D-Box Manual

- everything you ever wanted to know about the box but were afraid to ask -

Victor Zappi

victor.zappi@qmul.ac.uk

Andrew McPherson

a.mcpherson@qmul.ac.uk

Queen Mary University of London

centre for digital music

Hackable Design

The D-Box has been designed to support **modding** and **hacking**. If you remove the 2 side plates [the ones between the speaker and the sensor plate], its circuitry is completely **exposed.** Feel free to mess around with it, taking into account that:

- you can experiment a lot by hacking the *Matrix* [all the components on the breadboard, Fig. 1]. It's not very likely that the instrument will break or "brick" by changing this circuitry;
- changing the connections to the *BeagleBone Black* [the black board on the bottom of the box, Fig. 2] is a bit more dangerous for the life of the instrument. Anyway, <u>we don't forbid hacking here</u>. Have fun;

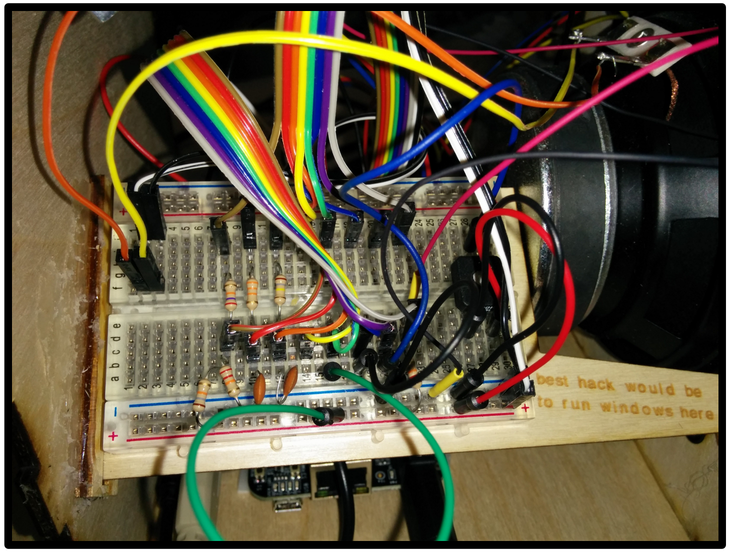


Fig. 1 - The Matrix, like in the movie

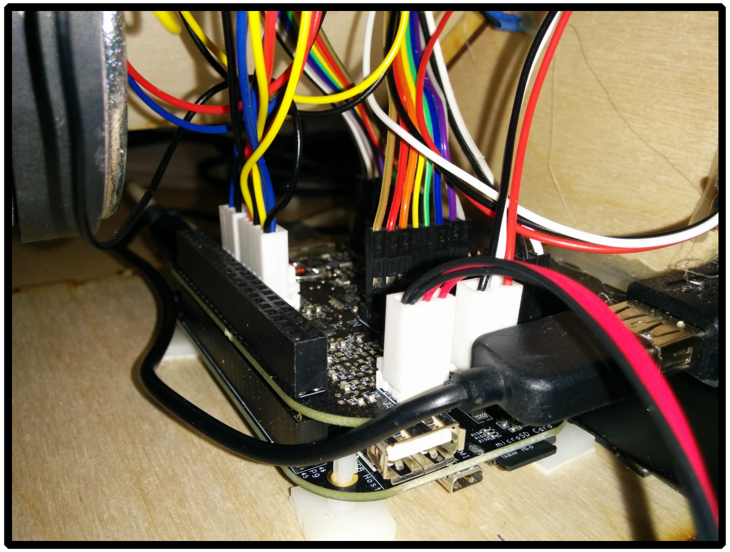


Fig. 2 - The BeagleBone Black

- in particular, attaching components to the 2 black *headers* on its sides [Fig. 3] could be lethal for the board. These inlets are not even programmed. For these reasons we suggest leaving this part of the board in peace;

- 2 *piezo discs* [contact microphones] are hooked up to the BeagleBone via the wires and the connector shown in Fig. 4. Feel free to disconnect them if you do not want to use them;

- if you want to use the headphone jack as the only audio output, disconnect from the Matrix the 2 wires that are currently connected to the speaker [Fig. 5].

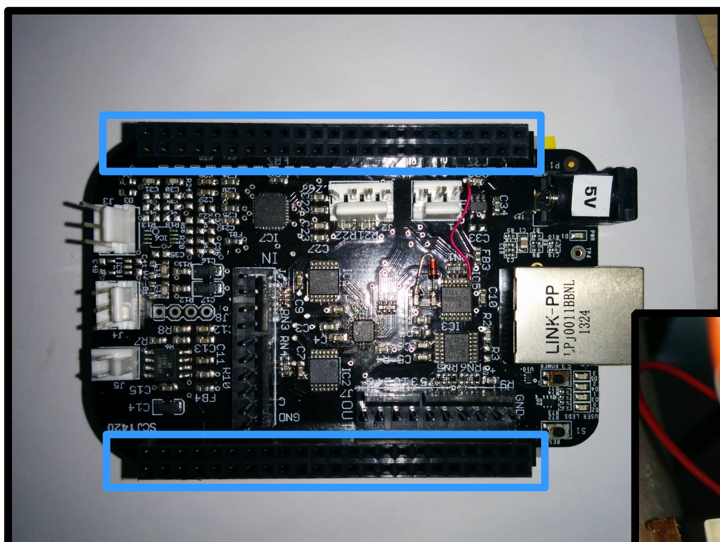


Fig. 3 - The 2 dangerous headers

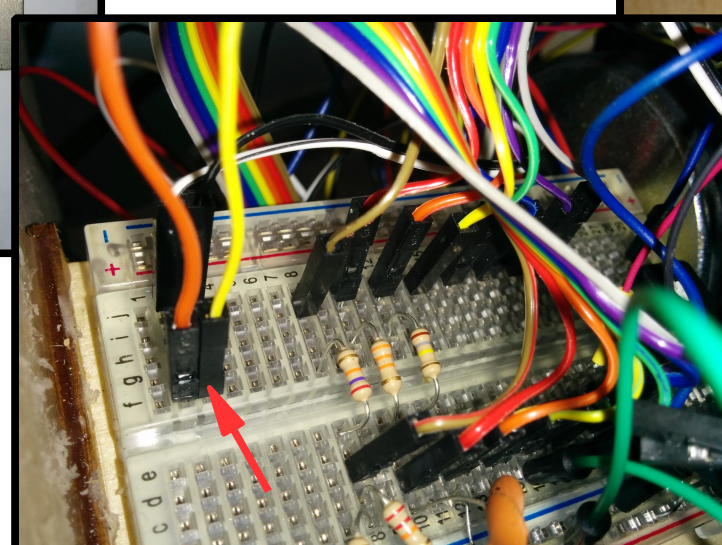


Fig. 5 - Speaker cables

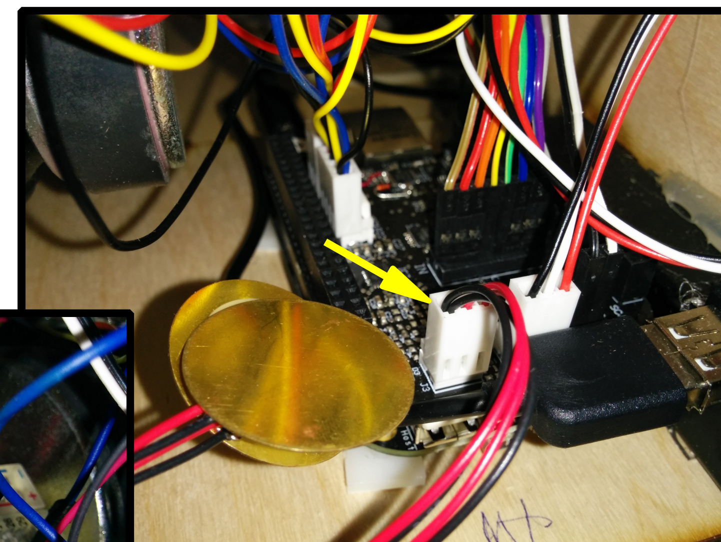
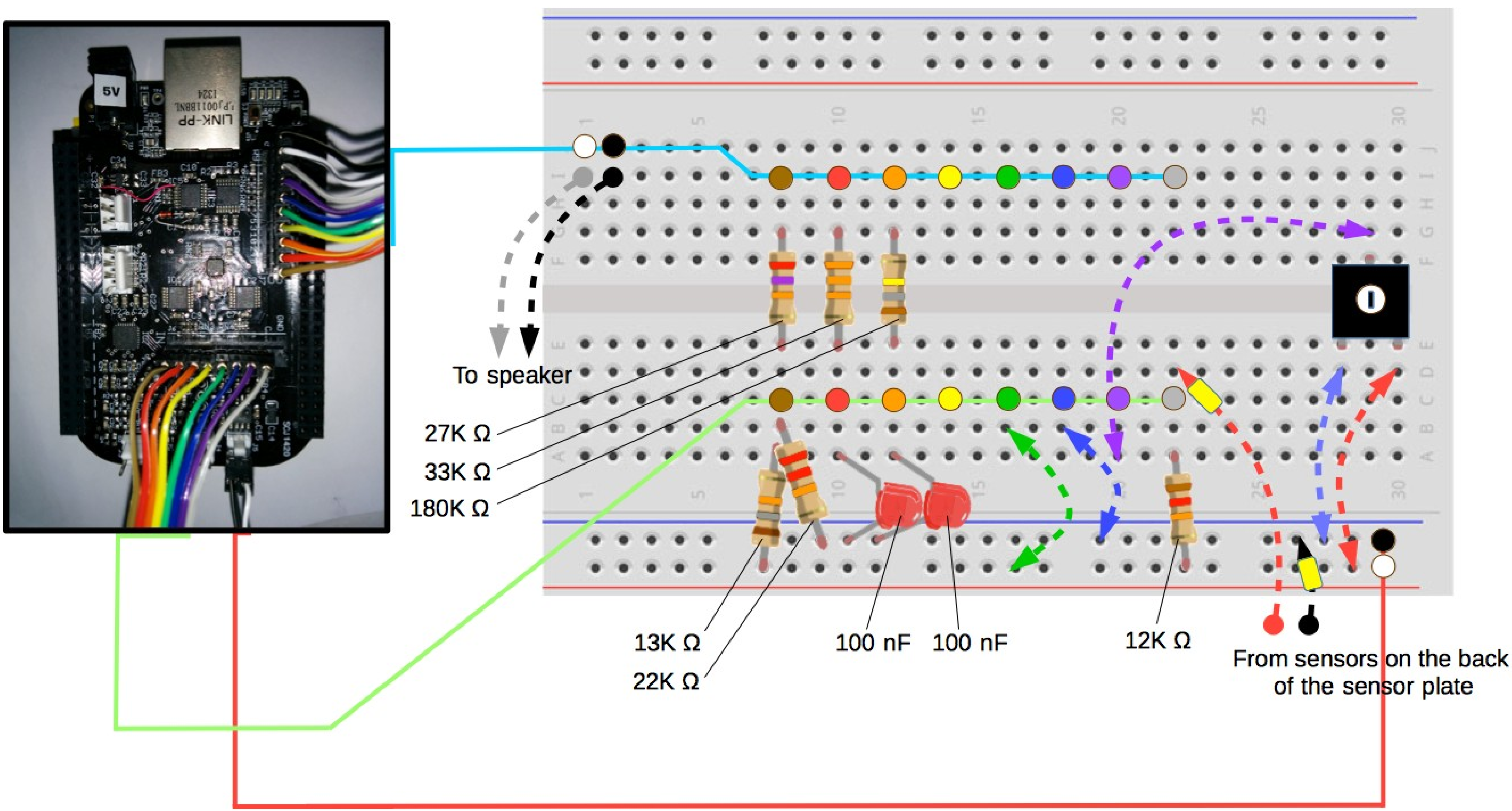


Fig. 4 - Piezo discs wiring

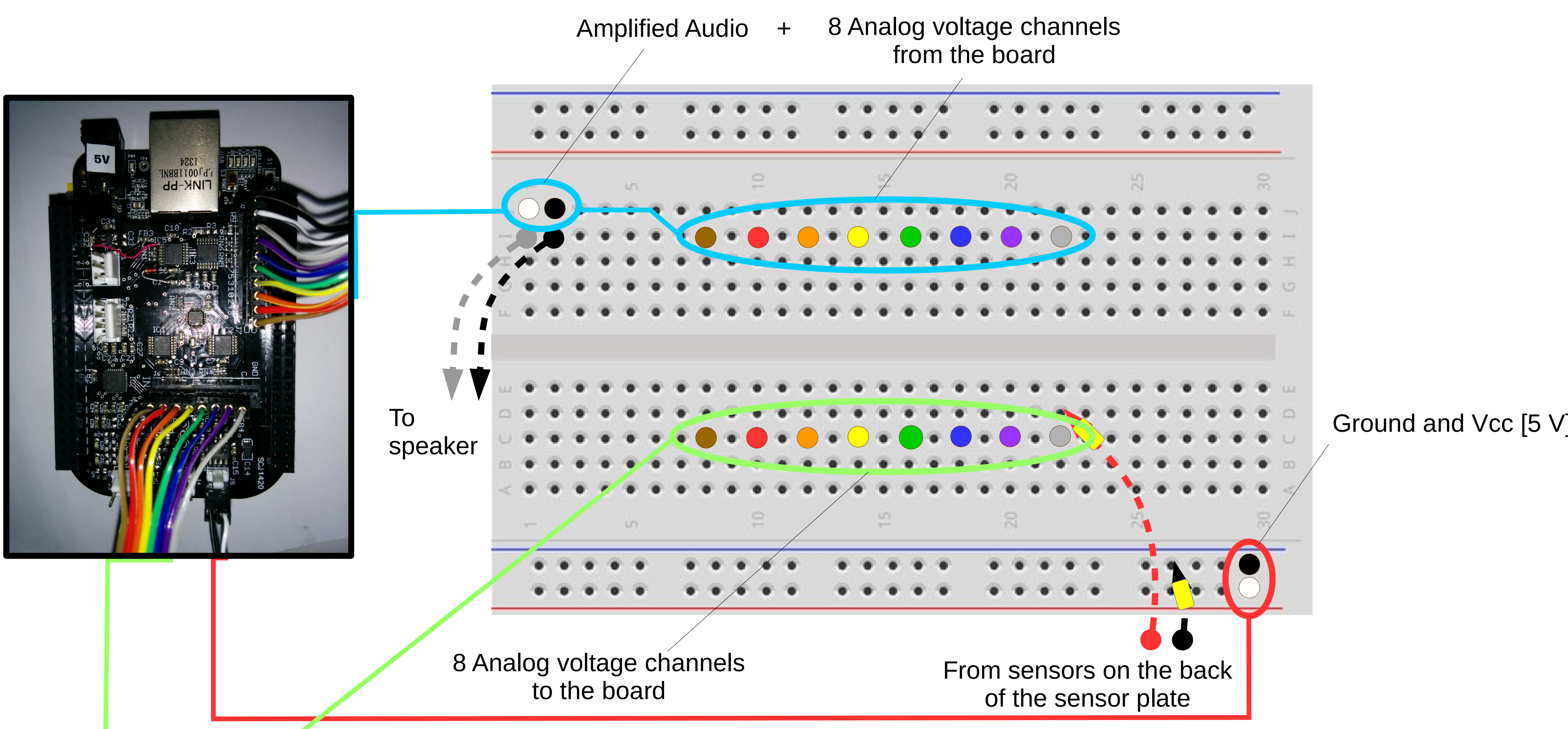
Original Configuration

This scheme could be handy if you'd like to revert to the original configuration at any point.



How the Matrix Works

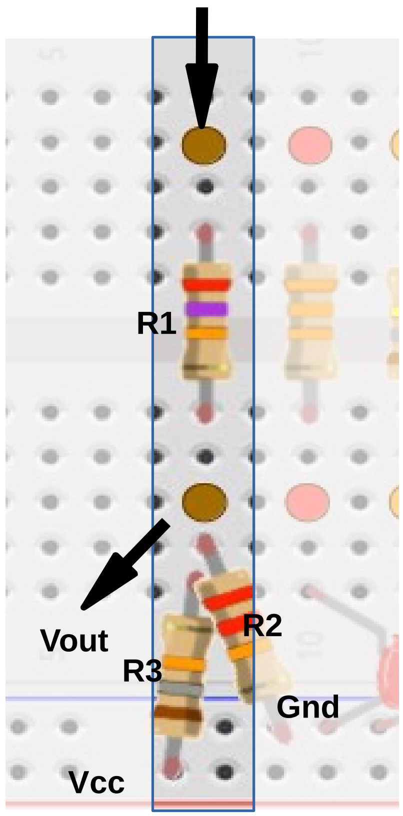
The Matrix is a set of circuits which send and receive voltages to and from the BeagleBone Black. In the following scheme you can find some details about the wires that connect the board and the Matrix:



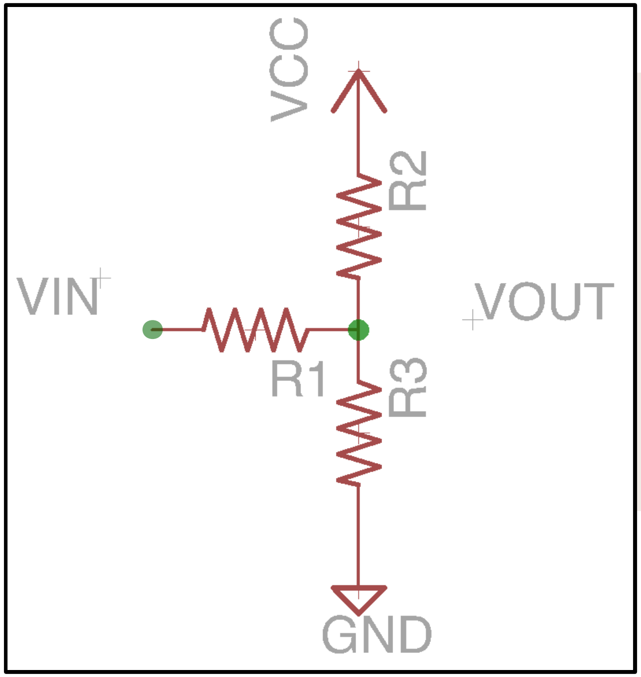
Brown Circuit: Pitch

Through this circuit the position of the finger touching the first touch sensor controls the **pitch** of the play back, in a **Control Voltage** style

The circuit receives in input the **position of the last finger** touching the sensor [0-5V]



In the original configuration, the circuit consists of a resistor ladder:



This limits the voltage range according to this equation:

$$Vout = \frac{Vcc(R_1 \cdot R_2) + Vin(R_1 \cdot R_2)}{R_1 \cdot R_2 + R_1 \cdot R_3 + R_2 \cdot R_3}$$

offset

delta

The circuit output voltage is read from the board and converted into **pitch**

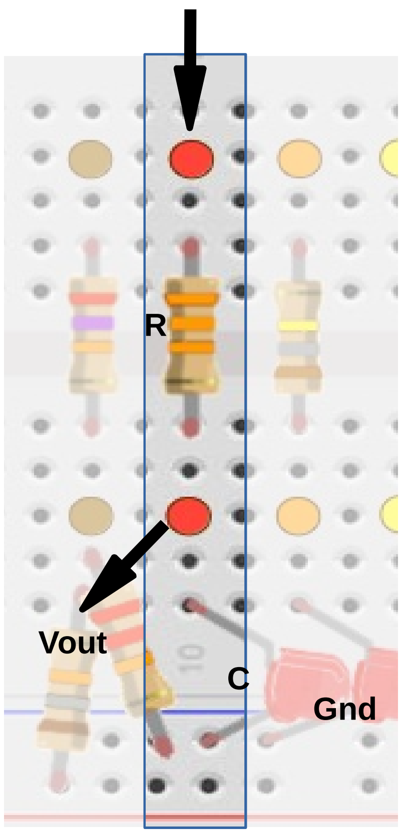
[0V is 2 octaves lower than the original sample pitch, 5V is 2 octaves higher]

Red Circuit: Clock

This circuit determines the **system clock speed**

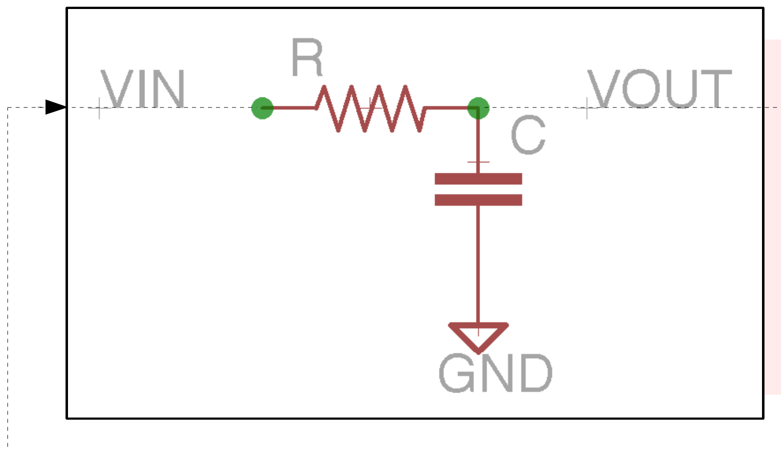
The circuit receives in input the current clock of the system, as a square wave

Vin



The output is a waveform with increased or decreased rate, fed back into the board

The original configuration is an RC charging circuit:



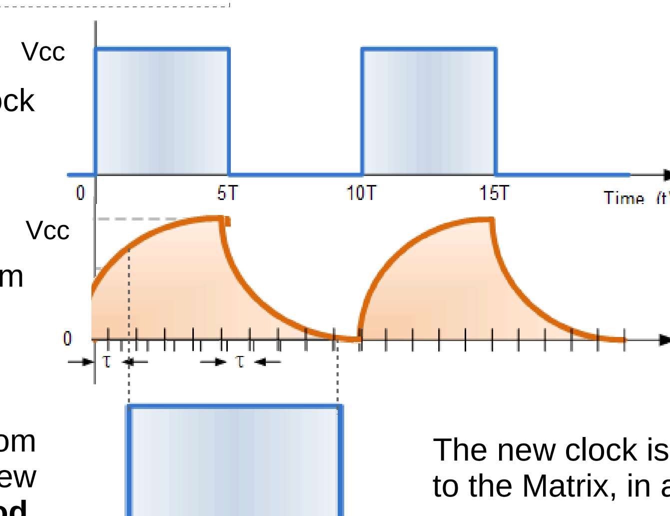
The circuit delays the instantaneous changes between rising and falling edge of the square wave, according to the time constant:

$$\tau = R \cdot C$$

The bigger the values, the higher the delay [**lower clock rate**]

Vin, current clock

Vout, delayed waveform

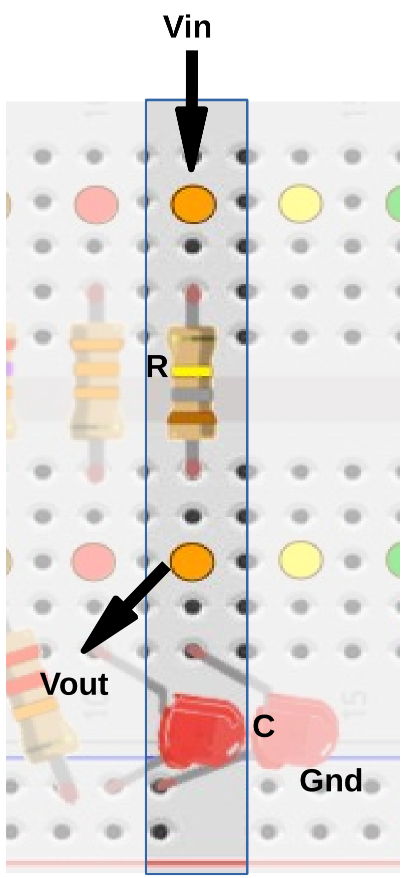


The delayed waveform is read from the board which calculates a new **clock period**

The new clock is then sent back to the Matrix, in a feedback loop

Orange Circuit: Wavetable

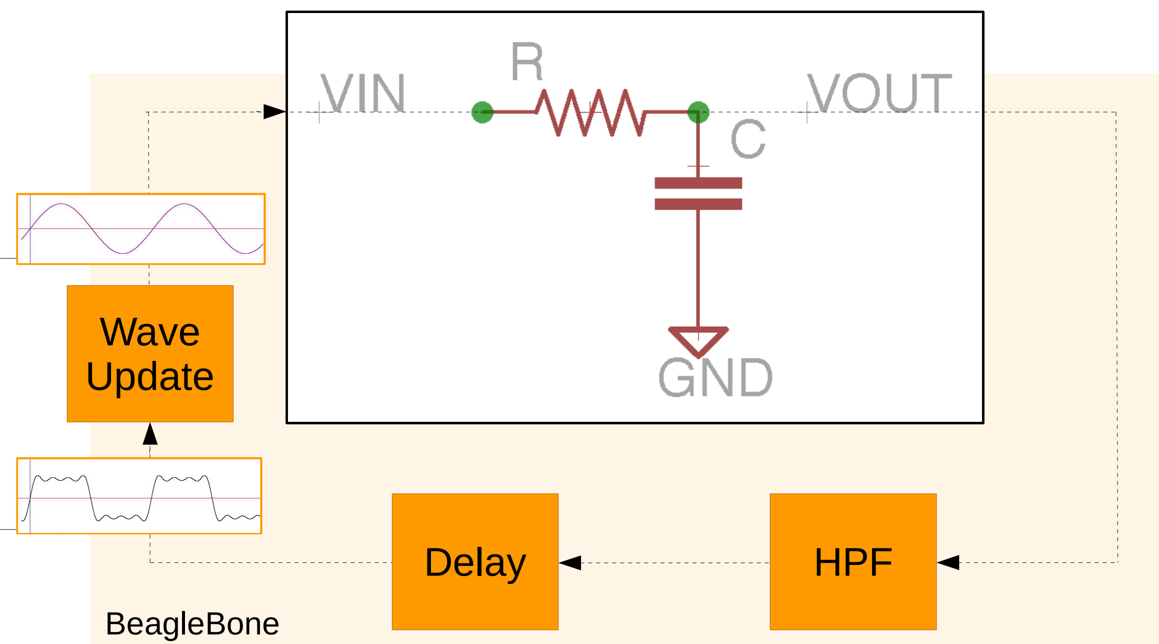
On the D-Box, audio files are loaded split into **partials**. These partials are played back by a bank of **oscillators**.
Each oscillator by default produces a sine, whose shape is saved into a **wavetable**.
The orange circuit allows to **change the waveform inside of the wavetable**, <u>changing the timbre</u> of the oscillator bank.



The original configuration is an RC charging circuit, connected in a <u>feedback loop</u> to a software high pass filter and a software delay stage running on the board:

Under certain conditions the circuit oscillates, **replacing the sine** in the wavetable with the resulting waveform

For example, with this



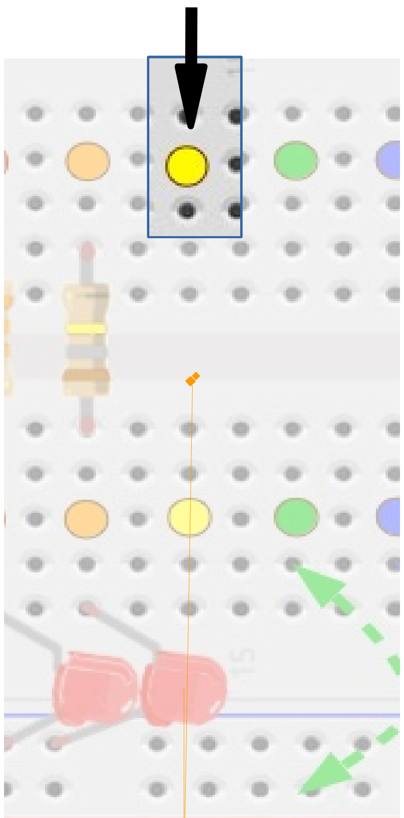
Among other possibilities, small capacitor values [or the absence of the capacitor], result into a more **squared** waveform and into a <u>brighter timbre</u>
while
big capacitor values result into a waveform more similar to a **sine**

Yellow Circuit: Finger Steps

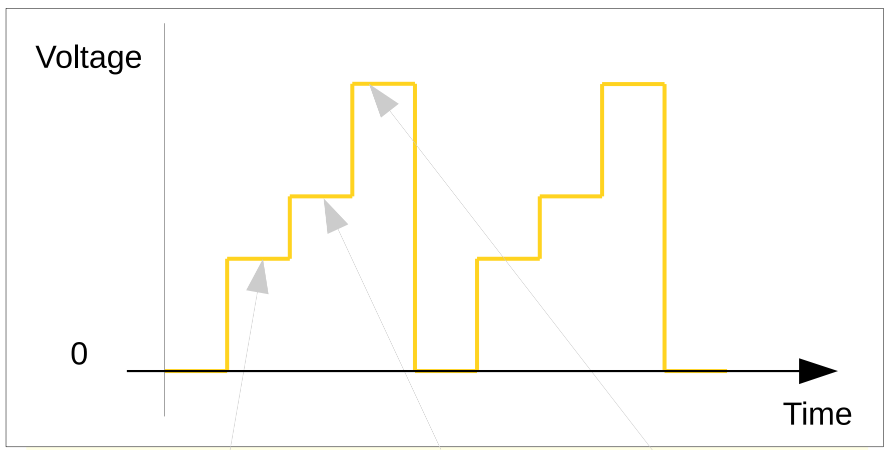
This circuit feeds the Matrix with an input which varies according to the position of **all the fingers** touching the second touch sensor [up to 5]. It can be used as **control signal**.

The circuit receives in input a periodic voltage which depends on the **position of the fingers**

Vin



Each period is a **multi-step square wave**.
Here is an example with 3 fingers:



First finger *Second finger* *Third finger*

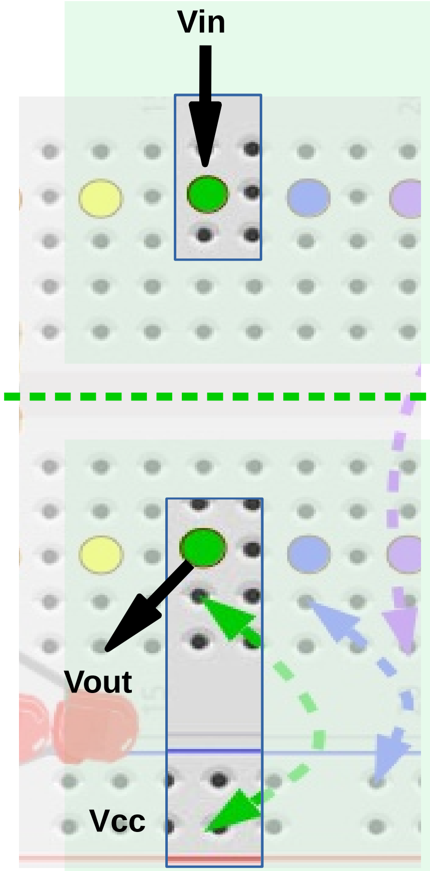
Fingers are **ordered** according their position on the sensor.
The voltage value linearly depends on position.
If the finger is on one extreme end the value is 0V, while on the other extreme end is 5V

As you can see by default the circuit is **floating**, that means Vin is received by the Matrix but not connected to anything else.

Feel free to **route Vin** wherever you want.

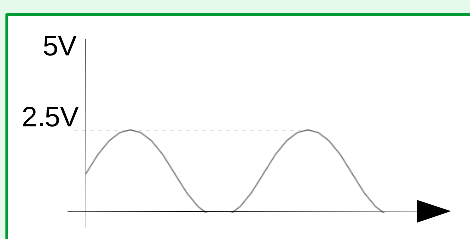
Green Circuit: Finger and Loop

In the original configuration the input and the output of this circuit are **not connected to one another**.
The input feeds the Matrix with the **position of the last finger** touching the second touch sensor.
The output determines the **loop points** on the played files.



The **position of the last finger** touching the second sensor is here received as a voltage level.
The range is 0V to 5V.
This voltage can be routed in the Matrix as **control signal**.

The D-box plays a **portion** of the loaded files, delimited by 2 loops points.
The waveform sent from the Matrix to the board is analysed and the local **minimum and maximum** voltage values are used to move the loop points.
Voltage is linearly mapped on the current file length, with 0V meaning the beginning of the file and 5V its end.

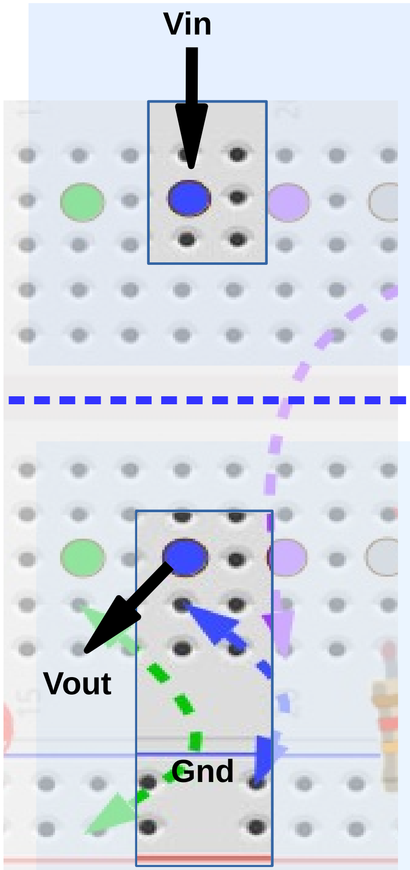


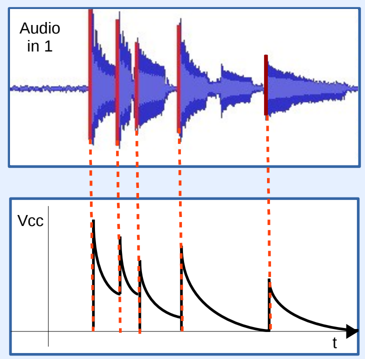
In this example, the local minimum and maximum are ***0V*** *and* ***2.5V****, so the file will be played from the* ***beginning*** *to the very* ***middle*** *of its length*

In the original configuration the output values is **constantly to Vcc** [5V]. This is a special case to force the **whole playback** of the file.

Blue Circuit: Onsets and Motion

In the original configuration the input and the output of this circuit are **not connected to one another**.
The input feeds the Matrix with a **voltage envelope** whenever an **onset** is detected on <u>audio input 1</u>.
The output controls the **frequency motion** of each partial, with respect to its **average** value.





Whenever an onset above a <u>certain threshold</u> is detected on audio input 1 [piezo disc], an **exponential envelope** is sent to the Matrix.

The peak of the envelope is **proportional** to the peak of the onset.

This voltage can be routed in the Matrix as **control signal**.

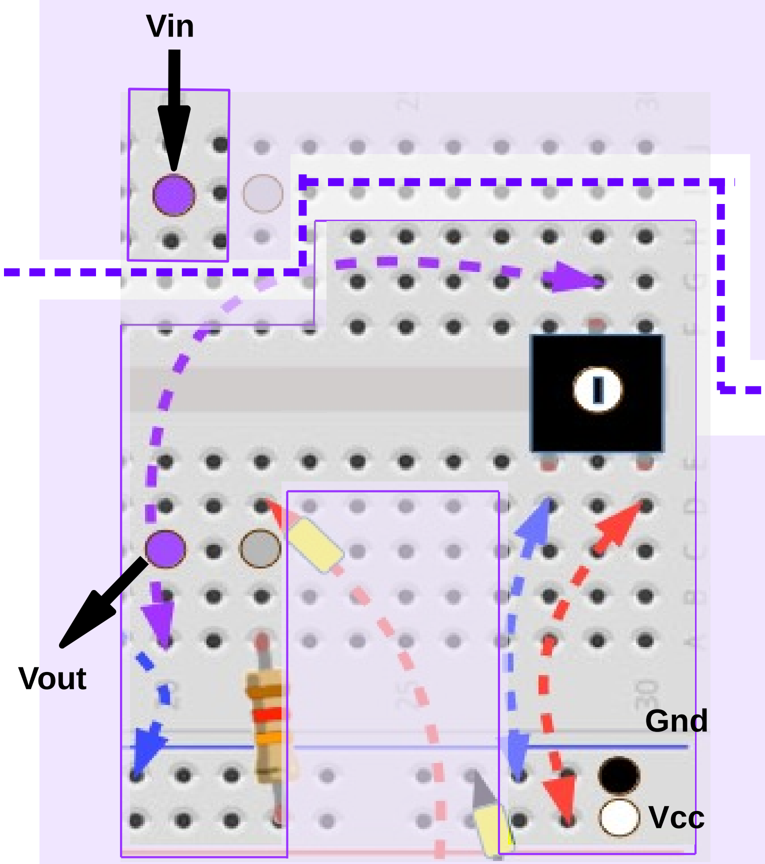
Each partial **changes its frequency and amplitude over time**, according to the analysis file loaded on the instrument.
The D-Box is capable of reducing the frequency motion of each partial, in a range centred around its average frequency value, which is calculated over the whole file.

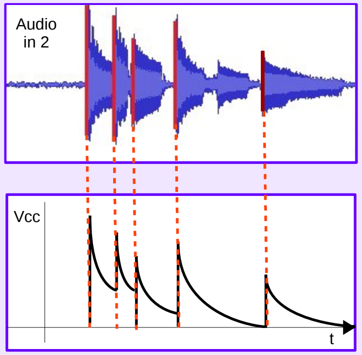
The blue output is read by the board which uses the voltage to **limit the motion range**. By default 0V are sent, meaning full motion range.

Increasing the voltage, the range diminishes and a **dissonant/robotized effect** is introduced.

Purple Circuit: Onsets and File

In the original configuration the input and the output of this circuit are **not connected to one another**.
The input feeds the Matrix with a **voltage envelope** whenever an **onset** is detected on audio input 2.
The output **selects the file** to play back.





An **exponential envelope** is sent to the Matrix when an onset is detected on audio input 2 [piezo disc].

The peak of the envelope is **proportional** to the peak of the onset.

This voltage can be routed in the Matrix as **control signal**.

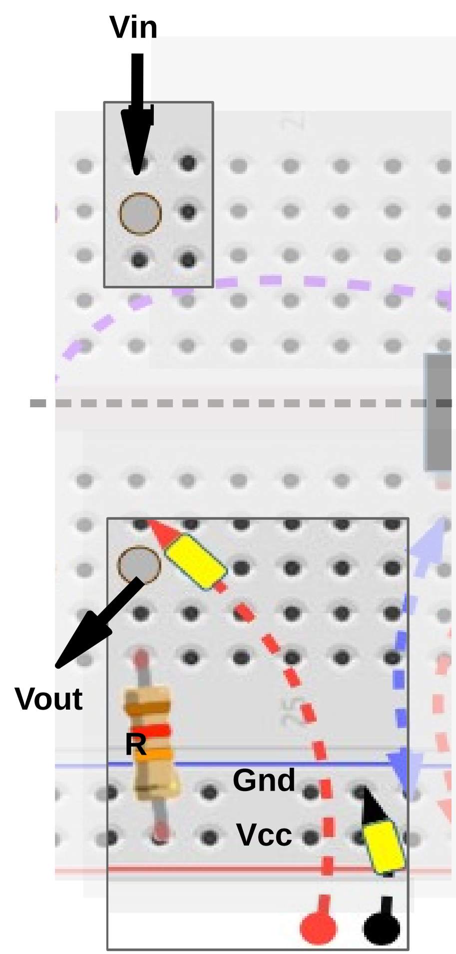
The D-Box can load multiple files. This circuit is used to **select** which **file** to play back from the YES_YES partition of the micro SD card.

In the default configuration, the Matrix output voltage can vary between 0V and 5V, simply turning the **turning the preset** [the small black potentiometer].

The board automatically divides this range into **as many sectors as the number of loaded files**.

Grey Circuit : Notes and Pressure

In the original configuration the input and the output of this circuit are **not connected to one another**.
The input feeds the Matrix with the position of the finger touching the pitch sensor, discretizing it into **chromatic notes**.
The output binds the **pressure** sensor underneath the pitch sensor to the **velocity** of each note.

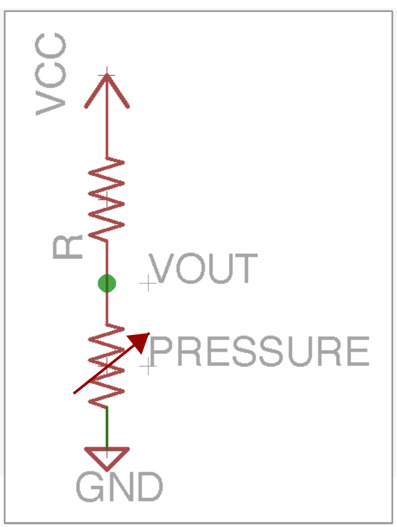


Pressure sensor resistance value

Pressure sensor gnd reference

When the voltage here received as input is routed to **replace Vin in the brown circuit**, the circuit and the board act as a feedback loop system, which adjusts the pitch to the **closest chromatic note**. In this scenario, the pitch touch sensor is actually discretized into different **areas**, each related to a specific note.

Again, acting on the resistor ladder in the brown circuit, the pitch range – in this case the number of discreet notes - can be modified.



This circuit is a **voltage divider**, where one of the two resistors varies its value according to the amount of **pressure** exerted on the pitch touch sensor.

When the sensor is fully pressed, its resistance **tends to 0Ω,** pushing Vout towards ground [0V]. Lighter touches determine higher resistance values and higher voltages on Vout.

Vout is read by the board and converted into **note velocity** [0V max velocity, 5V min velocity].

File Loading

On the D-Box you can load and play your own files. Files are loaded from the micro SD card hooked up on the BeagleBone [Fig. 6].

When you turn on the instrument, the blue LED on the side of the box will blink fast until the last file has been loaded and is ready to be played.

You can have a look at the default set of files the instrument comes with. Remove the SD card from the board [push in the card to release it] and stick it into your computer using the SD card adaptor we gave you. If you do not have a built in card reader on your computer, you may use a USB one.

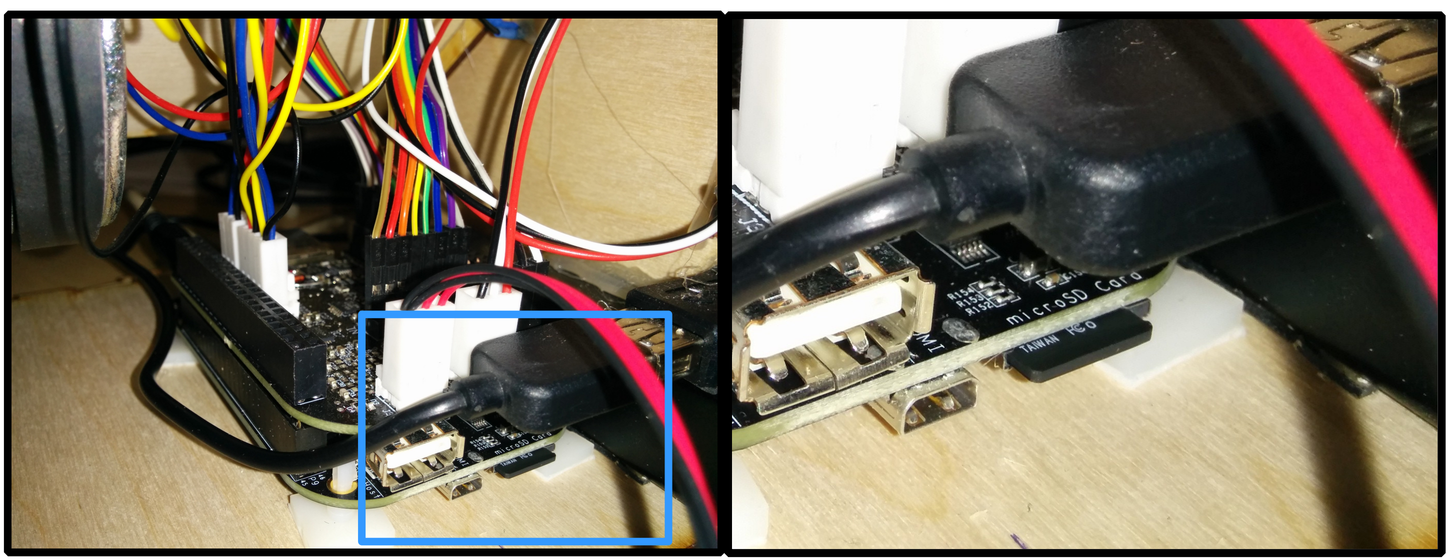


Fig. 6 - BeagleBone SD card slot

Your OS will mount the SD as two different partitions, one called “YES_YES”, the other “NO_NO”. Fight back your instinct to double click on NO_NO and simply open YES_YES.

As you will see, this partition contains 7 files with “.dbx” extension [Fig. 7].

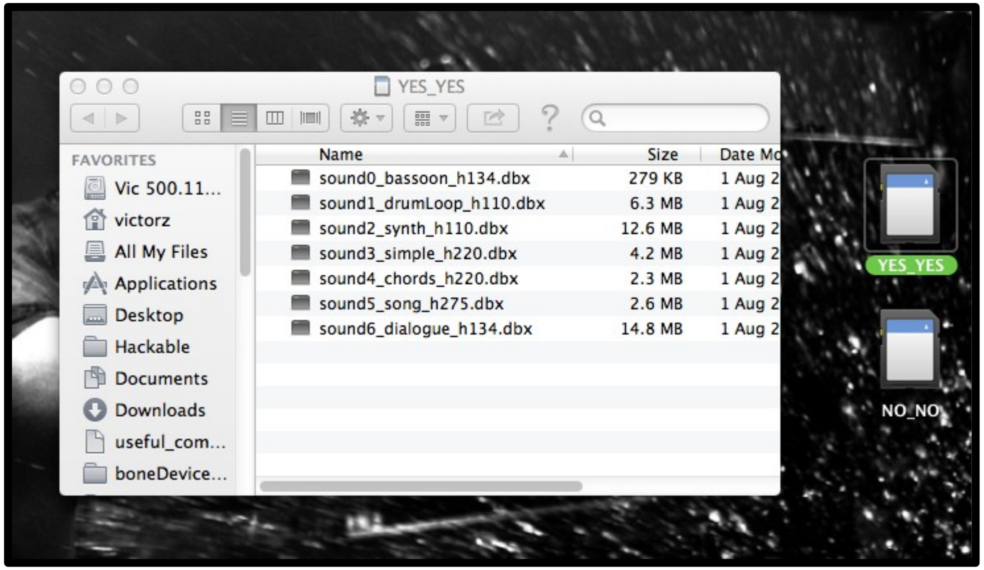


Fig. 7 - The YES_YES partition and the default files

These are the actual files that are played back on the instrument. They are not standard audio files, rather they are analysis files of standard audio files, decomposed in single partials. The D-Box can play back and modify these partials in real time.

If you want to replace the standard files with custom samples, you can use a free multi platform software, called SPEAR, to convert your audio in partial analysis files. Read ahead for more details.

SPEAR

SPEAR is a powerful application for audio analysis, editing and synthesis, developed by Michael Klingbeil.

With SPEAR you can export audio files into a **format supported** by the synthesis engine running on the D-Box [partial analysis file]. First, you have to **analyse** the file. This process decomposes the whole audio track into separate partials. Then, you can **export** it as a new D-Box compatible file.

For more information about the software, please refer to the official page:
http://www.klingbeil.com/spear/

It runs on Mac and Windows and you can download it here:
http://www.klingbeil.com/spear/downloads/

You are free to use all the functionalities included in SPEAR, however this guide will cover only the audio file analysis/export process, which is fundamental if you want to use **custom samples** with your D-Box.

Launch SPEAR, then go to “File > Open...” and choose the audio file you want to analyse and export. Only certain types of files can be opened in SPEAR, the list can be found in the “Open” window dialogue. Luckily, “.aiff” and “.wav” are supported (;

Once you have chosen your file [please use ony 44100 Hz files], a new window will appear, as shown in Fig.8.

1 Please leave this unchecked

2 Tweak this value to change the accuracy of the analysis

The lower the value, the more accurate the analysis

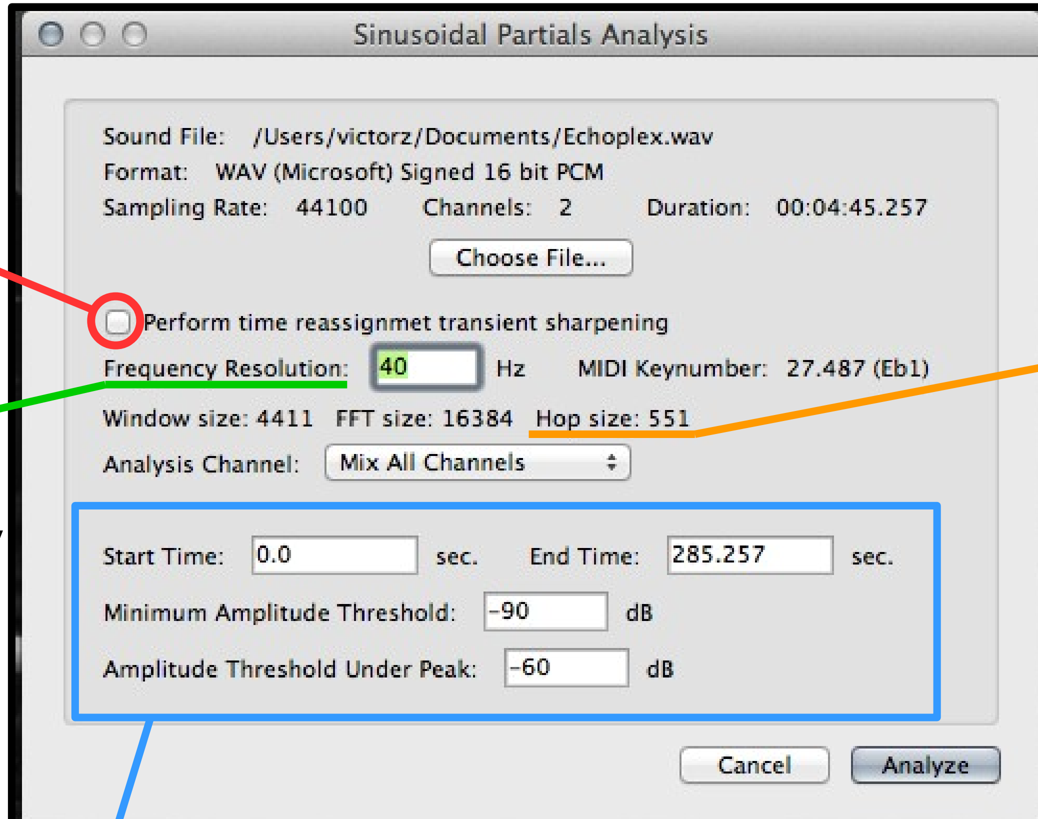


Fig. 8 - SPEAR open file dialogue

3 This value changes according to the Frequency Resolution

Please write it down, <u>we will need it later</u>

4 You can explore these parameters to change the result of the analysis

Further info on their effect can be found here: http://www.klingbeil.com/spear/analysis.html

Once you pressed “Analyze”, a new window will appear [Fig. 9], allowing you to listen to the result of the analysis by pressing the space bar. The file will **sound the same** on your D-Box.

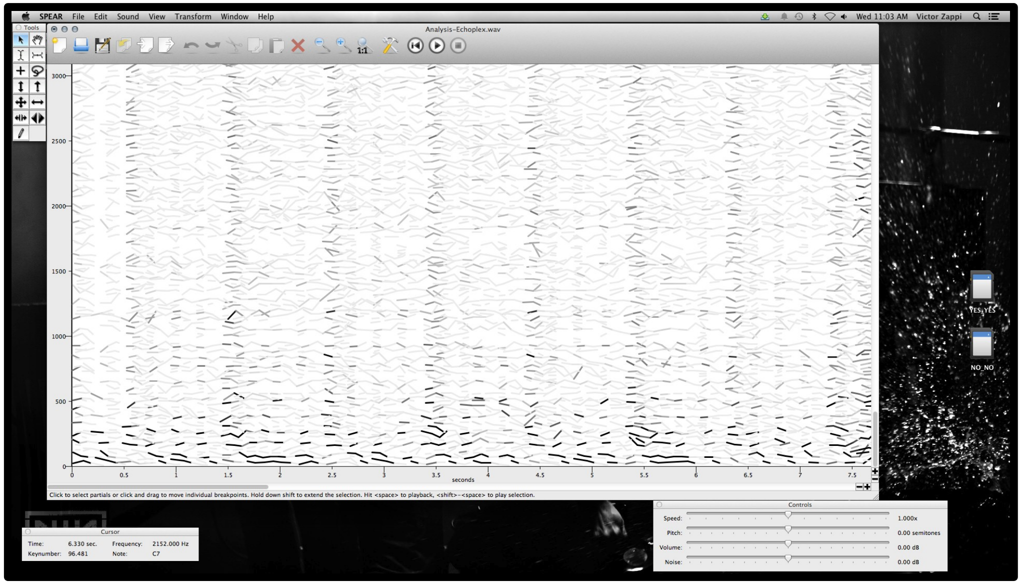


Fig. 9 - Partial analysis window

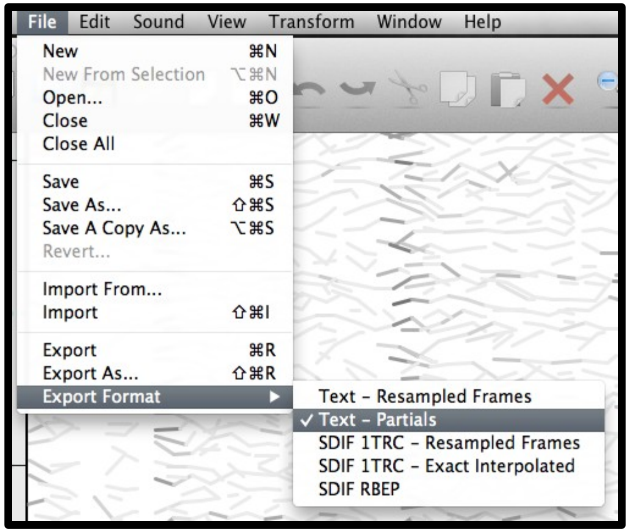


Fig. 10 - Export settings

If you are not satisfied with the result, simply reload the file and tweak the analysis parameters. Don't forget to write down the **Hop Size** you are using.

When the file sounds ok, go to “File > Export Format” and tick “Text – Partials”, as show in Fig. 10

You have to change this setting <u>every time you restart</u> SPEAR, since this is not the default configuration.

Finally, click on “Export As” and choose the name that you prefer

BUT

in the file name, we have to include the <u>Hop Size used in the analysis</u> [see Fig 8]. Simply add ”_h”+ Hop Size before the extension, as shown in the example:

NAME_hHOPSIZE.txt

e.g., the file Echoplex.wav analyzed with Hop Size 551 could be named:

Echoplex_h551.txt

Now you can use this file on your D-Box:, go to the YES_YES partition of the SD card, copy your new analysis file there, put back the SD card in the instrument et voilà!

If you carefully followed the instructions, the instrument will let you play [and hack] your sample. Files are loaded in **alphabetical order** - subfolders will be ignored.

Since you are smart [otherwise you wouldn't be part of this study], you have probably noticed that analysis files exported from SPEAR have a different extension from the ones you initially found on the SD, namely “.txt” vs “.dbx”. <u>They both work</u>! Read ahead for an explanation.

D-Box Dumper

A silly name for a handy tool. This simple application, which does not require any installation, **converts** SPEAR analysis “.txt” files into optimized “.dbx” files, which will load remarkably **faster** on your D-Box.

Even if not mandatory, this is particularly convenient when the audio files you want to load on the instrument are big or when you require a very detailed analysis [the more detailed the analysis, the bigger the exported file].

Using the dumper is very simple:

- move the executable we sent you into an empty directory [file is called “d-box_dumper”];
- copy the SPEAR “.txt” files you want to convert into the same directory [be sure you followed the instructions for the analysis/export process];
- double click on the dumper:
- one by one all your files will be read and new “.dbx” files will be *dumped*.

Now you can replace the converted “.txt” files on the SD with their “.dbx” versions, for **a faster loading experience**!

Troubleshooting

What to do if your instrument does not boot? Don't despair! First of all, open the box and check if the blue LEDs on the BeagleBone Black are blinking [Fig. 11a].

- If they are off, the battery might be low. Try to charge it for a while and then reboot.
- If they are off and the battery is ok, try to push the 2 buttons shown in Fig. 11B, first the rightmost one then the leftmost one.
- If it still does not boot, contact me!
- If the LEDs are steady, turn it off and on again. If they are still steady, contact me!

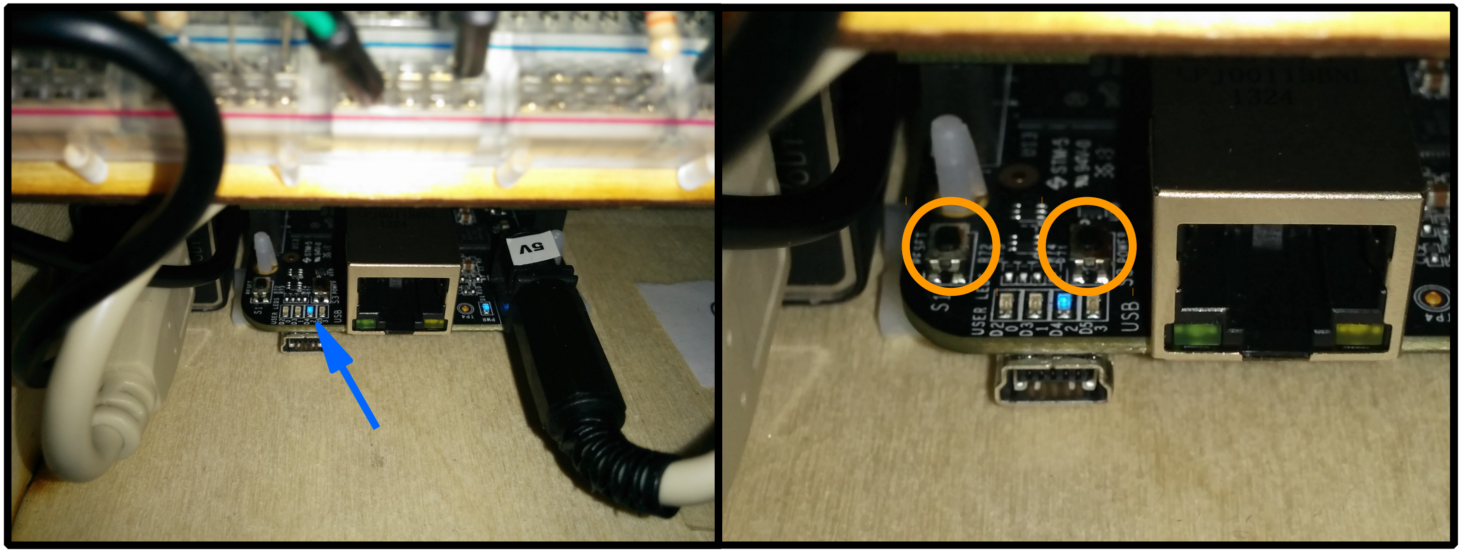


Fig. 11A and b - BeagleBone LEDs and the 2 buttons