

ESTIMATION OF THE THIELE-SMALL PARAMETERS IN MOBILE APPLICATION USING AS LINK PURE DATA PROGRAMMING AND MICROSOFT VISUAL STUDIO

Alexander Calva Romero, Ing¹, Héctor Merino Navarro, Ing²., Carlos Mayorga, Mas³, and Jonathan Ortiz, Mat⁴.

¹ Facultad de Ingeniería y Ciencias Agropecuarias, Universidad de las Américas, Ecuador, ucalva@udlanet.ec

² Facultad de Ingeniería y Ciencias Agropecuarias, Universidad de las Américas, Ecuador, h.navarro@udlanet.ec

³ Facultad de Ingeniería, Pontificia Universidad Católica del Ecuador, Ecuador, cjmayorgaa@puce.edu.ec

⁴ Departamento de Formación Básica, Escuela Politécnica Nacional, Ecuador, jonathan.ortizc@epn.edu.ec

Abstract— This paper is about a multiplatform mobile application developed to calculate the Thiele-Small parameters of a loudspeaker using Pure Data. Pure Data is a programming language mainly used with computers, which is known for its versatility in creating electronic music and simulations of wave oscillators. In order to develop this multiplatform mobile application, it has been proposed a computer platform which can be compatible with Pure Data functionalities. In that sense, Microsoft Visual Studio has been chosen and, more concretely, the framework known as Apache Cordova. Pure Data was used to make the Thiele-Small parameters calculations by introducing mathematical formulas and Apache Cordova was used to show the results through the graphical interface of the mobile device. This procedure was performed using a file named WebPd, that works as a library code for compatibility between Pure Data and Apache Cordova. WebPd can be found on the official website of Pure Data library developers. Although there are already several mobile applications created using Pure Data, they have the limitation of being compatible with Apple devices only. So basically, this work was able to use a programming language for computers in the design of mobile device applications. In addition, it is capable to run in almost every operating system used by Smartphones (Android, IOS and Windows Phone). On the other hand, the Thiele-Small parameters mobile application can be considered a useful tool for the design of speaker boxes within academic purposes.

Keywords-- Pure Data, JavaScript, WebPd, Programming, Parameters.

I. INTRODUCTION

The rise of smartphones is one of the greatest revolutions in technology consumption during recent years. In addition, the success of other devices such as tablets has led to the development of numerous computer applications with many uses.

This publication describes the implementation of an app focused on the design of loudspeakers based on the Thiele-Small parameters. Although there are other methods to achieve this goal, the challenge of this investigation was to use

Pure Data to link it with smartphones. In that sense, it is required to explore a new way to integrate this language, initially made for audio synthesis, with the world of mobile devices.

II. DEVELOPMENT: PROGRAMMING TOOLS

In order to run this application, the user has to introduce the main Thiele-Small parameters in the software. After this, Pure Data will apply the design formulas and will return the results back. Since Pure Data is not made for mobile apps creation, it is necessary to use an external framework that emulates Pure Data code in mobile devices; in this case Apache Cordova and WebPd were used.

Apache Cordova is a framework in which the user may program mobile apps (IOS, Android, Windows Phone) using JavaScript, HTML and CSS. On the other hand, WebPd is an interface between JavaScript and Pure Data. This way, combining both Apache Cordova and WebPd it may port Pure Data applications to mobile devices using easily programming and markup languages such as the mentioned before.

At least, it is important to mention that Microsoft Visual Studio was used as the Integrated Development Environment (IDE) to facilitate all this work. All this workflow is showed in fig. 1. [1]

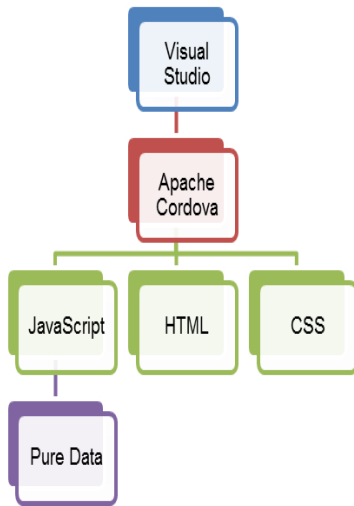


Fig. 1 Structure programming tools

A. Programming in Pure Data.

Pure Data is a graphical language for signal processing where the user may implement many different mathematical operations. Instead of using symbols like a common programming language, Pure Data uses boxes, with inlets and outlets, to represent operators. To give an example, in order to add two numbers, the user creates a binary addition box with two inlets (representing the two inputs values) and one outlet, which returns the output value.

Like any other programming language, Pure Data handles different types of information, like strings, numbers, Boolean algebraic values or other Pure Data specific tasks. Thanks to these capabilities the software is a powerful tool for signal processing operations.

To demonstrate this, the resonance frequency (1) of a box, in terms of the Thiele-Small parameters, has been implemented in the Pd as it is showed in figure

$$F_c = \frac{Q_{TC} * F_s}{Q_{TS}} \quad (1)$$

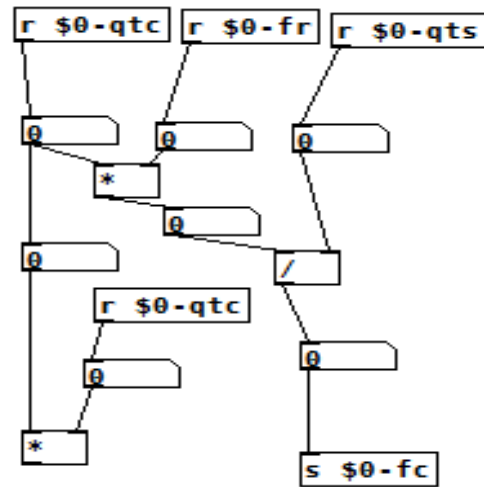


Fig.2 Programming in Pure Data to calculate Fc

In figure 2 the connections made between the objects to implement equation (1) are presented. The program receives from the user the main Thiele-Small parameters that characterize the system: QTS, which is the electromechanical parameter of the driver; QTC, which is the electromechanical parameter of the closed box; and fr, the resonance frequency of the driver. Pure Data receives the input values through JavaScript; next it resolves the mathematical operations and sends to JavaScript the result. [2]

B. Programming in JavaScript

JavaScript is a scripting language closely related to Web applications development. It is easy to use and it helps the programmers to develop mobile applications. Even though, the main functionalities of this program have been made on Pure Data, JavaScript was necessary in certain parts to link Pd with the mobile applications world that is, receiving and sending the values (2).

```
Pd.send(patch.patchId + '-qtc', [qtc]);
(2)
```

```
Pd.send(patch.patchId + '-qts', [qts]);
Pd.send(patch.patchId + '-fr', [fr]);
```

The result that Pd sends to the user through the graphical interface is showed by the following code (3):

```
Pd.receive(patch.patchId + '-fc', function (res)

    $("#entrada1").val(res);

});
```

Basically, these lines of programming made possible the communication between both languages and consequently, to develop the project.

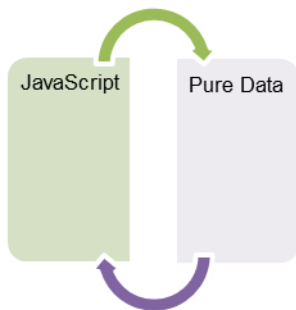


Fig.3 Communication between JS y Pd.

The following code (4) represents the calculation function to achieve a better understanding of these connections:

```
Function calculate()
(4)
{
var patch
$.get('./pd/name_file.pd', function (mainStr)
{
```

```
patch = Pd.loadPatch(mainStr);
var qtc = parseFloat($("#QTC").val());
var qts = parseFloat($("#QTS").val());
var vas = parseFloat($("#VAS").val());
var fr = parseFloat($("#FR").val());
```

//Send of values to Pure Data

```
Pd.send(patch.patchId + '-qtc', [qtc]);
Pd.send(patch.patchId + '-fr', [fr]);
Pd.send(patch.patchId + '-qts', [qts]);
Pd.send(patch.patchId + '-vas', [vas]);
```

//Result receptions

```
Pd.receive(patch.patchId + '-fc', function (res)
{
$("#entrada1").val(res);
});
}
}
```

C. WebPd to link Pd and JavaScript

It has already explained the meaning and the importance of linking JavaScript with Pure Data, but to make this possible was necessary another tool: WebPd. This point represents the main contribution of the project and publication. Since Pd is not software to make mobile apps as it has been mentioned before, JavaScript is not compatible with Pd by itself. In that sense, WebPd is the main axis of the link between Pure Data and Apache Cordova (through JavaScript) in order to make apps capable of running in the main operative systems of mobile devices.

Therefore, this is made by using a web API of audio as a tool of connection. This is a file created in JavaScript that contains lines of programming with functions needed for the compatibility between Pd and web environment. Even though the research led to other alternatives to achieve this link, the option of using WebPd was the easiest one as well as a new contribution to knowledge.

This innovation makes possible the interaction between graphical programming languages with web environment, opening a large variety of possibilities for future projects and, more concretely, using Pure Data for mobile apps creation. The method to include this file in the project is showed in figure 4. [3]

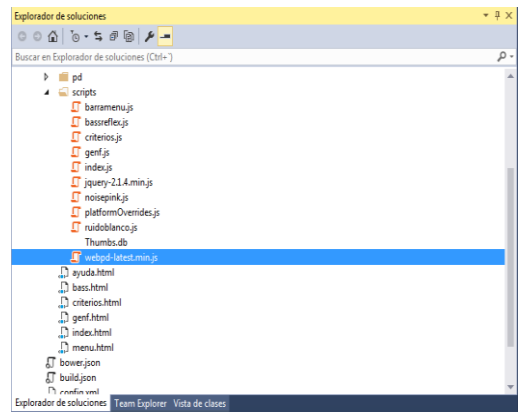


Fig.4 Including WebPd in the project for Apache Cordova.

D. Interface display on the Smartphone

Once the main objective is reached, next step is to make a graphical interface, where the user could introduce the data, as well as to check the results in the display. In this point, HTML, CSS and JavaScript will create the interactive web pages, while Apache Cordova will make possible to run the app in

different mobile operative systems (Android, IOS, Windows Phone).

Connecting audio signals between Pd and a Smartphone

In this part, it is important to remember that Pd was developed to work with audio signals. Considering to this, the present project included a tone generator in the loudspeakers app to give an example of the different possibilities when linking Pd with a Smartphone. This work was made in JavaScript as follows (5):

```
function sound()
{
var patch

$.get('/pd/sound.pd', function (mainStr) {
patch = Pd.loadPatch(mainStr);
var frec = parseFloat($("#FREC").val());
Pd.send(patch.patchId + '-sound', [frec]);

Pd.start();
})
};
```

To conclude this section of the article, figure 5 shows the final structure of the tools used to achieve the objective:

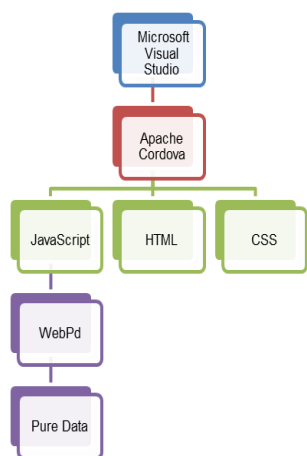


Fig.5 Programming tools used

III. CONCLUSIONS

It has been presented a new method of linking Pure Data with mobile devices. In order to give an example of application, it was developed an app that uses Thiele-Small parameters to design loudspeakers cabinets. To complete the functions of the

program, a tone generator was included too. Even though it is true there are other applications to do similar calculations, this app uses a graphical language to make easier the work of program developers.

About this, the research is a new contribution to the development of Pure Data software. It will help to the integration of this language to web environment through Microsoft Visual Studio and Apache Cordova.

It has been demonstrated that it is possible to send and receive information easily between Pd and a Smartphone, which can be used either for mathematical calculations or audio signal processing. It must be stated, however, that not all objects used in Pure Data are compatible with WebPd. For this reason it is recommended for programming mathematical operations.

In general terms, the application is very stable, light and easy to manage for users. Although it may seem obvious, during this investigation was also evidenced the relevance of using an updated version of Android operating system for a correct compilation, in the same way for IOS and Windows Phone operating systems.

The proposed application for this product is oriented to an educational and non-commercial context and it is available for free on the Play Store Android digital store. Loudspeakers design is an amazing world and the app pretends to help students to understand it. This is a first step and the idea is to improve the app in the future and offer a wide range of possibilities.

ACKNOWLEDGMENT

This research was supported by “Universidad de las Américas de Ecuador” and it should be mentioned as well the important contribution of the Systems Engineer Mr. David Cañar.

REFERENCES

- [1] Alarcón, J. PhoneGap o Apache Cordova ¿qué diferencia hay?. Campus MVP, Recuperado el 2 de septiembre de 2015 de <http://www.campusmvp.es/recursos/post/PhoneGap-o-Apache-Cordova-que-diferencia-hay.aspx> 2014.
- [2] García, C. Diseño de cajas acústicas. Electroacústica, País Vasco, España. [Versión electrónica], s.f. 7 de abril de 2015
- [3] GitHub. WebPd/ObjectList.md. Recuperado el 9 de agosto de 2015 de <https://github.com/sebpiq/WebPd/blob/master/OBJECTLIST.md> 2015
- [4] GitHub. Sebpiq/WebPd. Recuperado el 6 de agosto de 2015 de <https://github.com/sebpiq/WebPd> 2015
- [5] Kreidler, J. Programando Música Electrónica en Pd, Argentina. Recuperado el 5 de junio de 2015 de <http://lucarda.com.ar/pd-tutorial/index.html>. 2009.