Christophe Verbeken
Supervisor (Greece): George Kyranastasis
Supervisor (Belgium): Luc Van Huffel

Kavala, June 1999

Home Institution
Kaho St-Lieven
Departement Gent Campus Rabot
Gebr. Desmetstraat
9000 Gent
Belgium

Host Institution
Technological Educational Institution (TEI) of Kavala, Greece
Department of Electrical Engineering
Laboratory of Power Electronics
Prof. Dr. George Kyranastasis
Foreword

In my last year of my study, it was possible to do the final project work, during one semester, abroad. For me, it was a big challenge and at the same time a unique opportunity to leave my home and live in a completely different environment for several months.

At the beginning, I felt a little bit lonely without my family and friends in Belgium, but after a while, I started to learn and enjoy the way of living, in Greece. My environment at the TEI became so familiar to me that it started to be my second home. I made a lot of Greek friends here and my relationship with the other foreign students was at least exceptional. Together with the Socrates people, I did a lot of amusing activities and I visited the most interesting places in this country. My Greek adventure is one of the best I will ever have, in my live.

Thank you guys, for giving me this fantastic and unforgettable time!

To make this experience possible, I'd like to thank my parents and my supervisor in Greece, Mr. George Kyranastasis, who accepted me as a student in the TEI of Kavala, in Greece, to do my final project work. Also special thanks to my supervisor Mr. Luc Van Huffel and Mr. Dirk Hanselear of my home Institute in Belgium, who planned my journey, here, and assured me that I could come to Greece.

Thank you all!
Table of contents

Foreword .......................................................................................................................... 1
Table of contents .............................................................................................................. 2
1. Introduction .................................................................................................................. 4
   1.1. About Neapolis ................................................................................................. 4
   1.2. About my subject .............................................................................................. 4
   1.3. About this Report ............................................................................................. 5
2. Structure of Neapolis .................................................................................................. 6
   2.1. The different subprograms .............................................................................. 6
   2.2. The used Directories ....................................................................................... 7
3. Theory about the different Choppers ........................................................................... 9
   3.1. Introduction ....................................................................................................... 9
   3.2. Class A Chopper ............................................................................................... 10
      3.2.1. General .................................................................................................... 10
      3.2.2. Schematic ............................................................................................... 11
      3.2.3. Principle .................................................................................................. 11
      3.2.4. Waveforms of the different modes ............................................................. 13
   3.3. Class B Chopper ............................................................................................... 15
      3.3.1. General .................................................................................................... 15
      3.3.2. Schematic ............................................................................................... 16
      3.3.3. Principle .................................................................................................. 16
      3.3.4. Waveforms of the different modes ............................................................. 19
   3.4. Class C Chopper ............................................................................................... 21
      3.4.1. General .................................................................................................... 21
      3.4.2. Schematic ............................................................................................... 22
      3.4.3. Principle .................................................................................................. 22
      3.4.4. Waveforms of the different modes ............................................................. 27
4. Conversion of the theory to a suitable Visual Basic model .......................................... 31
   4.1. Opening Neapolis .............................................................................................. 31
   4.2. How are the Graphics drawn in Neapart 2 ....................................................... 31
   4.3. New Textfiles in Neapolis .................................................................................. 35
   4.4. New modules in Neapart 2 .............................................................................. 36
      4.4.1. Following sequence in a chopper module to draw ...................................... 36
   4.5. Changing the theoretical model ........................................................................ 38
4.5.1. Inductor's equivalent .....................................................................................38
4.5.2. The equivalent circuits and equations of the choppers .................... 38
5. Further changes in the interface to put the chopper in Neapolis .......... 43
  5.1. In Neapart 1 ..............................................................................................43
  5.2. In Neapart 2 ..............................................................................................44
6. My implemented parts at run time ..............................................................46
  6.1. Of the A Class Chopper ............................................................................46
  6.2. Of the B Class Chopper ............................................................................47
  6.3. Of the C Class Chopper ............................................................................48
7. Conclusion ..........................................................................................................................50
1. Introduction

1.1. About Neapoliis

The city name "Kavala" used to be "Neapoliis", and was the main port the ancient settlement of Philipi.

But also an educational program, which simulates Power Electronic Circuits, Electrical Motors and Motor Drives inherit the name "Neapoliis", simply because this software was developed in the Power Electronics Laboratory in the TEI of Kavala by Prof. Dr. George Kyranastasis.

The idea and the main structure already existed for several years, and was already implemented in an earlier version of the program. But many improvements resulted in a completely new version of Neapoliis. The main objective of the new version is to make the user interface easy to understand and easy to work with.

Since its existence, Neapoliis is continuously changed by students under supervision of Prof. Dr. George Kyranastasis in order to improve the existing program, or to add new devices or simulation facilities to the program.

The program is developed in order to help students to understand the behavior of several Power Electronic Devices, they learned about in the theoretical lessons. The program can be used to show the variables of the circuits, without any danger for the device or the users and make experiments changing the different circuit parameters. The curves on the screen are easier to understand than the simple numbers on the measurement instruments.

1.2. About my subject

When I started this project, I never programmed in Visual Basic. I learned the program language by reading a VB manual that I brought along from Belgium, and by making several test programs in this Visual interpreter to understand how it works. Afterwards, I was capable to start with my real subject.
I had to add some new devices in Neapolis, and change the interface in this way that those new devices fitted perfectly in the existing program of Neapolis. The three devices I implemented are the power converters:

- Chopper Class A
- Chopper Class B
- Chopper Class C

To implement those Chopper converters, I needed the first two parts of Neapolis (the MainMenu, the Definition part and the Simulation part)

I used the following method to finish my project successfully:

- Study of the program code of Neapolis, on how the existing devices work.
- Detecting where I had to add the new code, to implement my devices.
- Generate the necessary code (in Visual Basic) and the different text files for all the new devices.
- Testing the new written code and making some changes, if necessary, to improve the working device.

### 1.3. About this Report

In this report I try to explain the new implemented devices as follow:

- First, I mention how the different Chopper Converters works, by using the schematics, waveforms of all the variables and some formulas to explain the waveforms.
- Then I show how I convert the theoretical idea into useful Program code by explaining all steps to draw a graphic in Neapart 2, and a further working out of the theoretical formulas until they can directly by implemented in the program code.

I don't explain the Visual Basic-code that I wrote and changed because it isn't interesting for the reader, and it would be too much to print. In the new code that I added, I wrote enough comments to understand the way it works. This is easier for the next student who will analyze the code.
2. Structure of Neapolis

2.1. The different subprograms

Figure 2.1: The global structure of Neapolis
Figure 2.1 shows the main structure of Neapolis, and how it works. All the vital text files are mentioned which are used to store the major part of the information, to switch from one subprogram to the another within the software packet of Neapolis.

The most important text files are:

- **PD-files**: the name of all those files starts with "PD" and contains all the default data of the load, supply, program, model, graphics. These files are loaded when Neapart 1 is started.

- **SY-files**: the name of all those files starts with "SY" and is derived from the PD-files because they contain some information about the supply, the model and the load that is used in Neapart 2.

- **PR-files**: the name of all those files starts with "SY" and contains the information about the TimeStep and the number of periods that has to be drawn in the Simulation part. You can also find the information about the axis, the axis names and there range.

### 2.2. The used Directories

The main directory of Neapolis is "NEA4". In this directory you can find to sorts of folders:

- Folders of all the required data that the 4 exe-files of Neapolis uses.

<table>
<thead>
<tr>
<th>Name of the Folder</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datafile</td>
<td>The PD-files</td>
</tr>
<tr>
<td>Drawings</td>
<td>Circuit Pictures shown by Neapart3</td>
</tr>
<tr>
<td>Text/TextsDef</td>
<td>Text for Neapolis and Neapart 1</td>
</tr>
<tr>
<td>Text/TextsSys</td>
<td>Text for Neapart 2</td>
</tr>
<tr>
<td>Text/TextsRes</td>
<td>Text for Neapart 3</td>
</tr>
</tbody>
</table>
Figure 2.2: This table shows some folders under "C:/Nea4/"

Next to the main directory, another important directory, with the name of "Neafiles" can be found. In this directory you can found all the dat-files, the SY-files, the PR-files and the EX-files that the different subprograms of Neapolis uses.

- Folders which contains all the VB-code to develop the program.

<table>
<thead>
<tr>
<th>Name of the Folder</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convert</td>
<td>Forms of the convertors</td>
</tr>
<tr>
<td>Motors</td>
<td>Forms of the motors</td>
</tr>
<tr>
<td>Drivers</td>
<td>Forms of the drivers</td>
</tr>
<tr>
<td>Fileopen</td>
<td>Forms to open the usefiles</td>
</tr>
<tr>
<td>Forms</td>
<td>Forms of Neapolis, Neapart 2, Neapart 3</td>
</tr>
<tr>
<td>Def-Sys</td>
<td>Forms of Neapart 1</td>
</tr>
<tr>
<td>Modules</td>
<td>All the modules</td>
</tr>
</tbody>
</table>

Figure 2.3: This table shows the other folders under "C:/Nea4/"
3. Theory about the different Choppers

3.1. Introduction

For many industrial applications, it is required to convert a fixed-voltage DC source into a variable-voltage DC source. A chopper converts directly from DC to DC and is also known as a DC-to-DC converter. It can be used to step-down or step-up a DC-voltage source.

Choppers are widely used for traction motor control in much machinery. They provide a smooth acceleration control, high efficiency, and a fast dynamic response. Choppers can be used in regenerative braking of DC motors to return energy back into the supply, and these feature results in energy savings for transportation systems with frequent stops. Choppers are used in DC voltage regulators, and also used, in conjunction with an inductor, to generate a DC current source.

Choppers may be classified according to the number of quadrants of the $u_o$-$i_o$ diagram in which they are capable of operating (shown in Figure 3.1).

Figure 3.1: Classification of choppers by quadrants of operation
3.2. **Class A Chopper**

3.2.1. General

This converter is the basic Power circuit of a step-down single-quadrant chopper. "Step-down" means that the average energy is flowing from the source to the load-circuit. The term "single quadrant" signifies that the circuit variables $u_o$ and $i_o$ occur only in the first quadrant of the $u_o$-$i_o$ diagram (shown in Figure 3.2).

![Figure 3.2: Quadrant of operation of the Class A chopper](image)

The load-model can be separated into 3 circuit elements to show the model of a motor.
3.2.2. Schematic

![Schematic of the Class A Chopper](image)

Figure 3.3: Schematic of the Class A Chopper

3.2.3. Principle

Here are the different working modes explained, how the Class A Chopper works.

**Mode 1:** $T_{on} = 0 < t < kT$

Thyristor $T_h$ is *turned on*
Diode $D$ is *turned off*

![Schematic of the Class A Chopper in Mode 1](image)

Figure 3.4: Schematic of the Class A Chopper in Mode 1
In Figure 3.4, the equivalent circuit of working-mode 1 is shown. There you can see that the output voltage $u_o$ is equal to the supply voltage $u_i$, and that the current flows into the load. The input current $i_i$, which has the same value as the output current $i_o$, increases continuously until the time reached the value of $kT$. Then the input current reached $I_2$, as shown in the formula, below.

$$I_2 = I_i e^{-kT/L} + \frac{U_o - E}{R} \left(1 - e^{-kT/L}\right)$$

**Figure 3.5: Formula of the final current at the end of Mode 1**

**Mode 2:** $T_{off} = kT < t < T$

Thyristor $T_n$ is *turned off*
Diode $D$ is *turned on*

![Figure 3.6: Schematic of the Class A Chopper in Mode 2](image)

In Figure 3.6, the equivalent circuit of working-mode 2 is shown. There you can see that the output voltage $u_o$ is zero, and that the current flows into the load. The output current $i_o$ decreases continuously until the time reached the value of period $T$. Then the input current reached $I_1$, as shown in the formula, below.
\[ I_1 = I_2 e^{-(t-k)\frac{R}{L}} + \frac{E}{R} \left( 1 - e^{-(t-k)\frac{R}{L}} \right) \]

Figure 3.7: Formula of the final current at the end of Mode 2

3.2.4. Waveforms of the different modes

Figure 3.8: Discontinuous Output current of the A Class Chopper
The Figures 3.8 and 3.9 are showing two different current waveforms. If $t_{on}$ is short in relation to the period $T$, then discontinuous current will occur and the current waveform will decay to zero during mode 2. If $t_{on}$ is longer in relation to the period $T$, then continuous current will occur, so that the load current will not decay to zero during mode 2, but will decrease until the thyristor $T_h$ turned on again.
3.3. **Class B Chopper**

3.3.1. General

This converter is the basic Power circuit of a step-up single-quadrant chopper. "Step-up" means that the average energy is flowing from the load-circuit to the source. The term "single quadrant" signifies that the circuit variables $u_o$ and $i_o$ occur only in the second quadrant of the $u_o$-$i_o$ diagram (shown in Figure 3.10).

![Figure 3.10: Quadrant of operation of the Class B chopper](image)

The load-model can be separated into 3 circuit elements to show the model of a motor.
3.3.2. Schematic

![Diagram of a Class B Chopper](image)

Figure 3.11: Schematic of the Class B Chopper

3.3.3. Principle

Here are the different working modes explained, how the Class B Chopper works.

**Mode 1**: \( T_{on} = 0 < t < kT \)

Thyristor \( T_h \) is *turned off*

Diode D is *turned on*
In Figure 3.12, the equivalent circuit of working-mode 1 is shown. There you can see that the output voltage $u_o$ is equal to the supply voltage $u_i$, and that the current flows out of the load, back to the source. The input current $i_i$, which has the same value as the output current $i_o$, decreases continuously until the time reached the value of period $kT$. Then the input current reached $I_2$, as shown in the formula, bellow.

$$I_2 = I_i e^{-\frac{kT R}{L}} + \frac{U_o - E}{R} \left(1 - e^{-\frac{kT R}{L}}\right)$$

**Figure 3.13: Formula of the final current at the end of Mode 1**

**Mode 2:** $T_{off} = kT < t < T$

Thyristor $T_h$ is **turned on**

Diode $D$ is **turned off**
In Figure 3.14, the equivalent circuit of working-mode 2 is shown. There you can see that the output voltage $u_o$ is zero, and that the Electro Magnetic Field (EMF) drives current through inductor $L$ of the load. The output current $i_o$, increases continuously until the time reached the value of $T$. Then the output current reached $I_1$, as shown in the formula, bellow.

\[
I_1 = I_e e^{-\frac{(1-k)R}{R}} - \frac{E}{R} \left(1 - e^{-\frac{(1-k)R}{L}}\right)
\]

**Figure 3.15: Formula of the final current at the end of Mode 2**
3.3.4. Waveforms of the different modes

Figure 3.16: Discontinuous Output current of the B Class Chopper
The Figures 3.16 and 3.17 are showing two different current waveforms. If $t_{on}$ is short in relation to the period $T$, then discontinuous current will occur and the current waveform will increase to zero during mode 2. If $t_{on}$ is longer in relation to the period $T$, then continuous current will occur, so that the load current will not increase to zero during mode 2, but will increase until the thyristor $T_n$ turned on again.
3.4. **Class C Chopper**

3.4.1. General

This converter is a combination of the A Class- and the B Class Chopper and gives a satisfactory method to obtain regenerative braking and driving, of an application. The Power circuit shows a step-up step-down double-quadrant chopper. The term "double-quadrant" signifies that the circuit variables $u_o$ and $i_o$ occur in the first and the second quadrant of the $u_o-i_o$ diagram (shown in Figure 3.18).

![Quadrants of operation of the Class C chopper](image)

**Figure 3.18: Quadrants of operation of the Class C chopper**

The load-model can be separated into 3 circuit elements to show the model of a motor.
3.4.2. Schematic

Figure 3.19: Schematic of the Class C Chopper

3.4.3. Principle

Here are the different working modes explained, how the Class C Chopper works.

**Mode 1:** $T_{on} = 0 < t < kT$ and output current $i_o < 0$

- Thyristor $T_{h1}$ is *turned off*
- Thyristor $T_{h2}$ is *turned off*
- Diode $D_1$ is *turned on*
- Diode $D_2$ is *turned off*
In Figure 3.20, the equivalent circuit of working-mode 1 is shown. There you can see that the output voltage $U_o$ is equal to the supply voltage $u_i$, and that the current flows out of the load, back to the source.

The input current $i_i$, which has the same value as the output current $i_o$, increases continuously until the time reached the value of $kT$ or the output current $i_o$ reached zero. If the time reached the value of $kT$ then the input current reached $I_2$, as shown in the formula, bellow.

$$I_2 = I_1 e^{-\frac{kT \rho}{L}} + \frac{U_0 - E}{R} \left(1 - e^{-\frac{kT \rho}{L}}\right)$$

**Figure 3.21: Formula of the final current at the end of Mode 1**

Otherwise the output current $i_o$ reached zero so that $I_2 = 0$

**Mode2**: $T_{\text{on}} = 0 < t < kT$ and output current $i_o > 0$

- Thyristor $T_{h1}$ is *turned on*
- Thyristor $T_{h2}$ is *turned off*
- Diode $D_1$ is *turned off*
- Diode $D_2$ is *turned off*
In Figure 3.22, the equivalent circuit of working-mode 2 is shown. There you can see that the output voltage $u_o$ is equal to the supply voltage $u_i$, and that the current flows into the load.

The input current $i_i$, which has the same value as the output current $i_o$, increases continuously until the time reached the value of $kT$. Then the input current reached $I_2$, as shown in the formula, bellow.

$$I_n = I_i e^{-\frac{kT R}{L}} + \frac{U_0 - E}{R} \left( 1 - e^{-\frac{kT R}{L}} \right)$$

**Figure 3.23: Formula of the final current at the end of Mode 2**

**Mode 3:** $T_{off} = kT < t < T$ and output current $i_o > 0$

Thyristor $T_{h1}$ is turned off
Thyristor $T_{h2}$ is turned off
Diode $D_1$ is turned off
Diode $D_2$ is turned on
In Figure 3.24, the equivalent circuit of working-mode 3 is shown. There you can see that the output voltage $u_o$ is zero, and that the current flows into the load.

The output current $i_o$, decreases continuously until the time reached the value of period $T$ or the output current $i_o$ reached zero. If the time reached the value of period $T$ then the input current reached $I_1$, as shown in the formula, below.

$$I_1 = I_2 e^{-\frac{(1-k)u^R}{L}} + \frac{E}{R} \left(1 - e^{-\frac{(1-k)u^R}{L}}\right)$$

Figure 3.25: Formula of the final current at the end of Mode 3

Otherwise the output current $i_o$ reached zero so that $I_1 = 0$

**Mode 4:** $T_{off} = kT < t < T$ and output current $i_o < 0$

- Thyristor $T_{h1}$ is **turned off**
- Thyristor $T_{h2}$ is **turned on**
- Diode $D_1$ is **turned off**
- Diode $D_2$ is **turned off**
In Figure 3.26, the equivalent circuit of working-mode 4 is shown. There you can see that the output voltage $u_0$ is zero, and that the Electro Magnetic Field (EMF) drives current through inductor $L$ of the load. The output current $i_o$, decreases continuously until the time reached the value of period $T$. Then the output current reached $I_1$, as shown in the formula, bellow.

$$I_1 = I_1 e^{-\left(1-k\right)\frac{R}{L}} - \frac{E}{R} \left(1 - e^{-\left(1-k\right)\frac{R}{L}}\right)$$

**Figure 3.27: Formula of the final current at the end of Mode 4**
3.4.4. Waveforms of the different modes

Figure 3.28: Waveforms of the C Class Chopper with Duty-Cycle less then 50%
Figure 3.29: Waveforms of the C Class Chopper with Duty-Cycle just 50%
Figure 3.30: Waveforms of the C Class Chopper with Duty-Cycle more than 50%
The Figures 3.28, 3.29 and 3.30 are showing three different current waveforms. If $t_{on}$ is shorter than the half of period $T$ (Duty-Cycle less than 50%), then the current waveform will decrease under the zero-line, in steady-state (Figure 3.28). If $t_{on}$ is longer than the half of period $T$ (Duty-Cycle more than 50%), then the current waveform will increase above the zero-line, in steady-state (Figure 3.30). If the Duty-Cycle is just 50%, then the zero-line will be just in the middle of the current waveform (Figure 3.29).
4. Conversion of the theory to a suitable Visual Basic model

4.1. Opening Neapolis

In the MainMenu, you can choose a different system than the one who is actually active. Afterwards you go to the Definition part of the program, where:
- you can select a different type of supply or load
- change the default values of the system in use
- select and configure the graphics that has to be shown

All the changes in the Definition part are saved in the corresponded text files (Load.dat, Supply.dat...).

And finally, the simulation system can be opened to draw the respective graphics. The Simulation parts uses 2 major text files that are made when Neapart 2 is started. Those 2 text files (SY-file and PR-file) contains all the necessary information to draw all the graphics of the chosen system.

It is possible to change some default values (frequency, duty-cycle...) before the simulation is started. When the graphics are drawn, you can continue the simulation, or go back to the MainMenu, and start the Result Processing.

4.2. How are the Graphics drawn in Neapart 2

When Neapart 2 is started, all the necessary information is loaded from the text files (SY-file) according to the chosen system. Then the respective variables, that the simulation program needed to draw the graphics, are calculated. The most important ones are:

- **TimeStep**

Because any computer program can't draw an analog, fluent signal on the screen, you have to divide every graphic into small areas. Every time that the respective equations are calculated with the new value of TimeStep, a new small area can be added to the previous ones (shown in Figure 4.1). The smaller the little areas are, the sharper the graphic is.
Adding Choppers in Neapolis 4.0

Christophe Verbeken

\[
Timestep = \frac{1}{\text{Chopfreq} \times \text{NumPeriodPoints}}
\]

**Figure 4.1: Formula for the Timestep**

- **ChopOnTime**: This variable gives the period of time between the beginning of every period \( T \) and the moment, during the period \( T \), where an active switch (Thyristor) is fired (shown in Figure 4.3). This short time depends on the duty-cycle of the control signals.

\[
\text{ChopOnTime} = \frac{\text{PeriodTime} \times \text{ChopControlVoltage}}{100}
\]

**Figure 4.2: Formula for the Duty-Cycle**

**Figure 4.3: Graphic of one period**
2. Resampling
3. Using the bitrate-driven calculator
4. System requirements
5. Reporting problems

Set Encode Properties

Encoding properties specify the encoding bitrate, the output file format, and the output directory. As files are added to the program, they take on the default encoding properties. You can save time by setting the default encode properties prior to adding the files. Clicking the Encode Properties button with no files selected allows you to set the default encode properties.

![Fig. 3.1. Configuration of compression.](image)

Encoding properties can be applied to files individually or to entire file groups. To select a single file click on the file name. To select multiple files hold the SHIFT or CTRL keys while clicking on the file name. Click on the Encode Properties button to bring up the Encode Properties dialog.
Add Files

Files can be added to the program using any of the following methods:

1. It drags files and drops them on the program icon. The program will immediately open.
2. While the program is running drag files and drop them on the application window.
3. Use the Add Files button or menu option to bring up the standard file dialog. You can select more than one file by holding the SHIFT or CTRL keys while clicking on the file name.

Encode

After you have added all files to be processed and set the encoded properties, you are ready to begin the encode process. Click the Encode button to start. A progress dialog is displayed showing you the encode progress. You can minimize the application by clicking the minimize button in the progress dialog. You may use the PC while the encoding takes place. This may however increase the encode time.
Fig. 3.3. Encode part

Selecting the Bitrate

The bitrate represents how much data is used to represent one second of audio. The selected bitrate determines the audio quality. The higher the bitrate, the better the audio sounds. At 128 Kbit/sec, the audio has near-CD quality. The bitrate also determines how the encoded file can be streamed (divided by packets).

For example if you want to encode audio files, which can be streamed by people using 28.8 Kbps modems you would need to select a bitrate of 20 Kbits/second or less.

System Requirements

Audioactive Producer is compatible with Windows 95 and Windows NT 4.0.
**MP3 COMPRESSOR**

This is a simple software that can compress wave files format to MP3 or WAV-MP3 format (.wav). The possibilities that it offers are very limited in this field like for example, the speed to compress the files. It was really slow, that's why I will not explain anything else about this software.

![Fig. 3.4. MP3 Compressor program](image-url)
CHAPTER 4. NEAPOLIS WAV-MP3 PLAYER

4.0. INTRODUCTION

Finally the music has arrived to Neapolis 4.0. Now Neapolis’s users can enjoy with their favorite music during the work. For that they have to convert the CD audio format.

The users can grab CD audio and after that encode their own music to WAV-MP3 and then play in the Multimedia WAV-MP3 Player. That’s all.

The users don’t have to be worried about how much space these files take in the hard disk or in one CD. Each song takes something like 3 or 4Mb supposing that the length of song is over 3 or 4 minutes. The size of the song files is not a big problem for the PC’s users. Then, a player is the only thing they need.

Just when the users open the main window of the player for the first time, they know how they can get the music and reproduce them. This is the best feature of Neapolis Multimedia player; it’s easy to learn how to use it.

The users can select the list of the songs they want to listen to; they can also select from the same list which songs are going to be played. Everything depends on the mode the user has selected. There are several possibilities that this player gives to the users.

One of the biggest differences with other players is that the users can make their own groups of songs and access to them in a simple way. Thus they don’t have to select every time their favorite songs from the folders, because perhaps all the songs they want are not in the same folder.

As you can see with this player you do almost everything you want.

In this chapter I will explain everything about my work in this player. First I will start with the user manual. Afterwards I will tell how I did it, which is the structure of the source code and which were the major difficulties I found programming.
4.1. USER MANUAL

4.1.1. INTRODUCTION

DRIVES ACCESS

The first thing the users have to do if they want to listen to WAV-MP3 music in Neapolis is make click on “Browse” button. This button is on the left of the player placed on the fifth position in the “Setting Buttons”.

That button opens the rest of the window, which is dedicated to look for the song files there are in the drives. When the windows is opened the name button changes to “Back”. Thus the users know that if they click now on this button again this part of the window disappears from the main one.

The users can do the same action selecting the “Edition” Menu and clicking on the Browse option.

Both actions will show the browse part of the player on the screen. This part is added on the right of the main player window.

Now, the song files are selected from every available drive in the computer. When the users select one folder from some drive, its “.wav” (MP3) files are showed on the “source list”.

The users can select the files they want to add in the play list clicking on the list. “Select All” button is done to select all the files from the folder selected. There is also another option in “Edition” Menu that adds directly every file from the selected folder to the play list.

The users can take WAV-MP3 files from every drive. If the drive is not available, the program displays one message and it selects the drive “C” by default.

The users can work with the Browse window opened or closed. It only depends on the taste of the user. The player will work in the same way.

This player can reproduce all the files with Wave format, EVEN if the real format is MP3. Saying the truth this player is a WAV player that needs to have installed one codec to play WAV-MP3 format in Windows.

ADD THE SONG FILES

Once the users have selected the folder from where they want to take the song files, they must add them into the play list (placed in the center of the player, on the right of buttons list).

There are several different manners to add the files:

- Using “Add” button.
Doing double click on the files one by one.

The first one adds the files that are selected in the source list. You can use the second one when you have your songs stored in one specific folder and you want to add every file into the list.

Finally, when you wish to add one by one you do double click on the file and it is immediately added into the list. That's rather faster than select the file and then do click on the “add” button.

**SELECTION MODE**

Selection mode combines the possibilities of playing all the song files added on the reproduction list or only the files that are selected or the files the program wants to reproduce by randomize system.

The play mode opens a lot of advantages for the Neapolis users.

The users don’t have to be loading the files every time they want while the player is working. They choose the songs and then click on the option mode they want to work. So the mode selection system makes the user works with the player in an easier way.

Doing click on the option button you activate this mode and disactivate the other ones, therefore only one mode has to be working.

“Play All” makes possible to play in order all the songs that there are in the reproduction list, even if there are files that are already selected. The program doesn’t care about them.

Just the opposite happens with another option that is called “Play Only Selected”. This option makes possible to play the songs that the user has selected previously in order. It permits the user chooses the MP3 (.wav) files to play not removing them from the list. However, the user has to take care with the selected files, since they can be selected from the target list to delete them.

The last mode option that you can find is “Random Play”. This option is very useful, because the drive has stored the song files in one specific order in the folders, so the user will always listen to the songs in the same order and it is not very amusing. Then this mode makes possible to play all the songs that there are in the play list without a predetermined order. The program creates a random to choose which is the file to be reproduced, immediately is showed selected on the list and played.

Until every song has been played there is no song that was played twice. After that, another cycle starts again automatically.

“Random Play” mode shows two selected files, the file that is playing and the next one.
DELETE FILES FROM REPRODUCTION LIST

The users can arrange the reproduction list like they exactly wish. Sometimes it is very necessary to delete some files from the list, maybe because the user has added some file by mistake or because the user has already listened to all the songs, then he wants to change the configuration of the list. It is possible to have more reasons, everything depends on the user wants to do.

To delete the song files we can find two different ways. The first one is to delete only the files that the user has selected previously. The other one clears every file from the target list without stopping the reproduction, if it is playing, of course.

PLAYBACK

Once the user has finished of adding files in the reproduction list and has selected the work mode, it is time to start playing music.

There are several buttons that make possible to play with the files. These are “play”, “pause”, “stop” and “next” buttons. Each one has a specific and important function.

The user can start playing the file showed below the play list. Once the first song is playing, the user can decide to pause it. After pausing the music the timer label shows the played time flickering until the user does click again on the pause button. The music plays again.

The user has the possibility to stop the reproduction doing click on the “stop” button. Thus the song stops and the playing cycle starts from the beginning again. Now the Multimedia player is waiting for clicking on “play” button showing on the bottom of the player the next song to be played. Its length is on the “Timer Display”.

If the user is bored of listening to one song, the program is available to pass to the next song on the list (everything always depends on the mode that is working). This option can also be used to check all the song files from the list.

Usually, when the users are working in Neapolis, they don’t take care about the Neapolis music player. They only care about listening to music while they are working, so they don’t want to go back to the player to do click on the “play” button. Then “Continuous Play” option is the tool the player need. The songs from the play list are played cyclically. Therefore, the user can work not worrying about the player.

As data, “Random Play” option always has got continuous play check box working.

The last option we can find is “Volume Control”. This option makes it possible to turn the volume up or down.
"Set Music Type" is one of the best functions that the user can find in Neapolis Multimedia Player. With this function the user is able to create song file groups. Thus the user doesn't have to touch the location of the files on the drive's folders to play the files in the way he wants, he can go directly to the group that was made previously.

The user can also remove the new groups that are included on the list in a very easy way. The program has got already made some defined groups, which can not be modified. If the user wants to change something in these types, he has to go to Windows Explorer and move or delete everything he wants.

The groups haven't got any limit of song files, then the user can choose every song he wishes and includes in his music group.

You can find this option in "Edition" menu, like “Set Music Type”.

With this option another window is opened to set the types. Working in this window is very easy. The user only has to select on the play list the songs that he wants to have in the new group. Then, selecting the option “Set Music Type” the program is displayed. Now, the user can do two different things. The first, create a group with the files he has added on the reproduction list, or the second one, remove one of the groups already created previously.

Once the user has created one new group, it is immediately included on the list of the groups in “Music Type” menu. So now, this group is available for the user on the "Music Type" menu. Clicking on this option the user will activate the groups of music made before.

When the user has removed one new group, it disappears from the list of the group, but now the user can remove another one without getting out from the setting music type program. If there is no more groups to be removed the program is closed automatically, since there is nothing to do on it.

**VISIT THE HELP**

The help is only made to give information to the beginner user about which are the different parts of the Neapolis Multimedia player. Thus he has a basic idea about the player to start playing.

**TIMER DISPLAY**

This is one of the less necessary options on the player, however it is very useful to know the length of the song that is playing and which is the position of the playing file in minutes and seconds.
Doing click on the numbers of the timer the user can change its mode, the timer will increase or decrease the time.

4.1.2. PARTS of MULTIMEDIA PLAYER

**BROWSE PART**

The Browse is divided in three important parts. These parts are the Drive, the Directory tree and the WAV File List (source list). The user can take files with "wav" extension from every drive available to play them.

Below you can see the picture of the browse part.

This part of the player can be hide whenever the user wants. The player works in the same way if the Browse is opened or closed, but the user always has the chance to work with it.

Let’s see all the parts one by one, although they are connected each other intimately.

**DRIVE**

The drive makes possible to take WAV-MP3 files from every drive of our computer. You don’t have to have the files only in the hard disk. Thus it can be used for another much more important applications.
That is one of the major preoccupations the users have when they install software on their hard disk.

The List box shows every drive that there is available in the computer where the user is working.

This List box is normally closed; it shows the selected drive. Doing click on its button you get the complete list of all the drives, so you can choose from which drive you want to take the files.

**DIRECTORY TREE**

This is the typical directory tree the user can find in every application. Let’s remember how it works:

- The directory tree makes possible to choose (only in selected drive) the folder where you have stored the song files.
- With this list you can navigate through all the folders you have in the drive that you have selected.
- Doing double click on some folder you will open the files with “.wav” extension in it.
The user can observe which are the files with "wav" extension that the selected folder contents. WAV is the extension the programmer has selected for source list, so this kind of files will appear on it. Therefore the user only can select this kind of files and the player will never play one file with another extension.
The wave files are arranged in alphabetic order and if the list can not take all the files, appears one vertical scroll bar on the right of the list. The user can access to all the songs only scrolling the bar.

The songs the user wants to include on the play list are added from this list. Then I can say that this is the source list for the player.

Once the list shows all the wave files from the selected folder the user has to decide which songs he wants to add to his target list (play list). The user can do that action doing click on the music files.

When the user clicks on the files name, the background changes the color to blue, it means this file is already selected. Doing click on one selected file again, the file goes back to the “no selected” state. But if you make double click on one “.wav” file, this file will be included on the play list.

Adding more than one file, do click on the first one, then press Shift and the last one at the same time. All the songs form the first until the last are selected.

Clicking on the “Select All” button (bellow the source list) the program selects every “.wav” file from the active folder.

Fig. 4.5.
**PLAYER PART**

The player is the most important part of the project. The user finds in this section every function that the player needs to reproduce music, to select the work mode, to set music types, etc... All these functions will be explained later one by one.

The next sub-parts compose the player section:

- Player title.
- Timer Display.
- Play Back buttons.
- Play Mode.
- Play List (Target List).
- File Playing Label.
- Setting buttons (Link buttons).
- Menu.

---

Manuel Delgado
TEI of Kavala
PLAYER TITTLE

As you can see in the image (fig), two parts compose the title. One doesn’t ever change the name of our player, “Music WAV-MP3 Player”. The second one appears when the user has selected one of the music types, and disappears when the user modify again the play list (deleting or adding elements).

![Music WAV-MP3 Player (slowMusic)](Fig. 4.6. Title of the player)

Once, when the user clicks on one type the title shows its name, even if there isn’t any song file. The program will warn by a message.

TIMER DISPLAY

The user finds a very good help to know how long the length is and the play position of the music file that is playing. As I say, this part is only a little help, it isn’t very important for the general performance of the player, but its skin looks much better.

Three elements compose the timer play. Two of them control the position of the playing file, the other one control only the duration of the file. Thus the user knows how long the file is going to be playing.

![Duration](Fig. 4.7. Display)

On the picture you can see the position control timer on the right of the display at minutes and seconds format. The work of this “timer label” can be changed clicking once on it and automatically the mode will change.

You can find the playtime of the sound file on the left of the display. Its value only changes when another song is going to start or when the user stops the reproduction. Then it displays the length of the next music file for playing.
PLAYBACK BUTTONS

Playback buttons control whole the player. They are the most important commands. Next you can see the different elements that compose this part; the figure shows them in this order:

- PLAY
- PAUSE
- STOP
- NEXT
- CONTINUOUS PLAY
- VOLUME CONTROL

Fig. 4.8. Play-back buttons

Now I will explain one by one all of them. I think this is very necessary, because if the user wants to know how the player works, he has to know perfectly how these buttons work. The performance of them is more or less the same like the typical player you can find in Windows environment, but there are some important differences.

The program control when the buttons have to be enabled, thus the user follows the steps that the programmer wants, or better said, the user doesn’t follow any wrong way.

PLAY BUTTON

What can I say about this button that anybody knows? I suppose you are thinking the same, however I must give some information to you that can be very useful.

The button is only active or enabled when the system is stopped (there is not any song playing), then the player is waiting for a demand of the controller. The rest of the time the play button is not enabled.
Sometimes the program activates this button by itself, and then the user doesn’t have to click. For example, when one music type is selected, the play button immediately is activated, so the file starts reproducing.

Before clicking the button make sure that the songs of the list and the play mode are setting correctly.

PAUSE BUTTON

This button is the less important button in the “playback” control. Otherwise the user uses it very often, because sometimes he wants to stop the file without having to restart the cycle of the play list.

Doing click on the “Pause” button makes the timer display label flicker, showing the user that the player is waiting for doing click again on the button. The program disables automatically “play”, “stop” and “next” buttons.

When the pause is activated the user can’t work on the play list, because he can delete the file that is playing at this moment and the program can’t analyze this action.

STOP BUTTON

This button makes the program stop the reproduction. After that the program points the first file that is going to be played on the next cycle, when the user does click on the play button. It will depend on the play mode that is selected, of course.

Sometimes the program activates this button by the source code, because some operations need it, like for example when the next button is clicked the program stops before the song and then plays the next one.

Doing click on this button the user disables “play”, “pause” and “next” buttons.

NEXT BUTTON

Next button is created to advance song by song on the play list. Thus, if the user doesn’t like or doesn’t want to listen to one song, he can click on the button. The program sends the pointer to the next file for playing.

This button is always enabled during the reproduction of the files.

The user can check song by song the play list, and after that decides which files he wants to listen to.
CONTINUOUS PLAY

Continuous play button has two different states. The “on” state of the button makes possible to reproduce cyclically the play list all the times the user wishes (until the stop button is clicked). The “off” state permits to reproduce only once the play list, when it is over the program stops the reproduction.

“Random Play” mode activates automatically this option for all its work.

VOLUME CONTROL

Volume Control button opens the Windows volume control program. There, the user modifies the volume on the wave balance bar. Below you can see the Windows98 volume control system.

![Windows Volume control](image-url)

Fig. 4.9. Windows Volume control
Doing click on this button the Window95-98 volume control is loaded. Make sure before clicking on it that there is no program already opened, because the user could open several windows and that takes space in the PC memory.

**PLAY MODE**

Play Mode arranges the play list for the program to reproduce the files. These are the play modes you can find on the player.

- Play only Selected
- Play All
- Random Play

The options are on the left bottom of the player like the figure shows.

![Fig. 4.10. Play Mode](image)

Clicking on the option can activate only one of these options. There is always only one option active.

When the user changes the mode during the reproduction, the next song for playing is the first one placed on the list in this mode.

Sometimes the program dominates the selected mode depending on the actions the user has done.

**Play Only Selected**

With this option active the player reproduces the files that are selected on the play list. When the user has selected this option and there are not any selected files, clicking on the play button it will appear a message telling him that it is not possible to play any file with this mode active. (Fig 4.11)
Clicking "OK" on the message window the program undoes the action. This option is disabled after clicking on "Select None" button.

Play All

"Play All" is the default mode, so when the user starts the player, this play mode is already selected. This option is the normal work mode. The player reproduces in order all the songs from the play list. It doesn’t care if there is some file selected, it plays from the first song of the list to the last one; naturally, if the user stops the reproduction, the pointer will go back to the beginning again.

Clicking on some of the music types on the menu the program calls immediately to this mode. Afterwards the user can select another mode.

Random Play

"Random All" option creates a random system to select the file that is going to be played at every moment. Randomize system works every two songs, therefore the user can be including more songs on the play list not altering the normal process of the player.

The program is playing randomly all the songs without repeating any song twice until one cycle has been finished, that is the main condition for "Random Play" mode. Thus the user doesn’t always listen to the music in the same order. When another mode is working (the music is playing), clicking on this option the program initializes the pointer for the play list on the new mode and starts to play automatically. The mode doesn’t allow deleting any file from the play list during the reproduction.
Continuous play is always active in this mode. So the program never stops by itself, the user has to stop it.

**PLAY LIST (Target List)**

Play list also called “target list” is placed in the middle of the player, thus the user can access to it easily. It links the browse with the play system.

The list shows the files with the extension “.WAV”, but you have to remember that they are not wave files but MP3 files. However, wave files can also be played.

The user owns some folders that contain some songs. These songs are taken from the source list and added on target list (fig.). As I have already explained in some pages before.

![Play List](image)

Fig. 4.12. Play list

The list can take an undefined number of songs on it. A scroll bar is displayed on the right side of the play list when the songs that the user has added don’t fit in it. Thus the user can access to all the songs.

The user can make some selections on it. What for? Two main reasons:

- To reproduce the wav-mp3 files in “Play Only Selected” mode.
- To delete the files from the play list (the files disappear, but they are not deleted from the drive folders).
The selection can only be done one by one, because the only way that I have found to be able to select two songs those are not together on the list.

The list is enabled all the time except when the user clicks on the pause button. It means that the user can be setting the list at every moment. This is one of the best advantages of the player.

The list contains the songs that the user can use to make one group at “Set Music Type” option in the “Edition” menu.

**FILE PLAYING LABEL**

“File Playing” label shows the song file that is playing or the song that the system is going to play when the user clicks on “play”.

![File Playing](Fig.4.13. File playing)

The label has got three different states:

- Flickering: It shows the file that is being played.
- Normal or not flickering: When the player isn’t playing any song file and the play list is not empty.
- Not visible: When the system is stopped and the play list doesn’t have any music file in it.

**SETTING / ACTION BUTTONS**

Setting buttons are placed on the right of the “player part” or in the middle if the “browse part” is displayed.

These buttons are designed to link the source list from the “browse part” and the “target list” (play list) from the “player part”.

They are not always enabled for the user; sometimes the program disables all of them for not having to do any strange action that the system doesn’t know in its program mode.

The buttons hide their names when they are not enabled.

Let’s see one by one the function of each button: