position of the components, display and buttons:

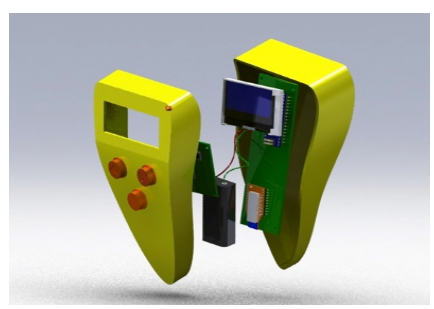

Figure 14: Exploded view of the remote and components.

IX. CONCLUSIONS

The team developed a functioning prototype of the proposed system: internal unit and remote that communicate with each other and a mobile application. It was also proved that serial communication between the internal unit and buoy's hardware is possible.

Bluetooth's popularity was justified as it proved to be ideal for close range communication. It's relatively low power consumption coupled with sufficient bandwidth, high accessibility of hardware and software and rich knowledge base are strong advantages. The team had no major problems with finding libraries and modules and writing software.

Another crucial component of the project were Arduino boards. They were proved to be reliable, versatile and very easy to program thanks to the dedicated IDE and powerful community. The tam believe that the choice of microcontroller could not have been better.

Android also was not a letdown as all desired features were implemented in the final application. Therefore the team strongly recommends using most of mentioned technologies and devices in similar projects. It is also hoped that numerous, smaller comments made previously will be useful.

Probably the most problematic component was the screen. Utilizing it required modification of existing library. A lack of proper documentation from the manufacturer was also an obstacle. After this experience, the team learned that when choosing a component it is equally important for it to fulfill technical requirements as to make sure it will be easy to integrate with other parts of the project. Time is an important resource and may be worth considering if it can be spend to compensate for example for a lack of proper support.

References

[1] Putty Software. Accessed: 2016. http://www.putty.org/

[2] Arduino Microcontroller website. Accessed: 2016. https://www.arduino.cc/en/Main/FAQ

[3] Smartphone Market Share 2015. http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[4] Application fundamentals in Android. Accessed: 2016. https://developer.android.com/guide/components/fundamentals.html.

[5] Android Activity. Accessed: 2016 https://developer.android.com/reference/android/app/Activity.html

[6] Processes and threads. Accessed: 2016. https://developer.android.com/guide/components/processes-and-threads.html

[7] Communicating with the UI thread. Accessed: 2016. https://developer.android.com/training/multiple-threads/communicate-ui.html

[8] Tasks and back stack. Accessed: 2016. https://developer.android.com/guide/components/tasks-and-back-stack.html

[9] Storage options. Accessed: 2016. https://developer.android.com/guide/topics/data/data-storage.html

[10] Escola Politècnica Superior d'Enginyeria de Vilanova i la Geltrú , "EPS programme," European Project Semester, 01 2016. [Online]. Available: http://www.epsevg.upc.edu/eps/the-eps-programme [Accessed 2016]

[11] "Injection Molding Process, Defects, Plastic," 2016. [Online]. Available: http://www.custompartnet.com/wu/InjectionMolding#materials

[12] "3D Printer Filament Comparison | MatterHackers," 2016. [Online]. Available: https://www.matterhackers.com/3d-printer-filament-compare

[13] "Plastics International," 2016. [Online]. Available: http://www.plasticsintl.com/materials.php.

[14] "3D Printing Systems – UP Box 3D Printer," [Online]. Available: http://3dprintingsystems.com/products/3d-printers/up-box-3d-printer/

[15] "Colido 2.0 Plus Impresora 3D | Innovación en la impresión 3D," [Online]. Available: http://colido.es/colido-20-plus

[16] J. a. B. F. Itten, The elements of color, New York: Van Nostrand Reinhold Co, 1970.

[17] "3D printing with TPU," 2016. [Online]. Available: http://www.shapingbits.com/3d-printing-guide/tpu-3d-printing/

[18] "IDC: Smartphone OS Market Share," [Online]. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[19] "Getting Started | Android Developers," [Online]. Available: http://developer.android.com/training/index.html

[20] "Android Studio Overview | Android Developers," [Online]. Available: http://developer.android.com/tools/studio/index.html

[21] "LAB," [Online]. Available: http://www.lab.upc.edu

[22] "Arduino - ArduinoBoardNano," [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardNano.

[23] Ibrahim, Microcontroller projects in C for the 8051, Oxford: Newnes, 2000.

[24] "I2C and SPI protocols | Byteparadigm". Availabe: http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/