PDF explanations/translation

[Partie 1]

Hello,

We will make a series of videos about zip card producted by Interface-Z.

You probably know several types of electronic cards. This one is specific to the use of a software named MAX. I think that it will make the happiness of many designer in digital art; because it is a card which can directly integrate the patches MAX MSP.

The objective of this series of videos will be, first of all, to present the card. First, we will see how it works; then how it works connected to the computer in external mode to activate motors or others, or you can use it in autonomous mode, without computer. Finally, we will see in a clearer way the particular programming that you need to have for the card.

This card can be called "patch interpreter".

With this card, you can do anything: robotics, sculpture, interactive installations ... It depends on what you plug in.

This card is very interesting, because of its small size, the part in the center can be cut. It is a small card, but it has all the necessary elements.

Note that this card uses the Midi protocol.

There is the Midi input and output protocol, this protocol will be central to the use of this card.

There is a switch on the card to activate the mechanism. It is used to indicate whether the card works in internal or external mode, autonomously or with the computer. There is a second switch that allows you to choose the resolution of the elements used (the sensors).

There are several resolutions: 7 bits = 128 steps Midi (0 to 127)

12 bits = 94 steps

We can observe 4 analog inputs, which are in the form of branches. 4 analog inputs to which we can connect sensors. 5 digital inputs for the buttons. 1 actuator output on which we can connect daughter boards.

Thanks to the Midi input and output couple, we can add other elements if necessary. For example, we can add a card in the Midi input in order to have more sensors; while in the Midi output we can add daughter cards for the use of servomotors.

However, the necessary equipment is limited to the card and a Midi interface, and a computer if your card does not work autonomously.

[Partie 2]

This tutorial will help you to discover how the second generation ZIP card works in external mode. We will see its use connected to a computer.

We are going to use a Midi interface and a transistor board which will be powered with 12V.

The card is powered with a DC voltage between 9V and 15V. Here we use a voltage of 9V. The card has a polarity, we must put the plus towards the plus, and the minus towards the minus, following the indications on the card.

Be careful when connecting the Midi (input/output), the Midi out enters the Midi In of the card.

(For the connections see the diagram on the screen at 1min56)

Then we connect the daughter board/transistor. We can now power the card.

We can observe a departure from the computer that comes out of the serial output to the Midi interface, then the Midi cables are used to communicate with the card and the Midi card is used to communicate to the transistor card.

Once the connections are made, we will use 12V LEDs connected to the output of the transistor board. Then, by launching the MAX patch, we will check the good connection between the computer and the card.

The first MAX patch to launch is the direct test ZIP. We have to check that everything is on the Midi output (sending and receiving). This allows us to check the sending of data by the analog and digital sensors.

We can try to connect sensors to the analog input and buttons to the digital input. (4min24).

Once the assembly is done we look at what is received by the computer program. It is then possible to note data on MAX, those captured by the card.

Now that we know that our card receives data, we need to check that we can send data. To do this, activate the Toogle (5min28).

Attention = be sure to know the voltage required for your actuators!

You can now make your assemblies. The most important thing will be to take note of the " control change " programs of MAX. At the output, there are digital boat allocations and the " control change " allows to send data to each output of the daughter board.

Next time, we will see how to use the card without a computer.

[Partie 3]

We will thus pass on the autonomous mode. We have to launch 2 patches on MAX : the standalone test ZIP, directly integrated in the board, and the loading ZIP.

The test patch is presented with 5 objects, a loadbang and abstractions. The abstractions are sub-patches. They have been created specifically for the ZIP, we will develop them in the next tutorials.

Attention MAX has also some rules of operation.

We have here various abstractions such as Ana_In_7bits which is used to retrieve data from analog sensors, or num_in which is used for data from digital sensors. The last two objects PWM_OUT and PWM2_OUT, are the transistor outputs.

This patch is an equivalent of the previous one adapted to the board to be autonomous.

Here is the loading patch (2min27). There is no need to go into depth in this patch, just to check the connections to the Midi interface. On this patch there are buttons for incrementing/decrementing the patch numbers.

Now you can choose your MAX patch. You can drag a patch to a specific number with the buttons. It is possible to jump from patch to patch.

To start, move to position 0 and switch on the patch. You can check the errors or the correct loading of the objects with the console. The patch is well loaded when it respects the rules of the ZIP.

How to check if the patch really works? We will connect an analog sensor (breath sensor in the video) and an actuator. The actuator connects to the input and output number 4, it has here a motor function. Then press the Toogle to activate the patch.

With the breath sensor, if you create air, data enters the sensor.

To see if the card is really autonomous, you have to remove the midi sensor and activate the switch number 2 and restart the card so that it becomes autonomous. The yellow LED must be off. The card and the sensor must govern in the same way as before (see video 6min47).

That's it! You don't need a computer anymore!

It may happen that the patch is not accepted in the card at the first try. It is good to know that the ZIP card can only contain 64 objects maximum.

Moreover, the use of objects specific to the ZIP card only works on the ZIP card.

This card is not dedicated to sound or video use, but rather to electrical capture and redistribution.

[Partie 4]

In this last part we will study the specificities of MAX programming for the ZIP card.

We are going to create a musical sequencer with a transistor board.

The programming differs from external to autonomous. A patch can only have 64 objects maximum, if you want more, you have to create several patches and jump from patch to patch (jumper).

We will see the simple and specific objects.

he first object placed is a loadbang. The programming goes from top to bottom and from right to left. There is only one ana_7bits object and one num_in.

Regarding the programming of a sequencer, we play with time. (See the first part of the patch in the 3min44 video). It is used to establish a duration which elapses.

By bringing the electromagnets closer together we can see the time that is passing. We observe the time going from 0 to 5 seconds. We can send a bang to each electromagnet separately at each press of the button. (Demo 4min40).

After that, we will use buttons to capture the time between two pulses. To do this we use the num_in object. The 3 buttons used are placed on the 3 outputs of the num_in.

The first button calculates the duration, creating the rhythmic pattern. The second button generates a loop and allows to set the loop to infinity or not. The third button allows you to launch the pattern, trigger the loop.

These three functions are essential to this sequencer.

In the num_in (see 6min36).

How does the sequence look like? Thanks to the time that elapses (Z counter). The time runs out and will be banger only after a certain time. The buttons are all-or-nothing sensors, so they go either to 0 or to 127. (127 when the button is pressed; 0 when it is released).

The first button bangs the time received by the electromagnets. Since we are limited to 8 electromagnets, it was necessary to create an accumulator which makes it possible to count the steps, it goes until 8 and returns to 0 (see 9min39), which makes it possible to recreate the sequence once the 8 steps made.

The values of time are sent in the Gate then communicate to each electromagnet.

At the time of the creation of object in the ZIP card, there is a limitation of memory, a Gate_8 would take too much space then it is necessary to take a Gate_6. There are thus 6 outputs. In the sixth output, the toggle will have the function of output 7 and 8, which then return to 2.

Another important point, the rule of the limit of the Trigger. They take a lot of memory and therefore cannot carry more than 3 parameters and only Bang_0, Integger and 1.

The second button is a Trigger_bang. It launches the first delay triggering a domino effect (See 13min45). Each last Bang is used to activate the next one. The last Trigger_bang will go

inside the Gate, this is what defines the creation or not of a loop. The Gate will be opened or closed thanks to button N°3. If you press the button the pattern will be repeated.

As we are working in Midi interface, things are simpler. There is a value between 0 and 127. The electromagnets are activated at 127 and deactivated at 0, thus creating a percussive element.

We will therefore send a value of 127 for each of these electromagnets. Then, we must send the value 0, so that they do not remain constantly lit.

The sequencer is created (see 17min05), we must now choose these outputs. For it to be autonomous, we need a PWM_OUT and a PWM2_OUT.

Nota Bene = There is a restriction of the number of links (strings) per object. Only one string per output is allowed, so as not to have a string overlap error. Each feature takes memory, so we are restricted.

We can't use for this card: a sub-patch, but we can switch from one patch to another with the "jump" object. You can access the patch of recognized objects, if needed, directly in MAX. Above are all standard objects, externals and attractions that you can use. The parameters are specified.

We can conclude by integrating the patch into the sequencer created via the ZIP card. (See 21min03)

If the word "success" is displayed, it means that your patch is good and integrated into the card.