Recover Underwater Position by Acoustic
RUPA

Tristan Dewalle
Intern in DCNS Energies

From 06/03/2017
To 20/07/2017

Table des matières

Introduduction

RUPA is an OpenCPN plugin, designed during my internship in DCNS Energies. The aim of that plugin is to have a friendly user HMI to locate underwater structures. To make that tool more efficient, I extended it to the management of the life of campaigns of measurement. In that manual, I will describe you how to use it (quite simple, don't worry) and if someone want to use it to improve it, in the "Developper part" I will explicit what my logic was.

Ho, by the way, I'm french, so sorry if sometimes my english is too bad (don't hesitate to contact me in this case).

User part

I hope this tool is quite easy to use, but here are some instructions, just in case.

Please, refer to the glossary if you're not sure what I'm talking about.

About the devices

About software, RUPA was developped with :
— Linux Mint 18.1 Serena
— OpenCPN 4.4.0
About the hardware :
— A deck unit from Sonardyne
— A Transponder from Sonardyne
— A FTDI (to connect computer and deck unit via USB)
— A GPS from trimble (just for information)

Installation

You have to install :
— opencpn (not needed for compilation, but as it's a plugin for it...)
— gcc-multilib
— g++
— autotools-dev
— cmake
— wx3.0-headers
— liwxgtk3.0-dev
— libghc-zlib-dev
— gettext
— libbz2-dev
— libmysqlcppconn-dev

— mysql-client
— mysql-server

You have to install ftd2xx too. To do that :

— go to http://www.ftdichip.com/Drivers/D2XX.htm
— download the good version (actually in the linux row, and for me the third or fourth column)
— go where you downloaded it
— extract it
— open a terminal where the folder is extracted
— sudo cp ./releases/build/lib* /usr/local/lib
— cd /usr/local/lib
— sudo ln -s libftd2xx.so.1.1.12 libftd2xx.so
— sudo chmod 0755 libftd2xx.so.1.1.12

If you want more explanation, please, look at the "FTDI Drivers Installation Guide for Linux".

Before launching it, you have to use the database :

— mysql -u root -p
— type the password you entered during installation
— GRANT ALL PRIVILEGES ON rupa.* TO 'youruser'@'localhost' IDENTIFIED BY 'password'
— SET PASSWORD FOR 'youruser'@'localhost' = PASSWORD('')(this step is deleting the password but if you want to keep one, skip that step and set it in RUPA/src/RUPA_Utility.h, l.62)
— quit
— mysql
— create database rupa
— quit

Then build RUPA :

— cd /path/where/you/want/to/download/rupa
— git clone https ://github.com/DCNS-Energies/RUPA.git (if not already done)
— cd RUPA/build
— rm -rf * (to correct something that have to be fix no need for sudo)
— cmake .. .
— make
— sudo cp librupas_pi.so /usr/lib/opencpn
— cd ../database
— mysql rupa<schema.sql

Usage

Campaign

It is supposed you have clicked on the "RUPA" button, while you're in OpenCPN. Here, you have 2 tabs and 5 buttons. The "Current Campaign" tab list what campaigns are not finished and "Finished Campaign" list what campaigns are finished.

The "New Campaign" button allow you to enter the minimal information you use find usefull to describe a campaign (the geographical area, the name of th campaign). Be careful, at this time, it's not possible to edit that simply.

The "Delete Campaign" button remove all the datas you had got : what you entered thank's to the "New Campaign" button, but every measurement you should have done. Be really carefull, there is no way to recover what you erased (really, there is absolutely no tricks to do that).

The "Manage Campaign" button open a new window : "Manage Campaign" in wich you can see each structure.

The "Close" button close the window (yep, really ;-)).

The "(Un)Finish Campaign" button set a Campaign as finished or current, and move it to the corresponding table.

New Campaign

The "Campaign's Name" field let you name the campaign as you want.

The "Geographical Area" field let you indicate where the campaign takes place.

The "Install Later" button will save what you typed and show you the "Campaign" window.

The "Install Now" button will save what you types too, but here, it redirect you to the "Manage Campaign" window.

Manage Campaign

Developper part

How to improve RUPA

Customization panel

There can be several customization panels. One of the most important is the possibility to change circles colors (could be done by adding a column into the database, for the measurement, then, adapt a little bit the code.

Auto detection of the structure position

That operation is named *trilateration*, I will let a code doing that. I didn't integrate it to the plugin, because it implies to make calculation on earth, and I can't spend to much time on it (it's not at all a "must have" for my project) I took the code frome someone else, using Eigen, the code is really small, so I let you have an eye on it. I'll put it in th "improvement" directory. Feel free to do what you want with that.

Modify Campaign parameters window

As underlined in the user guide, there is no possibility to modifiy the parameters of a campaign (even if there is realy little chance that it change, it could be nice to have the possibility to do so.

Temporary Measurement

Make it possible to have a temporary measurement, probably linked to a shortcut, to make it lighter to use for who don't care about campaigns (for example for divers who just have to recover something).

RS232

Make it possible to configure the RS232 bus differently, and to choose what message to sent (if using other device). More, making instead of scanning for a message, making an interrupt-callback system. I didn't do that, because, actually, I don't know how to do that (I tried a little bit with the boost library but I failed to catch a value changement), but I think the current way to detect reception (by scanning) is a really dirty way to do that.

Glossary

Campaign : it regroups every single structure in one Geographical area, for a type of survey (feel free to interprate it as you want of course).

Structure : it's an object on wich is fixed at least one transponder, and one sensor.

Burst : a collection of measurements.

Measurement : The distance between the transponder and the deck unit.

Deck Unit : Human-machine interface to communicate with the transponder