# MIDI Captain Series Controllers

## Super Mode Functionality and Configuration Instructions

MIDI Captain ONE

MIDI Captain DUO

MIDI Captain NANO 4

MIDI Captain MINI 6

MIDI Captain GOLD / BLUE / STD 10 SW

ONE DUO NANO 4 MINI 6 GOLD BLUE STD

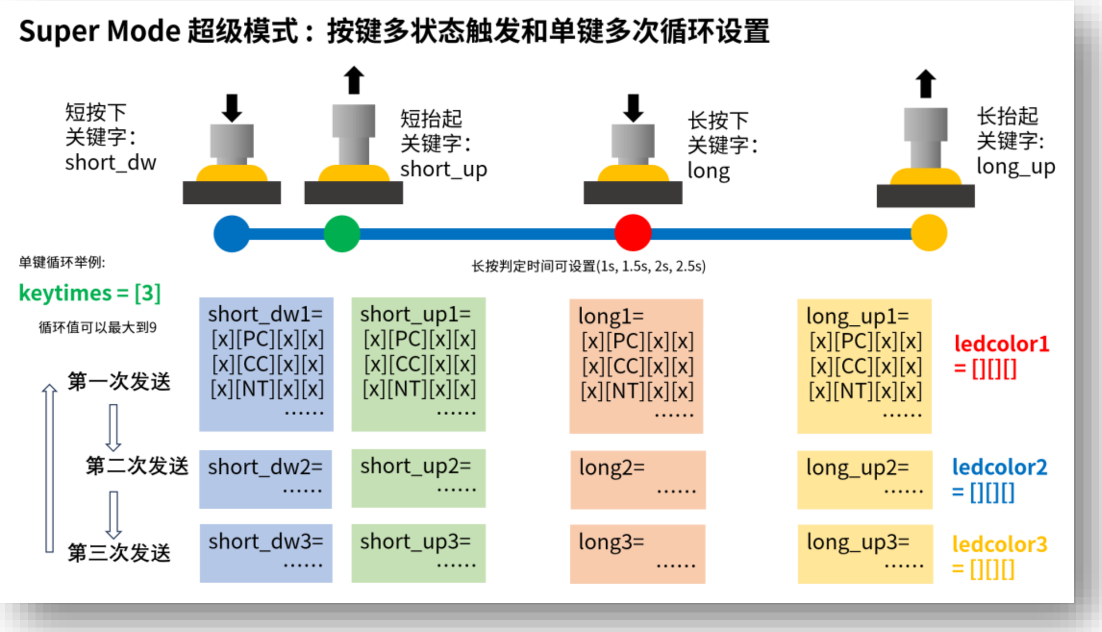
V1.2 Jan 2024

## Contents

# Super Mode Functionality and Configuration Instructions

## Overview

Super Mode is the factory default firmware for the MIDI Captain ONE / DUO / NANO 4 / MINI 6, the 1 / 2 / 4 / 6 switch versions. The MIDI Captain 10 switch Blue/Gold/Black STD versions can also acquire Super Mode control capabilities through firmware upgrades. Super Mode integrates multi-state buttons, multi-command MIDI control combinations, and simulation functions for USB-HID keyboards and mouse click. It offers extremely rich and flexible configuration capabilities, meeting the control needs of most host devices or software/Apps with MIDI IN control, such as effect processors, Amplifiers, synthesizers, and computer music software.



**Super Mode 超级模式：按键多状态触发和单键多次循环设置**

| | | | | | |
|---|---|---|---|---|---|
| 第一次发送 | short_dw1=<br>[x][PC][x][x]<br>[x][CC][x][x]<br>[x][NT][x][x]<br>...... | short_up1=<br>[x][PC][x][x]<br>[x][CC][x][x]<br>[x][NT][x][x]<br>...... | long1=<br>[x][PC][x][x]<br>[x][CC][x][x]<br>[x][NT][x][x]<br>...... | long_up1=<br>[x][PC][x][x]<br>[x][CC][x][x]<br>[x][NT][x][x]<br>...... | ledcolor1<br>= [][][] |
| 第二次发送 | short_dw2=<br>...... | short_up2=<br>...... | long2=<br>...... | long_up2=<br>...... | ledcolor2<br>= [][][] |
| 第三次发送 | short_dw3=<br>...... | short_up3=<br>...... | long3=<br>...... | long_up3=<br>...... | ledcolor3<br>= [][][] |

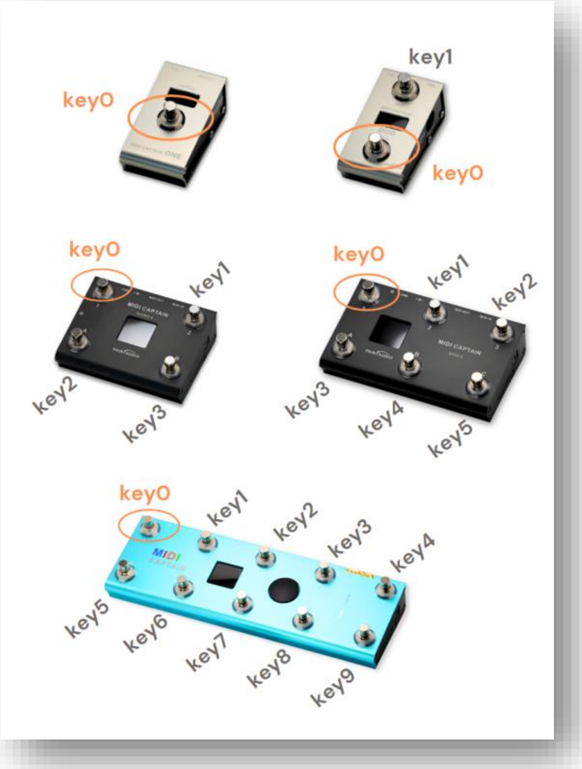
In Super Mode, different MIDI control command sets can be configured for the keypress at four different moments: short press down (short_dw), short press release (short_up), long press down (long), and long press release (long_up). These command sets can be a combination of multiple types of instructions, including PC commands, CC commands, and Note commands. Additionally, at the short press down moment (short_dw), it can be configured as a combination of keyboard or mouse actions, enabling control operations such as mouse clicks, keyboard shortcuts, and combined key presses on computers or tablet devices. This enriches the control capabilities in music software usage, player control, page-turning control, and even in-game key controls. Keyboard and mouse commands, can also be mixed with MIDI control commands.

The MIDI Captain 10-switch version can have three different control programs: Normal Mode, Geek Mode, and Super Mode, which can be switched by holding down different buttons while powering on. Below are the power-on operations for switching between different programs after the latest update.

- Hold the A button while powering on to enter Normal Mode
- Hold the B button while powering on to enter the 'Full Bright/Half Bright' Normal Mode, which has the same main functions as the regular Normal Mode, but the difference is that the LED indicator changes between full brightness and half brightness. Both modes share the same button function configuration parameters, and the configuration content is in the setup directory under USB mode (the way to enter USB mode will be mentioned later).
- Hold down key 2 while powering on to enter Geek Mode

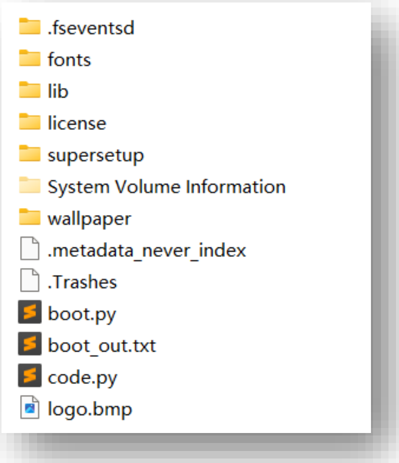
- Hold down key 3 while powering on to enter the 'Full Bright/Half Bright' Geek Mode, which shares the same key configuration as the regular Geek Mode. The configuration files are located in the geeksetup directory under USB mode.
- Hold down the C key while powering on to enter Super Mode, which is the mode this manual focuses on. The configuration files are located in the supersetup directory under USB mode.

## Enter USB Setup Mode



The image shows the key0 and other key numbering mapping rules for different versions of controllers. MIDI Captain series controllers with 1, 2, 4, 6, 10 switches all use the same method to enter USB configuration mode, that is by holding down the key0 button while powering on, and using a USB cable to connect to a computer, supporting Windows 8/9/10/11 and Mac operating systems. Please refer to the manual for controllers of the key0 button location.

After the above operations, the computer will recognize the USB device, and you will see a new disk drive in the file system, named X:\MIDICAPTAIN. Upon entering the drive, you can see a file directory and structure similar to the one below, with slight variations between different versions. The following is just an example:



.fseventsd
fonts
lib
license
supersetup
System Volume Information
wallpaper
.metadata_never_index
.Trashes
boot.py
boot_out.txt
code.py
logo.bmp

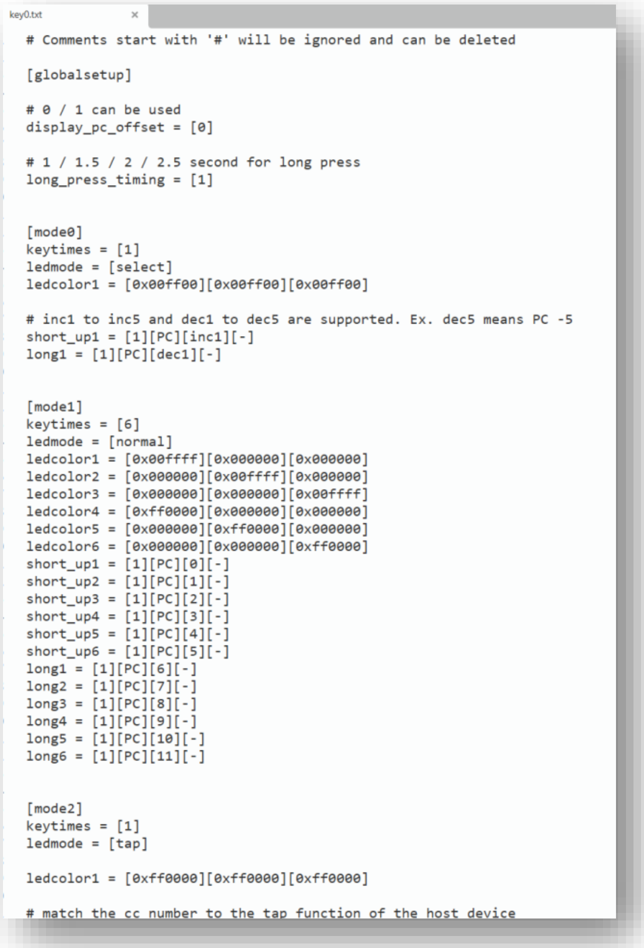
Within this file structure, depending on the device version, you may find folders named setup, geeksetup, or supersetup, which are the key configuration folders for different modes. After entering, you can find the corresponding keys or pages to set their respective functions. The license directory is the hardware authorization directory of the device. Please back it up to your computer promptly after first entering USB mode, so that if it is accidentally deleted or lost, you can copy it back, and the device can be restored to use. Of course, if you lose this file directory and its contents and have not backed it up in time, you can also go to www.paintaudio.com to the corresponding product page, find the FAQ section at the bottom, and follow the steps in Q20 to restore the device's authentication information. Alternatively, you can contact us at support@paintaudio.com or Wilson-zuo@paintaudio.com to remotely resolve this issue. The device is always recoverable.

The wallpaper folder named 'wallpaper' listed above is the default wallpaper used by the program during operation. In the future, we plan to offer more wallpaper downloads on the official website. You can also create your own personalized wallpapers according to the format requirements. renamed such as wpX.bmp, for the device to recognize and use it. The method for setting the wallpaper will also be mentioned later.

# Keywords and Configuration Explanation for Super Mode

## MODE Concept for MIDI Captain ONE/DUO

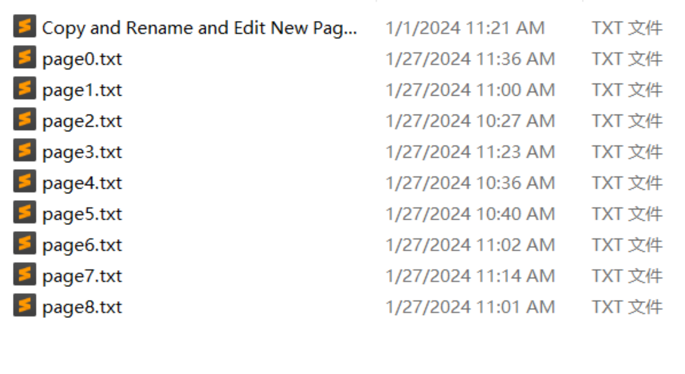
Firstly, Super Mode has a completely similar program organization and configuration method on both MIDI Captain ONE 1-switch and MIDI Captain DUO 2-switch. Each button has 4 completely independent function groups Mode 0 1 2 3, which can be switched in real-time using the device's mode switch. By entering the X:\setup\ directory and opening the corresponding button's configuration file keyX.txt, we can see the following file content. Note that at the very top of key0.txt, the [globalsetup] section contains the device's global settings, and the settings for the button itself begin from under [mode0]. In other button files, there is no content for global settings.



```
# Comments start with '#' will be ignored and can be deleted

[globalsetup]

# 0 / 1 can be used
display_pc_offset = [0]

# 1 / 1.5 / 2 / 2.5 second for long press
long_press_timing = [1]


[mode0]
keytimes = [1]
ledmode = [select]
ledcolor1 = [0x00ff00][0x00ff00][0x00ff00]

# inc1 to inc5 and dec1 to dec5 are supported. Ex. dec5 means PC -5
short_up1 = [1][PC][inc1][-]
long1 = [1][PC][dec1][-]


[mode1]
keytimes = [6]
ledmode = [normal]
ledcolor1 = [0x00ffff][0x000000][0x000000]
ledcolor2 = [0x000000][0x00ffff][0x000000]
ledcolor3 = [0x000000][0x000000][0x00ffff]
ledcolor4 = [0xff0000][0x000000][0x000000]
ledcolor5 = [0x000000][0xff0000][0x000000]
ledcolor6 = [0x000000][0x000000][0xff0000]
short_up1 = [1][PC][0][-]
short_up2 = [1][PC][1][-]
short_up3 = [1][PC][2][-]
short_up4 = [1][PC][3][-]
short_up5 = [1][PC][4][-]
short_up6 = [1][PC][5][-]
long1 = [1][PC][6][-]
long2 = [1][PC][7][-]
long3 = [1][PC][8][-]
long4 = [1][PC][9][-]
long5 = [1][PC][10][-]
long6 = [1][PC][11][-]


[mode2]
keytimes = [1]
ledmode = [tap]

ledcolor1 = [0xff0000][0xff0000][0xff0000]

# match the cc number to the tap function of the host device
```

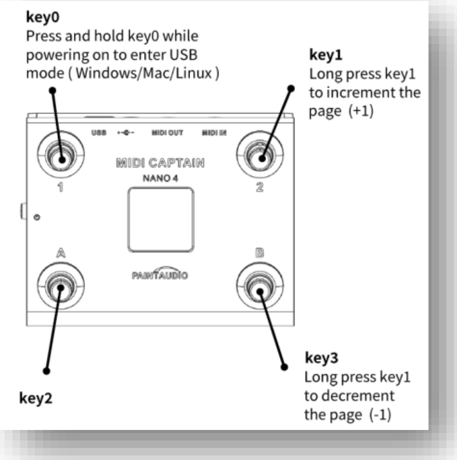
## PAGE Concept for MIDI Captain 4/6/10 switch versions

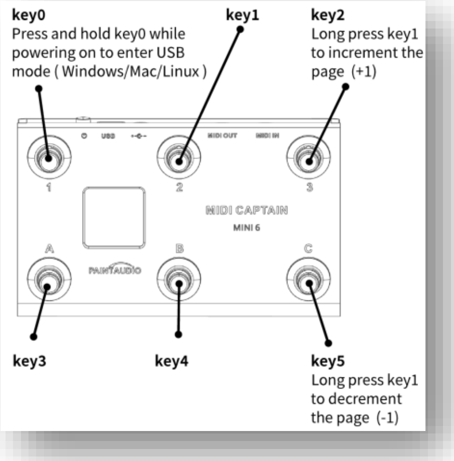
In the MIDI Captain NANO 4 / MINI 6 and 10-switch versions, the concept of a page is used to organize the functions of all buttons within the same page, and we can configure up to 99 different pages to achieve completely different button functions. This means that we can pre-arrange the control content for different host devices, or commands that exceed the total number of the foot switches, into different pages, and can switch pages in real-time during use to call up different configurations. We can see multiple pageX.txt files under MIDICAPTAIN\supersetup\, each containing the function configuration for all buttons on the current page. We can copy, paste, and sequentially rename more pages ourselves, with a maximum limit of 99 pages.



| | | |
|---|---|---|
| Copy and Rename and Edit New Pag... | 1/1/2024 11:21 AM | TXT 文件 |
| page0.txt | 1/27/2024 11:36 AM | TXT 文件 |
| page1.txt | 1/27/2024 11:00 AM | TXT 文件 |
| page2.txt | 1/27/2024 10:27 AM | TXT 文件 |
| page3.txt | 1/27/2024 11:23 AM | TXT 文件 |
| page4.txt | 1/27/2024 10:36 AM | TXT 文件 |
| page5.txt | 1/27/2024 10:40 AM | TXT 文件 |
| page6.txt | 1/27/2024 11:02 AM | TXT 文件 |
| page7.txt | 1/27/2024 11:14 AM | TXT 文件 |
| page8.txt | 1/27/2024 11:01 AM | TXT 文件 |

The method for switching pages is to long-press the two buttons on the right side of the 4, 6, 10-switch version MIDI controllers, to achieve page+ and page-. The diagram below uses the 4, 6-switch version as an example to demonstrate the method of page switching.


key0
Press and hold key0 while powering on to enter USB mode ( Windows/Mac/Linux )
key1
Long press key1 to increment the page (+1)
USB
MIDI OUT
MIDI IN
MIDI CAPTAIN
NANO 4
1
2
A
B
PAINTAUDIO
key2
key3
Long press key1 to decrement the page (-1)


key0
Press and hold key0 while powering on to enter USB mode ( Windows/Mac/Linux )
key1
key2
Long press key1 to increment the page (+1)
USB
MIDI OUT
MIDI IN
1
2
3
MIDI CAPTAIN
MINI 6
A
B
C
PAINTAUDIO
key3
key4
key5
Long press key1 to decrement the page (-1)

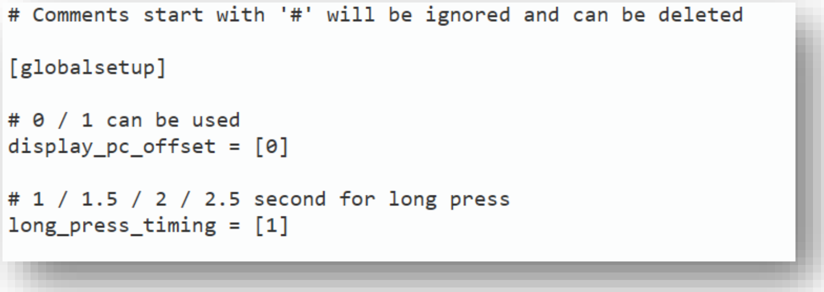
It is important to note that since the long-press page switching function of the rightmost button is fixed, please do not configure long-press and long-press release functions on the two right-side buttons again, as it will execute simultaneously with the page change, leading to undesired page move.

## Explanation of Keywords and Configuration Content

Notes of the formats: In the configuration file, please preserve the position and structure of the brackets [ ], all keywords and configuration content are case-sensitive and should not be changed, all commands configured for the same key trigger state should be xxx=[][][][][][][][]...Do not use line breaks. Extra spaces and blank lines do not affect the configuration. Lines starting with # are comment lines, which can be freely deleted or added. If the configuration format is incorrect, it will cause the program to be unresponsive or display error messages. You can always restore the operation of the device by copying back the backed-up configuration file or the configuration file from the updated program package.

## [globalsetup] Device Global Settings

Open the configuration file key0.txt, at the top [globalsetup] is the **global settings** keyword, and the content below it is the global settings for the device. Note: Global settings can only exist at the top of the key0.txt configuration file. The global settings content for devices with different switch versions is not exactly the same, below are typical global configurations for 1/2 switch and 4/6/10 switch devices, and the configuration content is explained respectively. The following image is the globalsetup for 1/2 switch:



```
# Comments start with '#' will be ignored and can be deleted

[globalsetup]

# 0 / 1 can be used
display_pc_offset = [0]

# 1 / 1.5 / 2 / 2.5 second for long press
long_press_timing = [1]
```

Ignoring comment lines, there are only two configuration items.

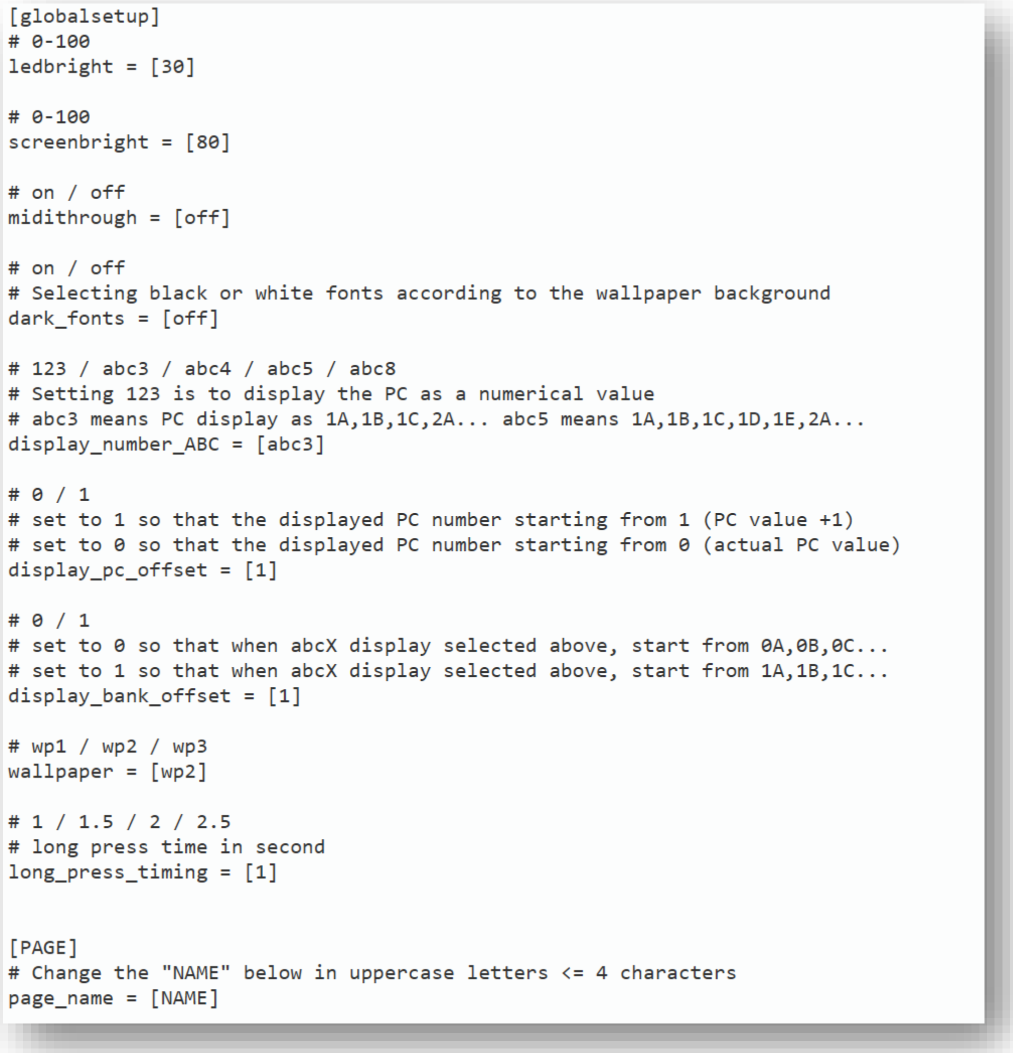
### display_pc_offset = [0]

Inside this square bracket, you can fill in 0 or 1, with the default 0. This means that when we press a key to change a host tone preset, which is the MIDI program change (often abbreviated as PC command), since the control value range of this command is 0-127, but some hosts' preset numbers start from 1 (still called by PC 0), there will be a situation where the PC command number displayed on the MIDI Captain is off by 1 from the host's display. In this case, setting display_pc_offset = [1] can eliminate this display misalignment. Note that this setting does not change the actual PC value sent; it just adds 1 to the display value.

### long_press_timing = [1]

This bracket can be filled with 1 / 1.5 / 2 / 2.5, which represents the time in seconds for a keypress to be considered a long press. By modifying this parameter, we can set the activation time for the long press function to 1 second, 1.5 seconds, 2 seconds, or 2.5 seconds. The default is 1 second.

Now let's look at the global settings for the 4 / 6 / 10 switch versions:



```
[globalsetup]
# 0-100
ledbright = [30]

# 0-100
screenbright = [80]

# on / off
midithrough = [off]

# on / off
# Selecting black or white fonts according to the wallpaper background
dark_fonts = [off]

# 123 / abc3 / abc4 / abc5 / abc8
# Setting 123 is to display the PC as a numerical value
# abc3 means PC display as 1A,1B,1C,2A... abc5 means 1A,1B,1C,1D,1E,2A...
display_number_ABC = [abc3]

# 0 / 1
# set to 1 so that the displayed PC number starting from 1 (PC value +1)
# set to 0 so that the displayed PC number starting from 0 (actual PC value)
display_pc_offset = [1]

# 0 / 1
# set to 0 so that when abcX display selected above, start from 0A,0B,0C...
# set to 1 so that when abcX display selected above, start from 1A,1B,1C...
display_bank_offset = [1]

# wp1 / wp2 / wp3
wallpaper = [wp2]

# 1 / 1.5 / 2 / 2.5
# long press time in second
long_press_timing = [1]


[PAGE]
# Change the "NAME" below in uppercase letters <= 4 characters
page_name = [NAME]
```

The 4 / 6 switch versions have a richer set of global settings:

Among them, display_pc_offset and long_press_timing are defined the same as the 1 / 2 switch versions, please refer to the previous settings instructions.

### ledbright = [30]

This setting is for the foot switch LED brightness, ranging from 0-100, with a recommended range of 20-50. Going higher will significantly increase power consumption. The default is 30.

### screenbright = [80]

This item is for setting the screen brightness, with a range between 0-100.

### midithrough = [off]

You can enter 'on' or 'off' to represent turning on or off the device's MIDI Through function. When enabled, the device will automatically forward MIDI commands and signals received from the MIDI IN port to the MIDI OUT port.

### darkfonts = [off]

For this item, 'off' means the screen will display light-colored fonts, suitable for dark backgrounds, while 'on' means the display will be in dark-colored fonts, suitable for light backgrounds.

### display_number_ABC = [123]

The content within the square brackets for this item can be 123, abc3, abc4, abc5, abc8, which means that when the command is a PC command, it can display the PC number value, or it can be converted into a format like 1A, 1B, 1C, with abc3 representing grouping by 3, abc4 representing grouping by 4, displayed as 1A, 1B, 1C, 1D, then 2A, 2B... abc5 and abc8 also differ in their grouping factors, and this design is to accommodate different hosts' grouping methods.

### display_bank_offset = [1]

This setting is designed to change the starting value of the group number display, which can be either 0 or 1. If set to 0, the first group will display as 0A, 0B, 0C... If set to 1, the first group will display as 1A, 1B, 1C... etc.

### wallpaper = [wp2]

This setting can change the background display or wallpaper on the screen. The content within the square brackets must be the actual wallpaper file name located in the wallpaper directory, without the file extension. Wallpapers can be created by the user, but they must be 240x240 pixel bmp files, and converted to either 8bit or 16bit.

The above are all the global settings available. In the switch version, there are also settings for the expression pedal port MIDI channel value and CC value, which can be customized by referring to the comments in the specific configuration file.

In the 4 / 6 / 10-switch versions, due to the multi-page feature design, for ease of memory, we can name each page with a 4-letter or less uppercase acronym. Please refer to the following [PAGE] content settings:

## [PAGE] Name Setting

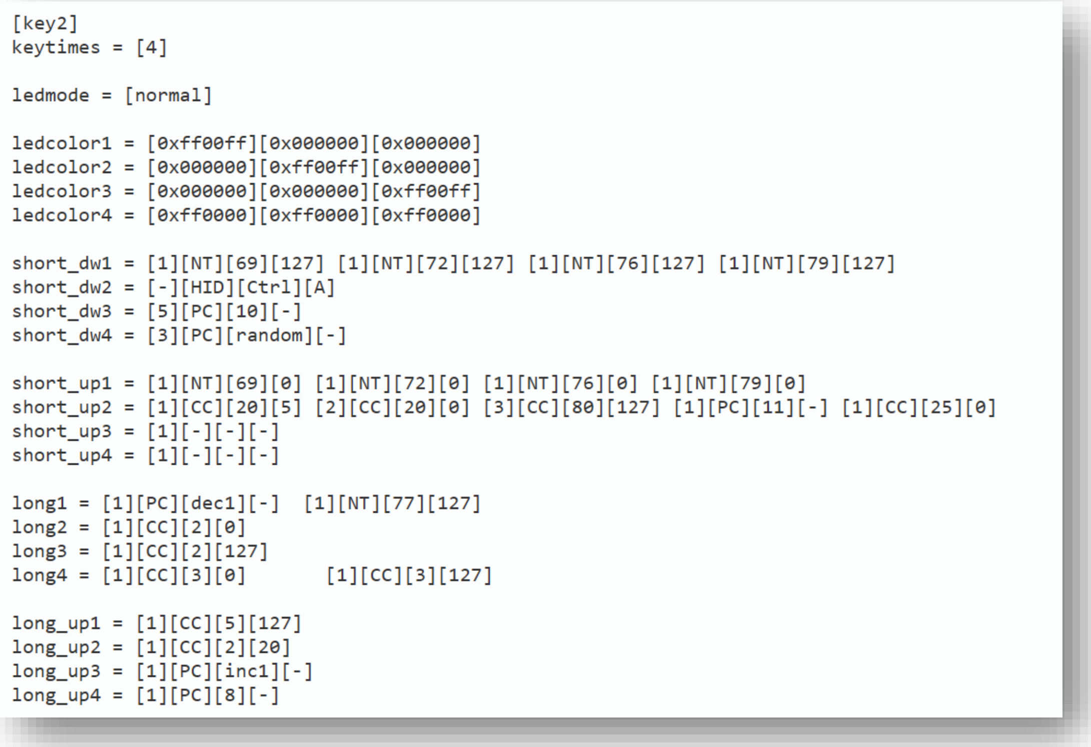
page_name = [NAME]

In this item, the brackets can contain any ⩽ 4 uppercase letters or numbers, for easy identification and memory of the page's button functions. For example, ABCD, PC8, S1, etc.

## [keyX] Setting

Now let's move on to the core button function configuration, which includes button presses, LED configuration, and command configuration. The commands can be MIDI instructions and HID keyboard/mouse instructions, and can also be used in combination. We'll start with MIDI instructions. Note that the entry point for button function configuration is slightly different between the 1 / 2-switch and the 4 / 6 / 10-

switch versions. For the 1 / 2-switch, the configuration starts below the mode item [modeX] in the corresponding keyX.txt file, while for the 4 / 6 /10-switch, it starts below the corresponding button item [keyX] in the respective pageX.txt file. The configuration method and content are almost exactly the same. Below, we will use a configuration file that almost fully utilizes the configurable items as an example to explain the meaning of each button configuration item, noting that some content in this configuration may not have practical significance.



```
[key2]
keytimes = [4]

ledmode = [normal]

ledcolor1 = [0xff00ff][0x000000][0x000000]
ledcolor2 = [0x000000][0xff00ff][0x000000]
ledcolor3 = [0x000000][0x000000][0xff00ff]
ledcolor4 = [0xff0000][0xff0000][0xff0000]

short_dw1 = [1][NT][69][127] [1][NT][72][127] [1][NT][76][127] [1][NT][79][127]
short_dw2 = [-][HID][Ctrl][A]
short_dw3 = [5][PC][10][-]
short_dw4 = [3][PC][random][-]

short_up1 = [1][NT][69][0] [1][NT][72][0] [1][NT][76][0] [1][NT][79][0]
short_up2 = [1][CC][20][5] [2][CC][20][0] [3][CC][80][127] [1][PC][11][-] [1][CC][25][0]
short_up3 = [1][-][-][-]
short_up4 = [1][-][-][-]

long1 = [1][PC][dec1][-]  [1][NT][77][127]
long2 = [1][CC][2][0]
long3 = [1][CC][2][127]
long4 = [1][CC][3][0]       [1][CC][3][127]

long_up1 = [1][CC][5][127]
long_up2 = [1][CC][2][20]
long_up3 = [1][PC][inc1][-]
long_up4 = [1][PC][8][-]
```

In the above configuration, we have comprehensively configured the action and function of the key2 button. Please refer to the diagram in Section 2 or the instruction manual to find the corresponding relationship for the physical button associated with key2.

Before introducing the meaning of specific entries, let's take a look at the basic format of the configuration commands. There is no strict sequence required for all the configuration content below [key2], but it is better to follow the order shown in the example for ease of understanding. The keyword names for each item cannot be changed in case or spelling. Entries can be separated by blank lines or comments starting with a #. No matter how many instructions are in the same entry, they must be on the same line without line breaks, ensuring that the left side of the "=" sign is the configuration item keyword name, and the right side is the setting value or command group enclosed in [ ]. If it is a command group, the number of [ ] must be a multiple of 4, meaning every four [ ][ ][ ][ ] constitutes a complete instruction unit. Commands can have arbitrary spaces added between them but must not contain line breaks. The first [ ] contains the MIDI channel number, which can be a value between 1-16, the second [ ] contains the command type, which can be one of PC, CC, NT, HID, or [-] if no action is desired.

If the command type is PC, the third [ ] can contain a PC value, or one of inc1, inc2, inc3, inc4, inc5, which represent incrementing the PC value by 1, 2, 3, 4, or 5 upon each trigger. Conversely, it can also be dec1, dec2, dec3, dec4, dec5, which correspond to decrementing the PC value by 1, 2, 3, 4, or 5. You can also fill in random, which means that a random PC value between 0-127 will be sent each time it is triggered. When the command type is PC, the content in the fourth [ ] is meaningless and can be filled with [-]. We have summarized the special commands supported by PC as follows:

### [1][PC][inc1][-]

inc1 can also be inc2, inc3, inc4, inc5, which increase the current PC value by 1, 2, 3, 4, or 5

### [1][PC][dec1][-]

dec1 can also be dec1, dec2, dec3, dec4, dec5, to decrease the current PC value by 1, 2, 3, 4, or 5

### [1][PC][random][-]

Send a random PC value between 0-127.

If the command type is CC, the third and fourth [ ] are for CC number and CC value, respectively, which are the number of the CC and the value for that number. By combining CC commands, a variety of action types can be created, such as toggle, latch, momentary, etc.

If the command type is NT, it means sending a note command. In this case, we need to look up the correspondence between 0-127 and the note chart, and fill in the corresponding note value and velocity value in the third [ ] and fourth [ ]. A velocity value of 0 can be used to end a note. Complex chords can be achieved with a combination of multiple commands, and chord progressions can be played with a single button by configuring the number of cycles for that button.

If the command type is HID, it means sending keyboard and mouse emulation commands, used to emulate the pressing of a certain key or combination of keys on the keyboard, or even to send a sequence of key presses with a single press. It can also emulate the clicking of the left or right mouse button. The specific contents that can be emulated will be detailed in the HID section later.

Let's now explain the previous configuration examples one by one:

### keytimes = [4]

This item indicates that the key can perform a cycle of sending different commands with four presses. That is, the first press sends the command configured for the first time, including the configuration content of ledcolor1, short_dw1, short_up1, long1, long_up1. The second press sends the content defined in ledcolor2, short_dw2, short_up2, long2, long_up2, and so on. After the fourth press, it returns to the first one. Of course, the four states of the key do not trigger all at once during a single press. If it is a short press, only the short_dw and short_up states will trigger. If it is a long press, the short_dw, long, and long_up states will trigger. Generally, to avoid the overlapping triggering of short and long presses, if you need to define a long press, we usually do not define the function of a short press down short_dw. In this way, short_up and long become independent functions that can be completely controlled by the duration of the press. When filling in the configuration, even if keytimes = [1], you cannot omit the number 1 in short_dw1, short_up1, long1, long_up1. The states that are not needed among these four key states can be left unexpressed, as shown in the configuration below, only short_dw1 is written on.

```
[key0]

keytimes = [1]

ledmode = [normal]

ledcolor1 = [0xff0000] [0xff0000] [0xff0000]

short_dw1 = [1][PC][5][-]
```

### ledmode = [normal]

This configuration represents the action mode of the button indicator light. You can choose one among normal, select, tap. 'normal' means that when the button is pressed, the LED color immediately changes to the color defined in ledcolorX. In loop buttons, different ledcolorX can be defined to achieve color changes, on and off, and even 3-segment control of the light ring. 'select' means that for all foot switches defined as select mode, only the currently pressed switch will light up, achieving a selected effect. 'tap' is a LED blinking mode specially configured for speed tapping, regardless of the actual command sent, the LED will blink, and the blinking frequency is determined by the interval between two consecutive presses. It is important to note that when using tap mode, for 4 / 6 / 10 switch versions, please do not configure complex commands for the switch, usually only a fixed value CC command that the host can accept is needed, like the following:

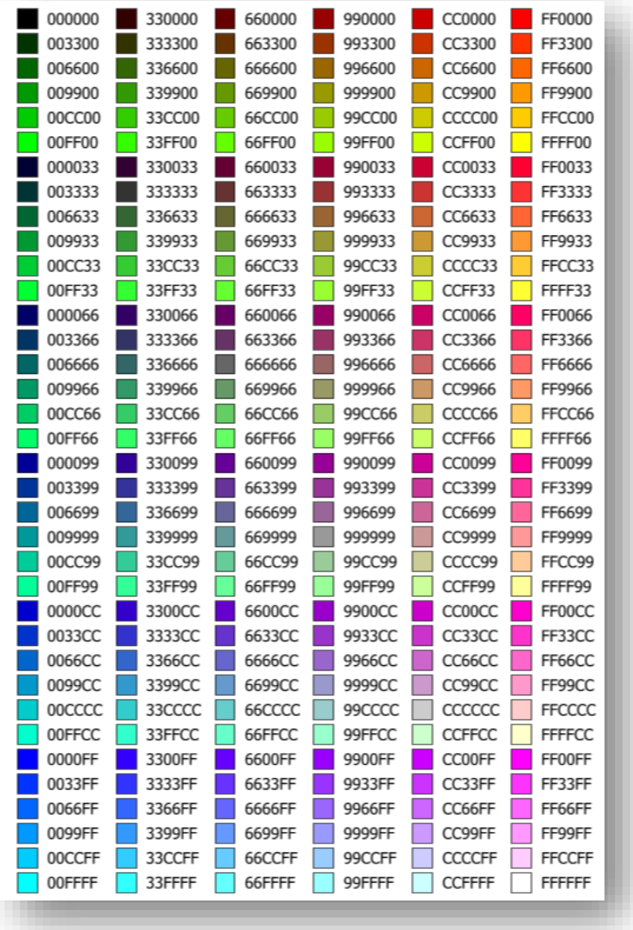
```
[key0]
keytimes = [1]
ledmode = [tap]
ledcolor1 = [0xff0000] [0xff0000] [0xff0000]
short_dw1 = [1][CC][15][127]
```

As a result, the display content does not need to change, reducing the program running block caused by display refresh, thus allowing for faster tapping settings or, in other words, responding to lower tapping intervals. If you find that sometimes the program does not respond to very fast tapping actions, this is normal. Please try tapping multiple times. The range of tapping speed settings is from 40 to 200 beats per minute. It should be noted that the tapping value displayed on the MIDI Captain is the result of local calculations by the device and does not have a synchronization and calibration relationship with the host. Moreover, the actual tapping speed value of the host is calculated based on the timing of the host receiving CC commands twice or more, which is not equivalent to the timing of foot tapping. Therefore, there may be a small difference between the displayed tapping speed and the host's displayed tapping speed value, which is normal. The flashing speed and the speed value of the controller's tapping are both intended as references.

### ledcolorX = [0xff0000][ 0x00ff00][ 0x0000ff]

This item is for setting the color values of the indicator lights. Super Mode fully unlocks the color settings for the 3-segment LED rings around the footswitches. With this command, you can specify any color for the LEDs and combine color segments. '0x' is a prefix for a hexadecimal RGB color expression method that represents 65,536 colors. For example, 0xff0000 represents red, 0x00ff00 represents green, 0x0000ff represents blue, 0x000000 represents off, and 0xffffff represents bright white. In fact, you can specify any of the 65,536 colors. Below are some color expression values, and you can fill in your favorite colors in this ledcolorX = [0x ][0x ][0x ] after the 0x.

### Color Value Lookup Table



| | | | | | |
|---|---|---|---|---|---|
| 000000 | 330000 | 660000 | 990000 | CC0000 | FF0000 |
| 003300 | 333300 | 663300 | 993300 | CC3300 | FF3300 |
| 006600 | 336600 | 666600 | 996600 | CC6600 | FF6600 |
| 009900 | 339900 | 669900 | 999900 | CC9900 | FF9900 |
| 00CC00 | 33CC00 | 66CC00 | 99CC00 | CCCC00 | FFCC00 |
| 00FF00 | 33FF00 | 66FF00 | 99FF00 | CCFF00 | FFFF00 |
| 000033 | 330033 | 660033 | 990033 | CC0033 | FF0033 |
| 003333 | 333333 | 663333 | 993333 | CC3333 | FF3333 |
| 006633 | 336633 | 666633 | 996633 | CC6633 | FF6633 |
| 009933 | 339933 | 669933 | 999933 | CC9933 | FF9933 |
| 00CC33 | 33CC33 | 66CC33 | 99CC33 | CCCC33 | FFCC33 |
| 00FF33 | 33FF33 | 66FF33 | 99FF33 | CCFF33 | FFFF33 |
| 000066 | 330066 | 660066 | 990066 | CC0066 | FF0066 |
| 003366 | 333366 | 663366 | 993366 | CC3366 | FF3366 |
| 006666 | 336666 | 666666 | 996666 | CC6666 | FF6666 |
| 009966 | 339966 | 669966 | 999966 | CC9966 | FF9966 |
| 00CC66 | 33CC66 | 66CC66 | 99CC66 | CCCC66 | FFCC66 |
| 00FF66 | 33FF66 | 66FF66 | 99FF66 | CCFF66 | FFFF66 |
| 000099 | 330099 | 660099 | 990099 | CC0099 | FF0099 |
| 003399 | 333399 | 663399 | 993399 | CC3399 | FF3399 |
| 006699 | 336699 | 666699 | 996699 | CC6699 | FF6699 |
| 009999 | 339999 | 669999 | 999999 | CC9999 | FF9999 |
| 00CC99 | 33CC99 | 66CC99 | 99CC99 | CCCC99 | FFCC99 |
| 00FF99 | 33FF99 | 66FF99 | 99FF99 | CCFF99 | FFFF99 |
| 0000CC | 3300CC | 6600CC | 9900CC | CC00CC | FF00CC |
| 0033CC | 3333CC | 6633CC | 9933CC | CC33CC | FF33CC |
| 0066CC | 3366CC | 6666CC | 9966CC | CC66CC | FF66CC |
| 0099CC | 3399CC | 6699CC | 9999CC | CC99CC | FF99CC |
| 00CCCC | 33CCCC | 66CCCC | 99CCCC | CCCCCC | FFCCCC |
| 00FFCC | 33FFCC | 66FFCC | 99FFCC | CCFFCC | FFFFCC |
| 0000FF | 3300FF | 6600FF | 9900FF | CC00FF | FF00FF |
| 0033FF | 3333FF | 6633FF | 9933FF | CC33FF | FF33FF |
| 0066FF | 3366FF | 6666FF | 9966FF | CC66FF | FF66FF |
| 0099FF | 3399FF | 6699FF | 9999FF | CC99FF | FF99FF |
| 00CCFF | 33CCFF | 66CCFF | 99CCFF | CCCCFF | FFCCFF |
| 00FFFF | 33FFFF | 66FFFF | 99FFFF | CCFFFF | FFFFFF |

## short_dwX = [ ][ ][ ][ ] [ ][ ][ ][ ] ...

This item configures the command or combination of commands to be sent upon the Xth key press. The brackets [ ] contain a complete command every 4 items, which are [MIDI Channel Number][Command Type][Value 1][Value 2]. The meanings of all parameters have been explained in detail previously. Here, we look at the example:

short_dw1 = [1][NT][69][127] [1][NT][72][127] [1][NT][76][127] [1][NT][79][127]

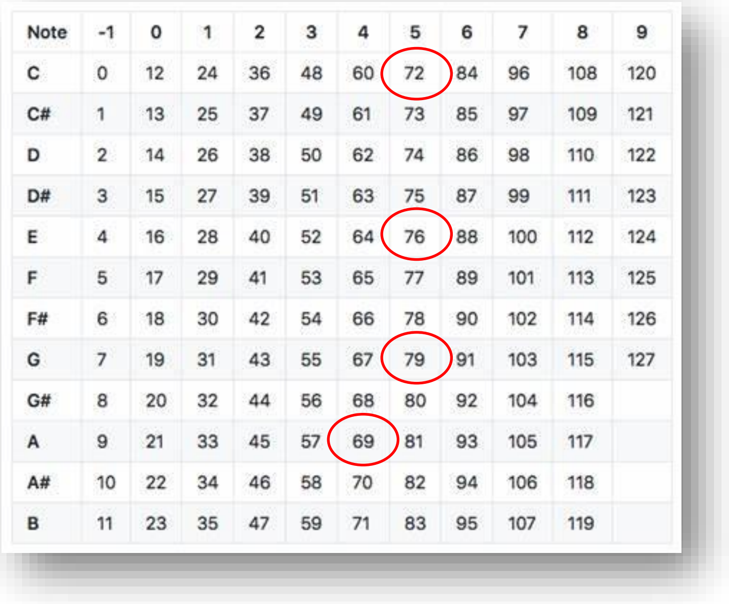
short_dw2 = [-][HID][Ctrl][A]

short_dw3 = [5][PC][10][-]

short_dw4 = [3][PC][random][-]

The configuration means that upon the first key press, four note commands are sent. Based on the note command table, the following notes are sent: A4, C5, E5, G5, which form an Am7 chord. The last parameter of the Note command, which is the value in the fourth [ ], ranges from 0-127 and represents the velocity of the note, with 0 stopping the sound.

## Note Pitch Lookup Table



| Note | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C | 0 | 12 | 24 | 36 | 48 | 60 | 72 | 84 | 96 | 108 | 120 |
| C# | 1 | 13 | 25 | 37 | 49 | 61 | 73 | 85 | 97 | 109 | 121 |
| D | 2 | 14 | 26 | 38 | 50 | 62 | 74 | 86 | 98 | 110 | 122 |
| D# | 3 | 15 | 27 | 39 | 51 | 63 | 75 | 87 | 99 | 111 | 123 |
| E | 4 | 16 | 28 | 40 | 52 | 64 | 76 | 88 | 100 | 112 | 124 |
| F | 5 | 17 | 29 | 41 | 53 | 65 | 77 | 89 | 101 | 113 | 125 |
| F# | 6 | 18 | 30 | 42 | 54 | 66 | 78 | 90 | 102 | 114 | 126 |
| G | 7 | 19 | 31 | 43 | 55 | 67 | 79 | 91 | 103 | 115 | 127 |
| G# | 8 | 20 | 32 | 44 | 56 | 68 | 80 | 92 | 104 | 116 | |
| A | 9 | 21 | 33 | 45 | 57 | 69 | 81 | 93 | 105 | 117 | |
| A# | 10 | 22 | 34 | 46 | 58 | 70 | 82 | 94 | 106 | 118 | |
| B | 11 | 23 | 35 | 47 | 59 | 71 | 83 | 95 | 107 | 119 | |

In the example, the moment the button is pressed for the second time, a keyboard emulation command [-][HID][Ctrl][A] is sent. This is not a MIDI command. Note that mouse and keyboard emulation commands can only be configured in short_dwX. The program does not support sending HID commands in the other three button states to avoid creating overly complex and difficult to understand and remember control actions. This command executes a Ctrl+A combination keystroke, which is the common select all operation.

Upon the third press, short_dw3 = [5][PC][10][-], sends a preset change command on MIDI channel 5 with PC=10.

Pressing short_dw4 for the fourth time = [3][PC][random][-], sends a random PC value on MIDI channel 3.

The fifth press will return to the command defined in the first press short_dw1, thus creating a loop.

### short_upX = [ ][ ][ ][ ] [ ][ ][ ][ ] ...

This set of instructions defines the actions sent by the keys at the moment of short press release, that is, the commands sent at the moment of key release when a long press is not triggered. With the configuration of keytimes = [4], it is also a 4-state cycle, executing the following commands: on the first release, it sends note-off commands to form the muting of an Am7 chord, meaning all note velocity values return to zero; on the second release, it sends the following mixed CC and PC commands: [1][CC][20][5] [2][CC][20][0] [3][CC][80][127] [1][PC][11][-] [1][CC][25][0], such commands are often used for separate control of multiple MIDI chained devices, using MIDI channels to correspond to the specified instruction receiving device, and sometimes before switching PC commands, 1-2 CC commands are used to select the PC's large group range, achieving a preset selection space beyond 0~127. Of course, all these are based on the MIDI input control table of the controlled host device itself. The third and fourth short press releases have no configured commands: short_up3 = [1][-][-][-] short_up4 = [1][-][-][-]; it is also possible to not write these two configurations at all, with the same effect.

The remaining configuration content:

long1 = [1][PC][dec1][-] [1][NT][77][127]

long2 = [1][CC][2][0]

long3 = [1][CC][2][127]

long4 = [1][CC][3][0] [1][CC][3][127]

long_up1 = [1][CC][5][127]

long_up2 = [1][CC][2][20]

long_up3 = [1][PC][inc1][-]

long_up4 = [1][PC][8][-]

These include the commands sent when a long press is triggered and the commands sent when a long press is released, which are essentially no different from the configuration of a short press and release, but the timing of the command sending is different. In this example, the first long press triggers the sending of a PC command [1][PC][dec1][-] and a Note command [1][NT][77][127], with subsequent presses sending CC commands or combinations of multiple commands. By configuring the commands for short press, short press release, and long press release, we can achieve stateful control, meaning that when we hold down the footswitch, certain actions are performed, and once released, it returns to the previous state.

## Key Configuration Function Summary

The above is the complete content of the custom key function configuration, which can be summarized as follows:

- Commands can be configured at four moments: when the footswitch is pressed down, on a short press release, on a long press, and on a long press release. It can be a single command, multiple commands, or even a mix of different commands.

- At the moment of pressing down, it can be configured as a keyboard mouse emulation command, which can be a single key or click action, or a combination of a string of keys and clicks. It can also be mixed with MIDI commands.
- Each command unit is composed of parameters within four square brackets [ ][ ][ ][ ]; the first bracket is for the MIDI channel, the second bracket is for the command type, PC/CC/NT/HID, the third bracket is for command parameter 1, and the fourth bracket is for command parameter 2. PC commands only have parameter 1, CC/NT commands both have two parameters, the first is the number, the second is the value, corresponding to a certain item on the host we want to control and its corresponding parameter. For HID commands, please refer to the content in the next section. PC commands have three special parameters, incX, decX, and random.
- The keys' LEDs have three types of lighting effects: normal, select, and tap.
- The same key can be configured to produce different sets of control functions upon multiple presses, achieving a multi-state cycle.
- All keys' configurations are combined to form a configuration page, and controllers with 4/6/10 switches can have up to 99 configuration pages. Page addition and subtraction are achieved by long-pressing the upper and lower

switches on the far right. On the 1/2 switch versions, each key has 4 MODEs to choose from, which are 4 completely different configurations.

## HID Keyboard and Mouse Simulation Function Configuration

Super Mode on the MIDI Captain series controllers enables keyboard simulation and mouse simulation functions. USB-HID function configuration can be performed at the moment of keypress. We use the following 4 configuration commands to explain the configuration rules and functions. To simplify the explanation, we assume keytimes = [1] and then only configure HID in short_dw1, although HID configuration can also be done in multi-state key cycles.

### short_dw1 = [-][HID][-][A]

The above configuration is for a single key press. The HID command still retains the structure where every four brackets [ ][ ][ ][ ] represent one command unit. The first bracket has no actual meaning, please always fill it with [-], the second bracket is for the command type, here it is fixed as [HID] to differentiate from MIDI commands, the third bracket is reserved for combination keyboard commands, which will be explained in the next example, and the fourth bracket contains the simulated key press, here [A] represents the A key on the keyboard, note that it should be in uppercase. With this command configuration, when we press the footswitch, it is equivalent to a click of the A key on the keyboard.

### HID Command Table

Below are the key values for all currently supported individual keys: note the case sensitivity.

Space (4/6/10 switches) SPACE (1/2 switches)

Esc (4/6/10 switches) ESC (1/2 switches)

Caps

PageUp

PageDown

Up

Down

Left

Right

End

Enter

Table

Home

Ins

Del

PrintS

Pause

BackSpace

F1

F2

F3

F4

F5

F6

F7

F8

F9

F10

F11

F12

1

2

3

4

5

6

7

8

9

0

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

Next, let's look at combination keys, which are composed of multiple function keys and individual keys.

short_dw1 = [-][HID][Ctrl][C]

In this command, the third bracket has been added with a Ctrl configuration, which achieves the effect of a Ctrl+C combination key, which is the common copy operation. In this type of command, the multi-function key values in the third bracket can be as follows: note the case sensitivity.

Ctrl

Shift

Alt

Option

Windows

They correspond to the respective keys on the keyboard. Note that keyboard emulation is fixed set to the standard American keyboard.

short_dw1 = [-][HID][-][Mouse_L]

This is a mouse emulation command that simulates a mouse left-click operation. Currently, mouse emulation commands only support left and right clicks, so the content for the fourth bracket only has the following two options.

Mouse_L

Mouse_R

Mouse emulation cannot currently be used in combination with multifunction keys.

Mouse emulation only simulates mouse button clicks; movements and hovering of the mouse are still performed by the computer's own mouse. The advantage of this is that when we want to use the controller to control the play and pause of a music software or player, we only need to move the mouse to the corresponding button position on the window interface, and then we can achieve the related clicking operations simply by using the footswitch control.

### short_dw1 = [-][HID][Ctrl][A] [-][HID][-][Space] [-][HID][Ctrl][C]...

Just like with MIDI commands, keyboard and mouse emulation can also send string commands. We only need to follow the rule where every four brackets represent one command unit, and sequentially stack the configuration in order. The program will execute the commands from left to right. The command string must not be written with line breaks, and the number of brackets must be a multiple of four. Each command unit must be complete; otherwise, it may cause errors during program execution.

### HID Function Precautions

Caution, when performing keyboard and mouse emulation, please use the Del, BackSpace, and other operations with care, as they can easily delete user and system files.

Keyboard emulation of cursor left, right, up, and down can often be used to perform page-turning actions. We can use these to turn music pages, browse through pictures and photos, and with a USB-OTG cable, these keyboard commands can also be recognized on mobile phones and tablets. Note that on Apple devices, once the keyboard emulation function is connected, it will replace the touchscreen keyboard input. Therefore, when we need to input via touch on the phone, we need to disconnect the controller.

The keyboard and mouse emulation features cannot be enabled when only the power adapter is connected or when a USB cable without data capability is inserted, and there will be a few extra seconds of waiting time during the startup process. On the 1/2 switch versions, the digital tube will display three dots..., and on the 4/6/10 switch versions, there will be no corresponding key prompts on the screen.

# Typical Configuration Examples

## Example 1: Two Buttons for PC Increment and Decrement

This example is for the 4/6/10 switch versions; if it is a 1/2 switch controller version, start configuring from the corresponding button's required [modeX].

Use two buttons to increase and decrease the PC, browsing through the host presets library.

Assuming the use of key0 for increment and key1 for decrement. The LED is set to select mode, displaying red and blue respectively, with the keys having only a single press state:

```
[key0]

keytimes = [1]
```

```
ledmode = [select]

ledcolor1 = [0xff0000][ 0xff0000][ 0xff0000]

short_dw1 = [1][PC][inc1][-]


[key1]

keytimes = [1]

ledmode = [select]

ledcolor1 = [0x0000ff][ 0x0000ff][ 0x0000ff]

short_dw1 = [1][PC][dec1][-]
```

## Example2 CC Toggle Command

Assign a CC toggle command to the key0 button, which sends 127 on one press and 0 on the next, with CC number 26. When sending 127, the LED shows yellow, and when sending 0, it shows a dim yellow.

```
[key0]

keytimes = [2]

ledmode = [normal]

ledcolor1 = [0xffff00][ 0xffff00][ 0x000000]

ledcolor2= [0x808000][ 0x808000][ 0x000000]

short_dw1 = [1][CC][26][127]

short_dw2 = [1][CC][26][0]
```

## Example3 Single Button PC Increment and Decrement

Implement a short press on key0 for PC increment and a long press for PC decrement, enabling presets browsing with a single button, with the indicator light in green.

```
[key0]
keytimes = [1]
ledmode = [normal]
ledcolor1 = [0x000000][ 0x00ff00][ 0x000000]
short_up1 = [1][PC][inc1][-]
long1 = [1][PC][dec1][-]
```

## Example4 Multi-State Button Cycling

For a song that requires switching between 5 tone presets using an effect processor, one button is used to cycle through the tones sequentially, and the indicator light shows segmented feedback to indicate the current number of button presses. The PC values for the 5 tones are 24, 98, 3, 20, 19. In the configuration below, the LED will light up in segments, with the first cycle of 3 segments being red and the second cycle being blue.

```
[key0]
keytimes = [5]
ledmode = [normal]
ledcolor1 = [0xff0000 ][ 0x000000][ 0x000000]
ledcolor2 = [0x000000][0xff0000 ][ 0x000000]
```

ledcolor3 = [0x000000][ 0x000000][0xff0000]

ledcolor4 = [0x0000ff ][ 0x000000][ 0x000000]

ledcolor5 = [0x000000][0x0000ff ][ 0x000000]

short_dw1 = [1][PC][24][-]

short_dw2 = [1][PC][98][-]

short_dw3 = [1][PC][3][-]

short_dw4 = [1][PC][20][-]

short_dw5 = [1][PC][19][-]

## Example5 Multi-commands Multi-host Devices Control

There are two MIDI cascaded devices (a MIDI cable goes from the MIDI OUT of one device to the MIDI IN of another, or from MIDI Through if the Through function is activated)

When key0 is pressed, the first device's PC switches to 10, the second device's PC switches to 6, while CC34, 3 is used to switch to scene 3. When key1 is pressed, the first device's PC switches to 2, the second device's PC remains unchanged, and CC34, 0 is used to switch the scene to 0. Set the MIDI channel of the first device to 1 and the MIDI channel of the second device to 2.

[key0]

keytimes = [1]

ledmode = [normal]

ledcolor1 = [0xffff00][0xffff00][0xffff00]

```
short_dw1 = [1][PC][10][-] [2][PC][6][-] [2][CC][34][3]

[key1]

keytimes = [1]

ledmode = [normal]

ledcolor1 = [0x00ffff][0x00ffff][0x00ffff]

short_dw1 = [1][PC][2][-] [2][CC][34][0]
```

## Example6 Single Key Chord Command Execution

Cycle through 4 chords with a single button press: Am7 – G7 – Fmaj7 – Cmaj7

The following is implemented on mode3 of key0 or key1 in the 1/2 switch version; if it's on the 4/6/10 switch versions, simply place the configuration content under the corresponding button [keyX].

The configuration defines that the sound should stop at the moment of short_upX. If the foot pedal is pressed too long, causing a long press to be triggered, the zero velocity value command in the short_upX instruction will not be reactivated, which is equivalent to continuing the sustain. If we want to immediately stop the sound when releasing a long press, we need to add the long_upX configuration, which is identical to short_upX, both sending the corresponding note's zero velocity command to silence it. In some hosts and software that support the General MIDI protocol, there is a fixed CC command to silence notes, which is CC 123 0, i.e., [1][CC][123][0]. In such cases, this single command can be used to achieve the silencing effect for all notes, which is more convenient.

```
[mode3]
keytimes = [4]
ledmode = [normal]
ledcolor1 = [0xff0000][0x000000][0x000000]
ledcolor2 = [0x000000][0x00ff00][0x000000]
ledcolor3 = [0x000000][0x000000][0x0000ff]
ledcolor4 = [0x00ff00][0x000000][0x000000]
short_dw1 = [1][NT][69][80] [1][NT][72][80] [1][NT][76][80] [1][NT][79][80]
short_dw2 = [1][NT][67][80] [1][NT][71][80] [1][NT][74][80] [1][NT][77][80]
short_dw3 = [1][NT][65][80] [1][NT][69][80] [1][NT][72][80] [1][NT][76][80]
short_dw4 = [1][NT][60][80] [1][NT][64][80] [1][NT][67][80] [1][NT][71][80]
short_up1 = [1][NT][69][0] [1][NT][72][0] [1][NT][76][0] [1][NT][79][0]
short_up2 = [1][NT][67][0] [1][NT][71][0] [1][NT][74][0] [1][NT][77][0]
short_up3 = [1][NT][65][0] [1][NT][69][0] [1][NT][72][0] [1][NT][76][0]
short_up4 = [1][NT][60][0] [1][NT][64][0] [1][NT][67][0] [1][NT][71][0]
```

## Example7 Press and Hold Momentary Control

Configure a button to control the on/off switch of a module in an effect unit, using the value of CC 20. The action required is that when the button is pressed and held, the effect is turned on, and as soon as the foot is released, the effect is turned off.

```
[key1]
keytimes = [1]
```

```
ledmode = [normal]

ledcolor1 = [0x00ffff][0x00ffff][0x00ffff]

short_dw1 = [1][CC][20][127]

short_up1 =[1][CC][20][0]

long_up1 = [1][CC][20][0]
```

Here, both short press release and long press release are set to send the CC value that turns off the module, ensuring that no matter how long the button is held, the effect module will be turned off as soon as the foot is lifted.

## Example8 Keyboard Page Scrolling Simulation

Simulate page up and page down while browsing photos on a computer using the key0 and key1 buttons.

```
[key0]

keytimes = [1]

ledmode = [select]

ledcolor1 = [0xffff00][0xffff00][0xffff00]

short_dw1 = [-][HID][-][Left]

[key1]

keytimes = [1]

ledmode = [select]

ledcolor1 = [0x00ffff][0x00ffff][0x00ffff]

short_dw1 = [-][HID][-][Right]
```

## Example9 Mouse Click Simulation

Simulate a left mouse button click.

```
[key0]

keytimes = [1]

ledmode = [select]

ledcolor1 = [0xffff00][0xffff00][0xffff00]

short_dw1 = [-][HID][-][Mouse_L]
```

# Add PAGEs

The 4/6/10 switch versions can increase the number of different button function combinations by adding pages, with the ability to set a sufficient number of pages, up to 99 pages. Control functions for certain devices can be placed on one page or several adjacent pages for convenient access during use. The device by default only has a few pages set up; to add more pages, we can simply copy and paste an existing page and then rename it in order. For example, if the device previously had up to page8.txt, we can copy and rename it to page9.txt, page10.txt, page11.txt... Then, enter the added page document to make detailed new configurations and function definitions for the buttons.

# Controlling page increment and decrement with external controllers

Controllers with 4/6/10 switch versions can perform page increment and decrement by connecting to an external MIDI controller. This means that in addition to serving as a MIDI Through function, the MIDI IN port can also receive external commands. Currently, the supported external commands are for controlling the page of the device itself. The format for receiving commands is as follows:

### CC 20 xx (where xx is between0and the Max page)

This means the received command is a CC command, with the CC number fixed at 20, and the CC value xx representing the desired page number to jump to. Note: Do not exceed the maximum page number currently set on the controller. For example, if there are only 8 pages made on the device, from page0 to page7, then the range for xx is 0-7. If a CC value greater than 7 is sent, the controller will not respond.

# Follow and Contact Us

Thank you for reading this document. You may need some patience and learning time to become familiar with the free configuration of button functions. Once mastered, we can use the controller to increase convenience in practice and performance, achieve more refined control and possibilities in music performance, create special effects, and enhance musical creativity and expressiveness. The MIDI Captain series products will become your faithful assistant. We will also continuously improve and optimize the functionality and user experience of our products. For questions about configuration and use, please contact us at:support@paintaudio.com orWilson-zuo@paintaudio.com Thank you for your support and purchase! Please stay tuned to PaintAudio and more of our products at www.paintaudio.com .