<Hat> manual

contents:

1. Hardware setup

Since there are a wide range of objects that can be mounted on <Hat>, it's hard to give a general explanation on how to attach them. But here are some general guidelines. Further in this manual you can read how you can calibrate <Hat> softwarematically, but always remember that a good adjustment of the hardware is crucial for it to work properly.

Note: **never let <Hat> play without having objects mounted**. Letting the beaters strike just in the air can damage them!

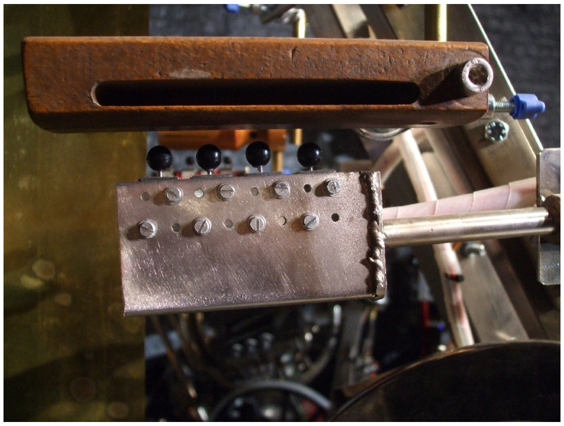
In general, objects should be mounted so that the beaters **do not reach their maximum trajectory** in striking the object, but just slightly less. If beaters do reach to their limit, that can cause damage to the magnets, and it causes an undesired clicking noise. When you install the objects closer to the hammer, you will loose resolution in loudness, but you will be able to play faster trills on them. (But note that the repeat rate is very high already for most beaters.)

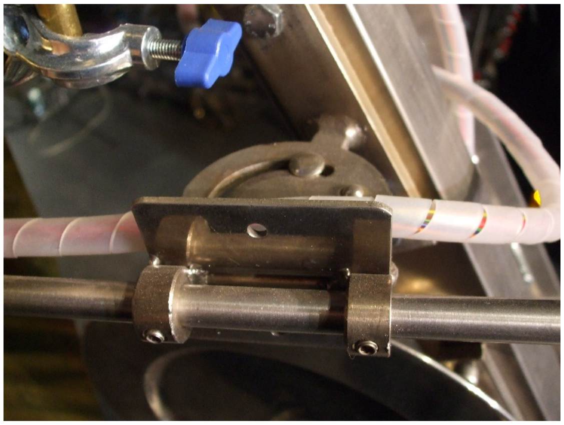
Attach all objects firmly. A **small shift** in height can make a **big difference** in **loudness** response!

Avoid mounting objects in any other metal than stainless steel. They can cause corrosion.

The different kinds of hammers have different characteristics. Here is a little oversight:

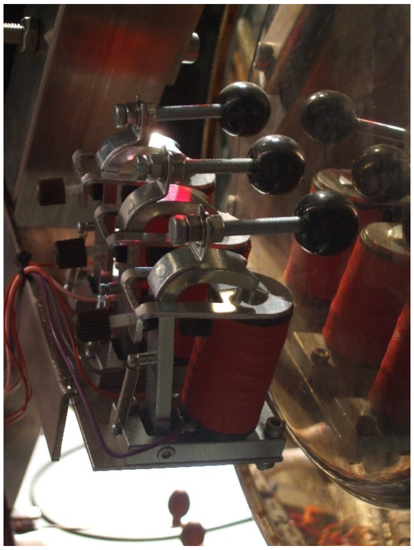
the 'hands'
note 48 - 55





- have quite a wide velocity range and fast repeat speed.
- their horizontal angle can be adjusted, so they can also beat objects that are not mounted on hat, but put next to it (for example cymbals, ...)

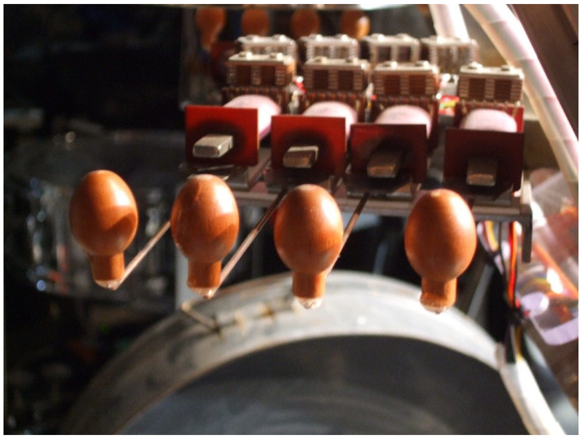
the vertical beaters
notes 56 – 61



- have an even bigger velocity range than the hands.
- the angle under which the beaters stand when still can be adjusted with the brown knob. In general, tilting them further backwards, thus giving them a longer beating trajectory, will improve the dynamics, but if tilted too far backwards, they might not work at low levels.
- the beater head is mounted on a screw that can be adjusted, but generally this is not necessary.

the upwards & 'back' beaters

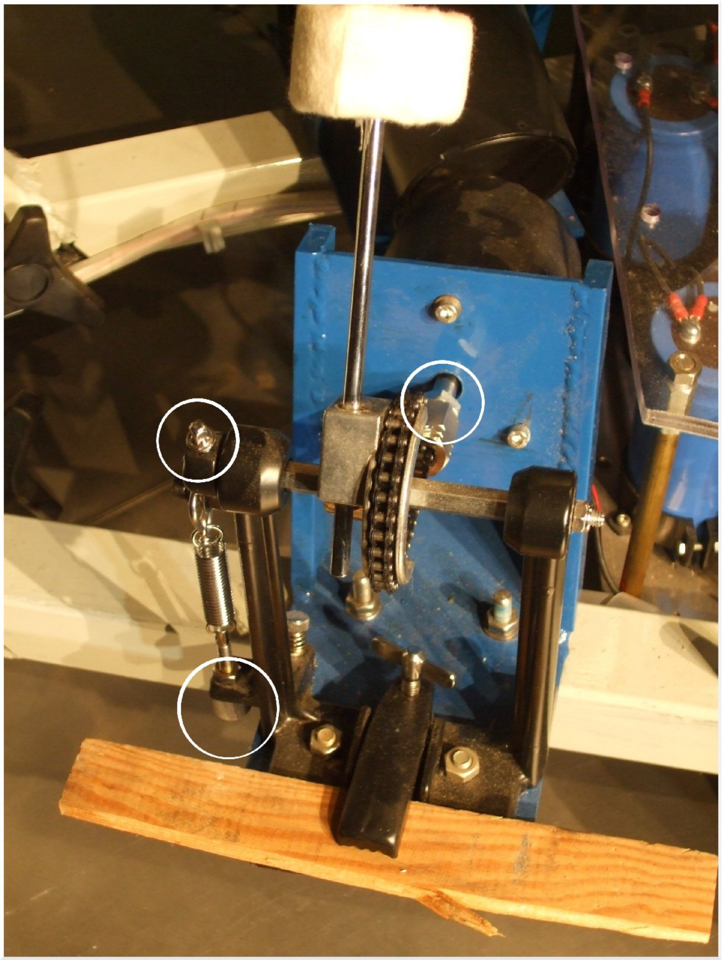
Notes 72 – 87



- have a lower dynamic range than the hands and vertical beaters, so they are best for objects that need a softer stroke
- the upward beaters are individually different in type, thus they differ in dynamic range. It may take some experimentation to find out which ones work best for which object.

the pedals

notes 36 and 37

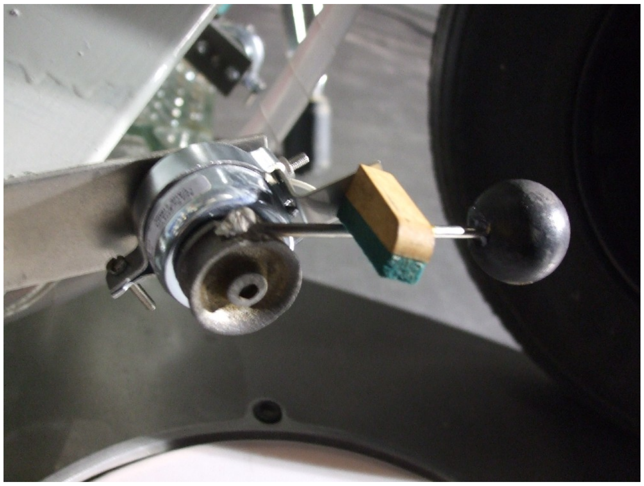


The pedals have a lot of adjustment possibilities. It's very important to arrange those according to the object you mount before you start calibrating with the software (see chapter 2.)

- by loosening the screw in the upper left circle, you can change the angle of the pedal
- with the screw in the lower circle, you can adjust the tension of the spring. If the spring is not tight enough, the pedal will not return after a beat, or only slowly, but by making it too tight,you limit dynamic range.
- You can also adjust how far the anchor of the magnet sticks out. Loosen the bolt and turn the anchor to shift it in our out. For maximum dynamic range, ensure that the anchor travels the furthest possible distance in its stroke. As with other striking mechanisms, place the struck object just close enough to eliminate the clicking noise that occurs when the plunger reaches its limit and strikes the body of the solenoid.
- by loosening the screw on the back of the beater, you can move the beater up or down.

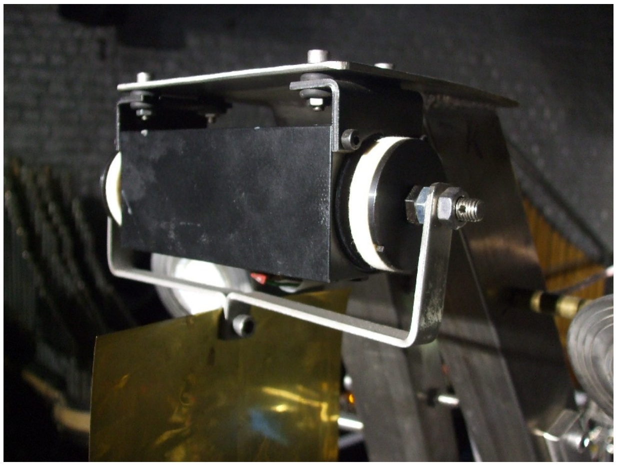
the 'toes'

notes 38 and 39



- designed to strike objects placed on the ground
- the bars the toes are mounted on can rotate to adjust the height. When transporting <Hat>, make sure they point not too low to the ground.
- also the placement of the damper (which is there because the beater makes some noise when it bounces back after striking) can be adjusted.

the shaker
note 40 and 41



- note 40 shakes to the left, 41 to the right, so you should always alternate between this notes

2. Getting started with the calibrate-hat program

Important note: whenever you switch off <Hat>, wait **at least 10 seconds** before you switch it on again!

Depending on which objects you attached to <Hat>, it needs to be calibrated so the midi velocities are mapped to meaningful values. For this you can use the program 'calibrate-hat.exe' .

Note: this program only works under Windows (Xp/Vista/7). On Linux you can use a Windows emulation like Wine .

Make sure all objects are attached firmly! A small shift in height can make a big difference in velocity response.

Here's a run through on how to calibrate <Hat>:

Step 1: To install the program, just copy the 'calibrate_hat' folder from the CD to a place of choice on your computer.

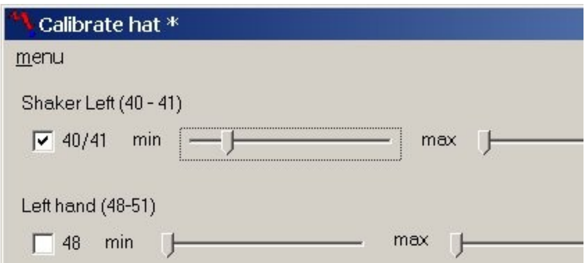
Step 2: Before running the program, make sure that <Hat> is **connected** to a **MIDI output** port on your computer and **powered on**.

Step 3: Go to the 'calibrate_hat' folder on your computer and double click 'calibrate-hat.exe' . You will be prompted to select a midi device. Choose the device <Hat> is connected to and click the 'continue' button.

Step 4: Next the main application window shows up. Here you find two checkboxes and two sliders (one for minimum loudness, one for maximum) for each beater on <Hat>, grouped by element. Those allow you to find the minimum and maximum velocities for each beater.

Now we need to determine the minimum and maximum velocities for each instrument.

Start by **clicking** the first **checkbox** (Shaker - 40/41) and **gradually moving its 'min' slider** right next to it to the **right**.



At some point, you will notice the shaker softly starts playing. Stop moving the slider further, or move it until you reach the **minimum loudness** you want to use.

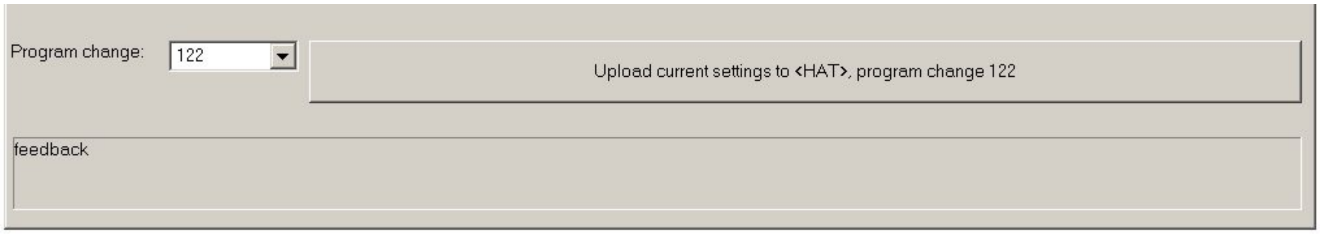
Click the checkbox again to deselect it, then click the checkbox to the right of it for the maximum. Now move the **'max' slider** first to the **same position** as the **'min'** one and then **gradually move** it further to the **right**. At first you will hear the sound level increasing, but at a certain level, moving the slider further will cease to affect an increase in dynamic, or, depending on which object you installed, it will simply begin sounding badly. Move the slider back to just below that point. Note that choosing a maximum higher than this point, only makes <Hat> and the objects that you install on it wear out, while it won't allow you to play any louder.

Now **repeat** this action for **each beater**.

You can at any time save the current settings by going to the menu and selecting 'save settings (as)'. This does not send the calibration data to <Hat> yet, but just stores the minimum and maximum levels locally on your computer. You can recall your settings with the 'load settings' option in the menu.

Step 5

Once you determined the right levels for all elements, you are ready to send the calibration data to <Hat>. **Choose** the **number** (122 – 126) of the bank you want to save the current data to (you can recall the bank later on by sending a program change with the corresponding number).



Program change: 122

Then click the '**Upload current settings**...' button.

If you didn't save the latest settings to your computer, you will be prompted to do so now. This is not strictly necessary, but its' a good idea to do so, as you can store more presets locally than can be stored in <Hat>'s memory, and remembering this setting may be a handy starting point when you want to do calibration for a similar setup.

Next the data is sent to <Hat>. This takes a while. A new window appears to show the progress. Make sure **no other MIDI data is sent** to <Hat> as long as the calibration is busy!

During the calibration, you will see a green led on <Hat> blinking every 3 seconds, followed by a red led blinking (there are 2 red leds, each on a different circuit board. One of each is blinking, depending on which note is being calibrated). If you don't see both the green and red led blinking, something is going wrong and <Hat> is not receiving the data. Check your midi setup and cabling in this case.

When the main application window appears again, the calibration is ready. From your sequencer (or other MIDI application), send the program change with the corresponding value (122 – 126) to use the current mapping. Now midi velocity 1 will correspond with the minimum loudness you chose, 127 with the maximum.

Note: <Hat> does remember the calibration tables, but does not remember which one you selected the latest when you switch it off and back on, so remember to always **send the program change again after powering it on**! It's a good practice to put the program change at the start of all your midi sequences.

Some final remarks.

It's not possible to get any feedback from <Hat> on what settings is being uploaded currently, but when you saved the settings before doing the calibration, the program will remember which settings where last uploaded to which table. You can see which settings have been uploaded by choosing 'overview of settings currently uploaded..' from the menu, which will bring up a window showing which settings files are uploaded to which tables (provided you did save the settings).

Note: if you move the program to another computer, you can copy these settings along by copying the 'hat_programs.dat' file from the old installation directory to the new one. Of course you also need to move the files with the setups to the other computer.

3 <Hat> midi specification

<Hat> is listening to **midi channel 10.**

Controller 66 is used to switch the solenoid power on (non-zero value) or off (0). You should always send this controller after you switched on <Hat>, otherwise the beaters won't work. It's a good practice to put this controller at the start of all your midi sequences. If you are not playing anything on <Hat> for a while, it's safest to set this controller to 0.

Note on commands make the beaters strike and make the lights switch on.

Beaters:

36 left pedal

37 right pedal

38 left toes

39 right toes

40-41 shaker (always alternate between this two notes)

48-51 left hand

52-55 right hand

56-58 vertical beater left

59-61 vertical beaters right

72-75 upwards left

76-83 back beaters

84-87 upwards right

Lights:

96 light left eye

97 light right eye

98 light frontal

99 light frontal

Note Off commands are not required for the beaters. Lights are switched off with note off commands. Note on with velocity 0 has the same function as note off.

Program Change: implemented to change to different user programmable velocity scaling lookup tables. When <Hat> is powered on, it's set to table 0, but for practical use you'll need to fill one of the programable tables (122 – 126) using the calibration program (see chapter 2 here above). It's a good practice to put a program change at the start of all your midi sequences.

Note: program change 127 selects a special table that is only meant to be used by the calibration program. It has an extremely wide range, of wich the loudest values can cause damage to <Hat> or the objects mounted on it if used without caution.

controller 123 (midi all-notes-off command) switches off all lights.

Sysex: implemented for user programmable velocity scalings. See specification below.

4. Sysex specification

You can send calibration data to <Hat> yourself with a sysex. Use caution if you ever do so. **Incorrect data** – especially velocity values that are too high – **can damage the instrument**.

It is highly recommended to always use the calibration program we delivered, which will do the sysex building and sending for you.

In case you do want to send the sysex yourself, you have to send a **separate sysex for each note**, which contains a table that makes the link between midi velocity and the actual beating force. This force is determined by the length of the electric pulse sent to the beater, and is expressed in PIC processor ticks.

For the table used by the calibration program, the nr of ticks varies from 40 (for velocity 1) to 1056 (velocity 127), but those are extremes that are respectively lower and higher than you will normally ever need for practical use.

After sending a sysex command, wait **at least 3 seconds** before sending another one.

The sysex format:

- each sysex starts with the following two bytes: *0xF0 0x7D*
- the following 4 bytes are note dependent. For notes lower than 72, they are *0x68 0x61 0x74 0x20* , for higher notes *0x63 0x61 0x73 0x32*
- next the byte *0x0A*
- then one byte corresponding to the table you want to fill (*0x7A* – *0x7E*). This determines which program change you will need to send later on for this table.
- next one byte corresponding to the note for which you want to send a lookup table.
- next two 0 bytes: *0x00 0x00*
- then 127 times a couple of bytes, corresponding to the actual nr of ticks for each velocity . This value should **never** be **0** and **never** be **higher then 1056**. The first of the two bytes corresponds to the MSB of the value, the second to LSB.
- the sysex ends with the byte 0xF7

When the sysex is well received, you will first see a green led blinking, then a red one. (There are two different LED's, each on a separate circuit board. For notes below 72 the LED on the lowest board will blink, for higher ones the one on the upper board.)

5. Further documentation and support

Detailed technical information can be found on the webpage http://logosfoundation.org/instrum_gwr/HAT.html .

Support questions can be addressed preferably by email to kristof.lauwers@logosfoundation.org.

We can also be reached by phone on working days between 9pm and 5am CET on the number +32 92 23 80 89 .