TEMPI

MAKENOISE

v1.1

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Changes / modifications not approved by the Make Noise Co. could void the user's authority to operate the equipment.

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications.

Limited WARRANTY:

Make Noise warrants this product to be free of defects in materials or construction for a period of one year from the date of purchase (proof of purchase/invoice required).
Malfunction resulting from wrong power supply voltages, backwards or reversed eurorack bus board cable connection, abuse of the product or any other causes determined by Make Noise to be the fault of the user are not covered by this warranty, and normal service rates will apply.

During the warranty period, any defective products will be repaired or replaced, at the option of Make Noise, on a return-to-Make Noise basis with the customer paying the transit cost to Make Noise. Please contact technical@makenoisemusic.com for Return To Manufacturer Authorization.

Make Noise implies and accepts no responsibility for harm to person or apparatus caused through operation of this product.

Please contact technical@makenoisemusic.com with any questions, needs & comments, otherwise...
go MAKE NOISE!

http://www.makenoisemusic.com

THANK YOU:

TEMPI Beta Analysts:	Walker Farrell, James Cigler, Bana Haffar, Robert AA Lowe, Rodent, Devin Booze, Lee Coleman, Alfonso Graceffo, Mike Johnson, and Richard Devine
TEMPI Firmware Engineer:	Matthew Sherwood
TEMPI Designer:	Anthony Rolando
Special Thanks:	CTRLSEL-C, CTRLSEL-V, and CTRLSEL-G

INSTALLATION

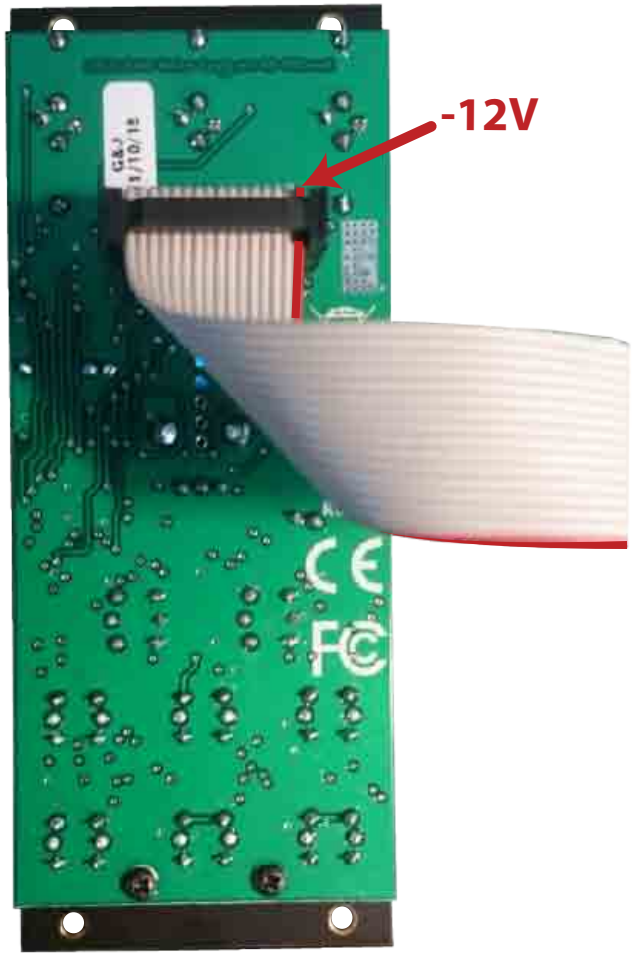
Electrocution hazard!

Always turn the Eurorack case off and unplug the power cord before plugging or un-plugging any Eurorack bus board connection cable cable.
Do not touch any electrical terminals when attaching any Eurorack bus board cable.

The Make Noise TEMPI is an electronic music module requiring 46 mA of +12VDC regulated voltage and a properly formatted distribution receptacle to operate. It is designed to be used within the Eurorack format modular synthesizer system.

Go to http://www.makenoisemusic.com/systems.shtml for examples of Eurorack Systems and Cases.

To install, find necessary space in your Eurorack synthesizer case, confirm proper installation of included eurorack bus board connector cable on backside of module (see picture below), plug the bus board connector cable into the Eurorack style bus board, minding the polarity so that the RED stripe on the cable is oriented to the NEGATIVE 12 Volt line on both the module and the bus board. On the Make Noise 6U or 3U Busboard, the NEGATIVE 12 Volt line is indicated by the white stripe.



Please refer to your case manufacturers' specifications for location of the negative supply.

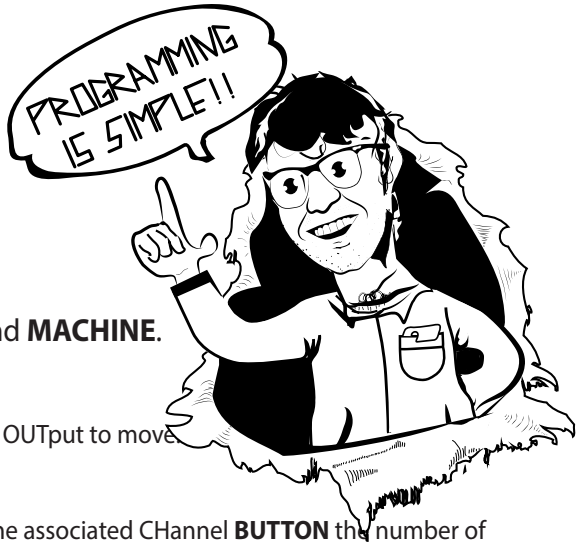
OVERVIEW

TEMPI is deep, but all you really need to know: patch a clock to the **TEMPO** INput and **[TAP]** any **CHannel BUTTON** at a clock rate of your choice. The **<FLASHING>** of the **Channel BUTTON's** LED changes speed to match your **[TAPS]** and a clock of that **TEMPO** is available at the respective CHannel OUTput. Adjusting the six tempi relative to one another will create seemingly infinite rhythmic variations on the theme that is your patch. Much joy may be had without any further knowledge, but I am certain you will want to know more, so read on.

TEMPI is a 6 CHannel, polyphonic, time-shifting module. It provides an intuitive method for the creation and recalling of complex clocking arrangements within a modular synthesizer system.

The primary User Interface and ProGraMming elements for the module are six large, illuminated buttons: **BUTTON-1** through **BUTTON-6**, and two smaller illuminated buttons: **PGM_A** and **PGM_B**. The module is able to store up to sixty-four clock/timing scenarios called **STATES**, arranged in four **BANKS** of sixteen. An LED is used to indicate the current **BANK** by **<COLOR>** and changes in **STATE** indicated by **<FLASHING>**. There are INputs for External **TEMPO**, a Gate INput for **MOD**, and **STATE SELECTION** via a CV INput (with Combo Pot for attenuation) and/or GATE INput.

The primary goal of this module is to have the maximum amount of artist-controlled musical variation with a minimum amount of data input.



There are two methods of ProGraMming TEMPI's CHannels: **HUMAN** and **MACHINE**.

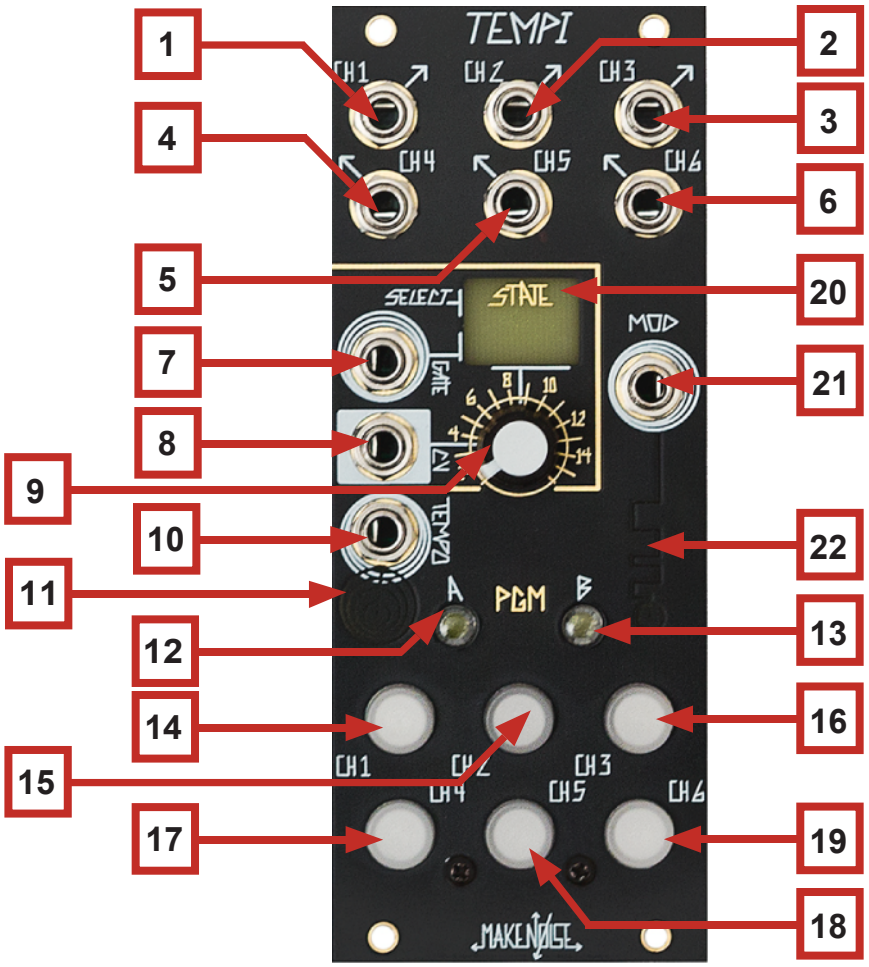
- **HUMAN** ProGraMming is simple:
  Just **[TAP]** the **BUTTON(s)** at the rate you would like the associated CHannel OUTput to move.
- **MACHINE** ProGraMming is simple:
  Just **[HOLD] PGM_A** (for DIVISIONS) or **PGM_B** (for MULTIPLES), and **[TAP]** the associated CHannel **BUTTON** the number of times corresponding to the desired DIVISION (÷) or MULTIPLE (*****).

For example:
To **MACHINE** ProGraM **CHannel-3** to "**÷4**" (divide by four), **[HOLD] PGM_A** + **[TAP] BUTTON-3** four times.

To **MACHINE** ProGraM **CHannel-4** to "***6**" (multiply by six), **[HOLD] PGM_B** + **[TAP] BUTTON-4** six times.

Once a **TEMPO** is established, whether by **HUMAN** or **MACHINE** ProGraMming, it may be fine-tuned by **[HOLDING]** the associated CHannel **BUTTON** and **[TAPPING] PGM_A** to slow it down incrementally, or **PGM_B** to speed it up incrementally.

Note: All six CHannels are synchronized according to the associated CHannel settings and **Leading TEMPO**.
Hint: feel free to skip ahead to Tips and Tricks on Page 27.



TEMPI Panel Controls

1. CHannel 1 OUTput
2. CHannel 2 OUTput
3. CHannel 3 OUTput
4. CHannel 4 OUTput
5. CHannel 5 OUTput
6. CHannel 6 OUTput
7. STATE SELECT GATE INput
8. STATE SELECT CV INput
9. SELECT CV Attenuator/Combo Pot
10. External TEMPO INput
11. TEMPO LED
12. PGM_A BUTTON / LED
13. PGM_B BUTTON/ LED
14. CHannel BUTTON-1 / LED
15. CHannel BUTTON-2 / LED
16. CHannel BUTTON-3 / LED
17. CHannel BUTTON-4 / LED
18. CHannel BUTTON-5 / LED
19. CHannel BUTTON-6 / LED
20. STATE LED
21. MOD GATE INput
22. MOD LED

FACTORY SETTINGS:

When you first power up a new **TEMPI**, **BANK A** will be filled with sixteen States representing a selection of classic and new Clock Divider/Multiplier settings. All these settings are user-editable and can be overwritten at will. Here's a short description:

STATE 1	All six CHannels are "**÷1.**" This setting is also the "Init" for all **STATES** in the other three **BANKS**.

STATES 2-7 are Clock Divider States:

2. Powers of 2:	**÷1, ÷2, ÷4, ÷8, ÷16, ÷32**
3. Primes:	**÷2, ÷3, ÷5, ÷7, ÷11, ÷13**
4. Integers:	**÷1, ÷2, ÷3, ÷4, ÷5, ÷6**
5. Evens:	**÷2, ÷4, ÷6, ÷8, ÷10, ÷12**
6. Odds:	**÷3, ÷5, ÷7, ÷9, ÷11, ÷13**
7. Fibonacci:	**÷2, ÷3, ÷5, ÷8, ÷13, ÷21**

STATES 8-13 are Clock Multiplier States:

8. Powers of 2:	***1, *2, *4, *8, *16, *32**
9. Primes:	***2, *3, *5, *7, *11, *13**
10. Integers:	***1, *2, *3, *4, *5, *6**
11. Evens:	***2, *4, *6, *8, *10, *12**
12. Odds:	***3, *5, *7, *9, *11, *13**
13. Fibonacci:	***2, *3, *5, *8, *13, *21**
14. Multiples and Divisions:	***2, *3, *4, ÷2, ÷3, ÷4**
15. PHASE Demonstration:	All six CHannels are set to the same **TEMPO**, but with six unique **Phase** values.
16. Non-Integer Division Demonstration:	Each successive channel is one Fine decrement slower than the previous, resulting in adjacent CHannels going slowly in and out of **Phase** with each other.

The Factory Settings of the {**ProGraM Edit**} Page are as follows:

- HUMAN Resolution 50%
- SHIFT = CW
- RUN/STOP = OFF
- SHIFT = Jumbled

To restore all Factory Settings, power off the system. [HOLD] PGM_B and BUTTON-4 simultaneously while powering back on.

QUICK REFERENCE:

MACHINE ProGraMming

VARIABLE CLOCK MULTIPLIER / DIVISOR

- **[HOLD] PGM_A + [PRESS]** CHannel **BUTTON1-6** *n* times for **VARIABLE CLOCK DIVISOR Coarse** adjustment (1÷*n*)
- **[HOLD] PGM_B + [PRESS]** CHannel **BUTTON1-6** *n* times for **VARIABLE CLOCK MULTIPLIER Coarse** adjustment (1**n*)
- **[HOLD]** CHannel **BUTTON1-6 + [PRESS] PGM_A** for **VARIABLE CLOCK FINE Decrement**
- **[HOLD]** CHannel **BUTTON1-6 + [PRESS] PGM_B** for **VARIABLE CLOCK FINE Increment**

{PHASE}

[PRESS] PGM_A + **PGM_B; PGM_A + PGM_B = <ON>**

- **[HOLD]** CHannel **BUTTON1-6:**
  - +**[PRESS] PGM_A** *n* times to **PHASE Fine Decrement MULTIPLIER/DIVISOR** value
  - +**[PRESS] PGM_B** *n* times to **PHASE Fine Increment MULTIPLIER/DIVISOR** value
- **[HOLD] PGM_A** + **[TAP]** CHannel **BUTTON1-6** to **PHASE Coarse Decrement MULTIPLIER/DIVISOR**
- **[HOLD] PGM_B** + **[TAP]** CHannel **BUTTON1-6** to **PHASE Coarse Increment MULTIPLIER/DIVISOR**

HUMAN ProGraMming

[TAP] BUTTON1-6 at least twice

- **HUMAN** ProGraMming of **MULTIPLE** and **DIVISIONS** of **Leading TEMPO**

{MUTE}

[PRESS] PGM_A ; PGM_A= <ON>

- **[PRESS]** CHannel **BUTTON1-6** to **MUTE** CHannels**;** **MUTEd** CHannels **= <RED>**

{MOD}

[PRESS] PGM_B ; PGM_B = <ON>

- **[PRESS]** CHannel **BUTTON1-6** to **MOD** CHannels**;** **MODded** CHannels **= <PURPLE>**

{STATE EDIT}

[HOLD] both **PGM_A** + **PGM_B**:

- **[PRESS] BUTTON-1** to **RECALL** current **STATE**
- **[PRESS] BUTTON-2** to **STORE** current **STATE**
- **[PRESS] BUTTON-3** to **RECALL** current **BANK**
- **[PRESS] BUTTON-4** to **COPY** selected **STATE**
- **[PRESS] BUTTON-5** to **PASTE** over selected **STATE**
- **[PRESS] BUTTON-6** to **MUTATE** over selected **STATE**

QUICK REFERENCE: (CONT'D)

{BANK EDIT}
[DOUBLE PRESS] PGM_A;
PGM_A = <FLASHING>

- **[PRESS] BUTTON-1** to **SELECT BANK**
- **[PRESS] BUTTON-2** to **STORE ALL BANKS** and current **TEMPO**
- **[PRESS] BUTTON-3** to **FOLLOW SELECT & TEMPO Busses**
- **[PRESS] BUTTON-4** to **COPY** selected **BANK**
- **[PRESS] BUTTON-5** to **PASTE** over selected **BANK**
- **[PRESS] BUTTON-6** to **MUTATE** over selected **BANK**
- **[PRESS] PGM_A** to Exit **{BANK EDIT}**

{ProGraM EDIT}
[DOUBLE PRESS] PGM_B;
PGM_B = <FLASHING>

MOD SETTINGS

- **[PRESS] BUTTON-1** for **HUMAN Resolution 100%**
- **[PRESS] BUTTON-2** for **HUMAN Resolution 50%**
- **[PRESS] BUTTON-3** for **HUMAN Resolution 25%**
- **[PRESS] BUTTON-4** for **SHIFT:**
  - **CW = <RED>**
  - **CCW = <BLUE>**
  - **RANDOM = <PURPLE>**
  - **OFF = BUTTON-5** selected
- **[PRESS] BUTTON-5** for **RUN/STOP:**
  - **RUN/STOP = <RED>**
  - **RUN/STOP ALL = <BLUE>**
  - **ALT RUN = <PURPLE>**
  - **OFF = BUTTON-4** is **[PRESSED]**
- **[PRESS] BUTTON-6**
  When **SHIFT** is active, sets behavior of **SHIFT**: **NOT Jumbled SHIFT = <OFF>** ; **Jumbled SHIFT = <RED>**
  When **RUN/STOP** is active, sets behavior of the **MOD** INput: **Momentary** = **<RED>**; **Toggled = <OFF>**

Leading TEMPO

There are three ways to determine the **Leading TEMPO**, which synchronize the six, **Variable Clock** OUTputs of the **TEMPI**:

1. With nothing patched to the **TEMPO** INput, the last-**STOREd TEMPO** is used (refer to **{BANK EDIT}** page **STORE,** described on page 23).

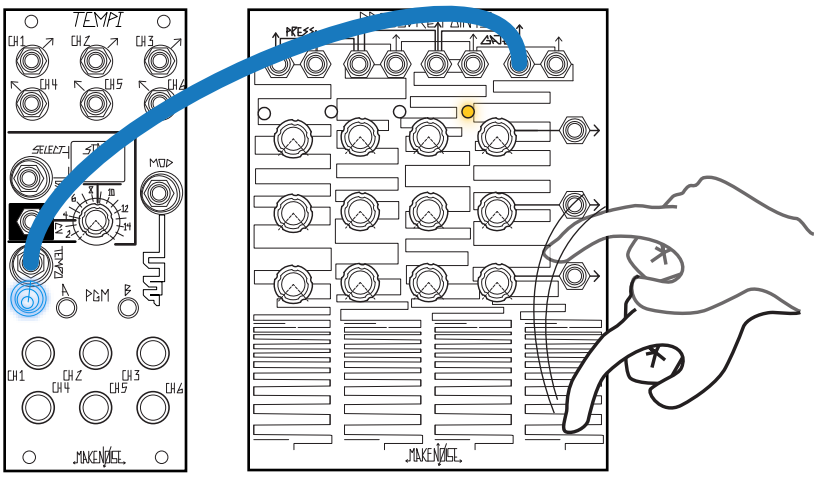
2. When an **External Clock** is patched to the **TEMPO IN**put, the **External Clock** source is followed.

3. When an internally-connected bus line is used via the **TEMPO BUS**, the generated Clock signal is **FOLLOWed** (refer to **{BANK EDIT}** page **FOLLOW**, described on page 25).

Note: While externally clocking, when changing TEMPO from fast to slow, there will be some delay, as the TEMPI needs a minimum of 2 Clock pulses to measure the incoming Clock rate. For a stable CLock, once the Leading TEMPO is learned, unpatch the Clock from the external TEMPO INput. As such, to create clock jitter, pre-process your external Clock before it is patched to the External TEMPO INput. Even as the External Clock rate changes, the TEMPI's Variable Clock OUTputs will remain synchronized as long as the External Clock is still patched to the TEMPO INput.

It is possible to **[TAP]** the **TEMPO** by patching a GATE OUTput, for example, from the **Pressure Points**, to **TEMPI's TEMPO** INput and **[TAPPING]** the desired **TEMPO** on the **Pressure Points**. Remember: at least two **[TAPS]** are required. The **TEMPI** will lock on to the new **TEMPO** and **FOLLOW** until a new **TEMPO** is **[TAPPED]**. **TEMPO** is indicated by the **<BLUE> TEMPO LED** that **<FLASHES>** just below the **TEMPO** INput jack.

To ProGraM **Leading TEMPO** via **[TAPPING]**:



Variable Clock ProGraMming:

The **Variable Clock** OUTputs may be **MACHINE** ProGraMmed by **[PRESSING]** and **[HOLDING]** some combination of **PGM_A** or **PGM_B** and/or **BUTTON(s)1-6** for the associated **CHannel** OUTputs.

The **Variable Clock** OUTputs are **MULTIPLES** or **DIVISIONS** of the **Leading TEMPO**. When **Variable Clock** OUTs are active and **MUTE disabled** and/or **MOD disabled**, the associated CHannel **BUTTON LED(s)** are **<BLUE>**, **<FLASHING>** to indicate Clock High/Low.

NOTE: Pulse Width of the **Variable Clock** OUTputs is always 50% Duty Cycle.

There are two ways to ProGraM **Variable Clocks**: **MACHINE** and **HUMAN**.

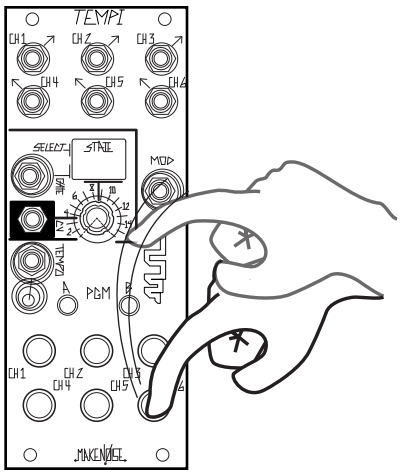
HUMAN ProGraMming

HUMAN Clock ProGraMming combines **MULTIPLIER/DIVISOR** and **PHASE** settings into one simple, tactile process. For **HUMAN** ProGraMming, patch the **Leading TEMPO** (i.e. MULT of the clock patched to **TEMPO IN**put or any CHannel **TEMPI** set to 1:1 using **MACHINE MULTIPLIER or DIVISOR**) so that the timing is audible, and ProGraM a **Variable Clock** by **[TAPPING]** the **BUTTON(s)1-6** for the associated **CHannel OUT**put(s) in order to complement the **Leading TEMPO**, resulting in either a MULTIPLE or DIVISION of the **Leading TEMPO**. **Notice, the User must [TAP] BUTTON at least twice in order to ProGraM a timing value.** Changes happen as soon as algorithmically possible. Previous values for a given CHannel are overwritten as soon as associated **BUTTON1-6** is **[TAPPED]** the second time. While **HUMAN** ProGraMming, **STATE** changes are ignored.

It is also possible to ProGraM several CHannels at once. This makes it more intuitive to create interesting timing arrangements.

IMPORTANT: these changes ARE NOT STOREd UNTIL you STORE the STATE. Without running the STORE operation, changes ARE NOT held when the power is cycled. This makes it easy to improvise with alternate versions of a STOREd theme.

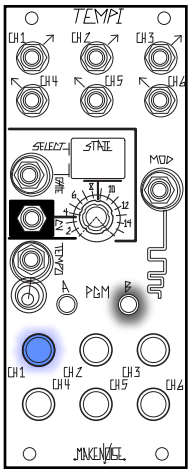
For **HUMAN** ProGraMming:



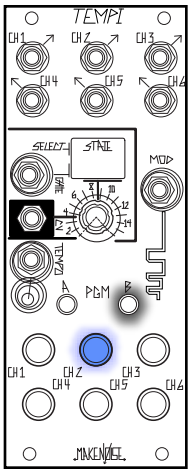
[TAP] Associated BUTTON(s)1-6

RESOLUTION

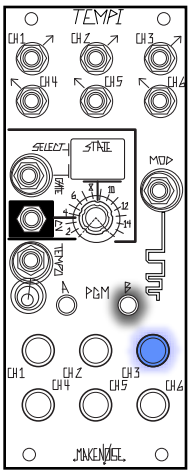
HUMAN ProGraMming has three levels of **Resolution**, determined by the ProGraMming of **{ProGraM Edit}** Page (to access, **[DOUBLE PRESS] PGM_B = PGM_B <FLASHES>**). The **Resolution** of **HUMAN** ProGraMming determines how the **Variable Clocks** are related to the **Leading TEMPO's PHASE and MULTIPLIER/DIVISOR**. The **Resolution** setting is indicated by **ONE** of these three **BUTTON1-3 LEDs = <ON>** while in **{ProGraM Edit}**:



HUMAN Resolution 100%
Results in the strictest relationships. This is the same as **MACHINE** ProGraMming **Coarse MULTIPLIER/DIVISOR** and **PHASE adjustments** (see "**MACHINE**" section).



HUMAN Resolution 50%
The Default setting, allowing for something less strict than **100%**, but easier to use than **25%.**



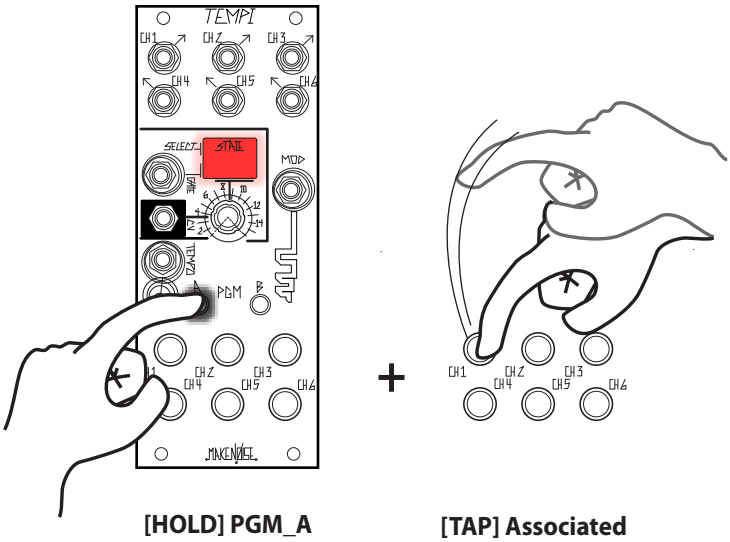
HUMAN Resolution 25%
Establishes in the most free relationships and in theory is the equivalent as **MACHINE** ProGraMming **Fine MULTIPLIER/DIVISOR** and **PHASE adjustments**.

MACHINE ProGraMming

There are two types of **MACHINE** ProGraMming: **Variable Clock MULTIPLIER/DIVISOR** and **PHASE**.

Variable Clock MULTIPLIER/DIVISOR ProGraMming requires a combination of **[HOLDS]** and [**PRESSES]** in order to adjust the associated **Variable Clock** OUTputs:

For **Variable Clock DIVISOR Coarse adjustment**:



[HOLD] PGM_A

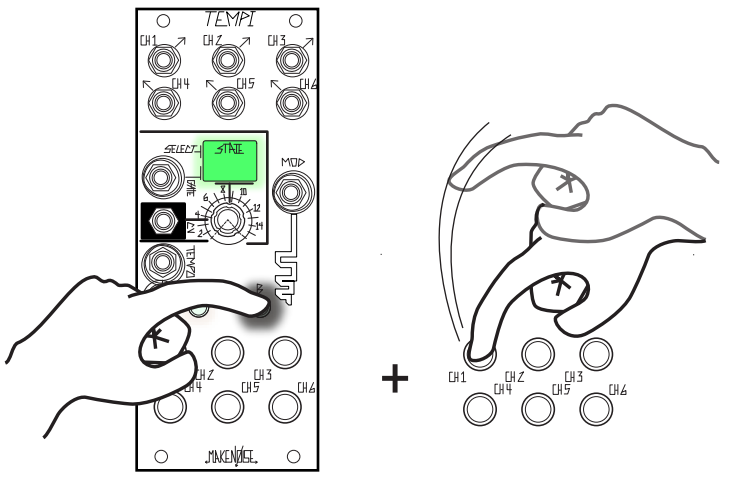
[TAP] Associated CHannel Button(s) 1-32 times

The user **[HOLDS] PGM_A** and **[TAPS]** the desired CHannel **BUTTON(s)** one to thirty-two times in order to ProGraM the associated Variable Clock OUTput(s). **NOTE: it's possible to [TAP] BUTTON(s)1-6 simultaneously, in order to ProGraM several CHannels at once.**

The **TEMPI** counts the **[TAPS]** and sets this number as the **DIVISOR** value. If thirty-two **[TAPS]** are exceeded, the count stays at maximum number of thirty-two-- the counting of subsequent **[TAPS]** is stops with changes take effect when **PGM_A** is **[RELEASED]**. The **STATE LED <LIGHTS RED>** to indicate a **Variable Clock DIVISOR Coarse adjustment.** Previous values for a given CHannel are overwritten as soon as associated **BUTTON(s)1-6** is **[PRESSED]** while **[HOLDING] PGM_A**.

To return to a 1:1 relationship with **TEMPO** (Default), **[HOLD] PGM_A** or **PGM_B** and **[PRESS]** associated CHannel **BUTTON1-6** once.

For **Variable Clock MULTIPLIER Coarse adjustment**:



[HOLD] PGM_B

[TAP] Associated CHannel Button(s) 1-32 times

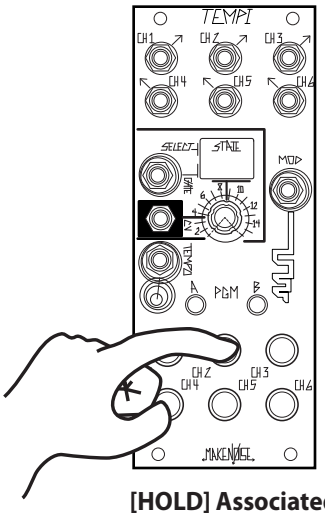
The user **[HOLDS] PGM_B** and **[TAPS]** the desired CHannel **BUTTON(s)** one to thirty-two times in order to ProGraM the associated Variable Clock OUTput(s). **NOTE: it's possible to [TAP] BUTTON(s)1-6 simultaneously, in order to ProGraM several CHannels at once.**

The **TEMPI** counts the number of **[TAPS]** and sets this as the **MULTIPLIER** value. If thirty-two **[TAPS]** are exceeded, the count stays at maximum number of thirty-two-- the counting of subsequent **[TAPS]** is stopped with changes taking effect when **PGM_B** is **[RELEASED]**. The **STATE LED <LIGHTS GREEN>** to indicate a **Variable Clock MULTIPLIER Coarse adjustment.** Previous values for a given CHannel are overwritten as soon as the associated **BUTTON(s)1-6** is **[PRESSED]** while **[HOLDING] PGM_B**.

To return to a 1:1 relationship with **TEMPO** (Default), **[HOLD] PGM_A** or **PGM_B** and **[PRESS]** associated CHannel **BUTTON1-6** once.

MACHINE ProGraMming MULTIPLIER/DIVISOR: (cont'd)

For **Variable Clock MULTIPLIER/DIVISOR Fine adjustment:**

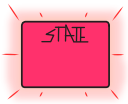


[HOLD] Associated BUTTON(s)1-6

\+

A

[PRESS] PGM_A to Variable Clock Fine decrement

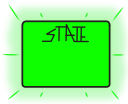


For **Variable Clock Fine decrement (**i.e. to make slower), the user **[HOLDS]** associated CHannel **BUTTON1-6** and **[PRESSES] PGM_A**. The **STATE LED <FLASHES RED>** to indicate a **Variable Clock Fine decrement** has been made.

or

B

[PRESS] PGM_B to Variable Clock Fine increment



For **Variable Clock Fine increment** (i.e. to make faster), the user **[HOLDS]** associated CHannel **BUTTON1-6** and **[PRESSES] PGM_B**. The **STATE LED <FLASHES GREEN>** to indicate a **Variable Clock Fine increment** has been made.

You may **Fine increment** or **decrement** across the entire possible **Variable Clock** range, but you may find it is much easier to make a **Coarse adjustment** and then make **Fine adjustments** in order to find unique timings.

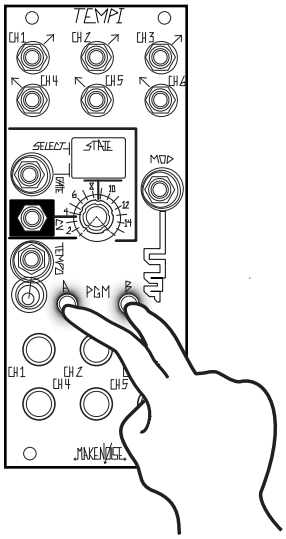
< END OF SECTION >

MACHINE ProGraMming {PHASE}:

The other way to **MACHINE** ProGraM is by **PHASE adjustments. To enter {PHASE} ProGraMming Page, [PRESS]** both **PGM_A** and **PGM_B** at the same time. While in the **{PHASE}** page, both **PGM_LEDs** are **<ON>** and **STATE** changes and **HUMAN** ProGraMming are ignored. **[PRESS]** both **PGM_A** and **PGM_B** to exit **{PHASE}** ProGraMming Page.

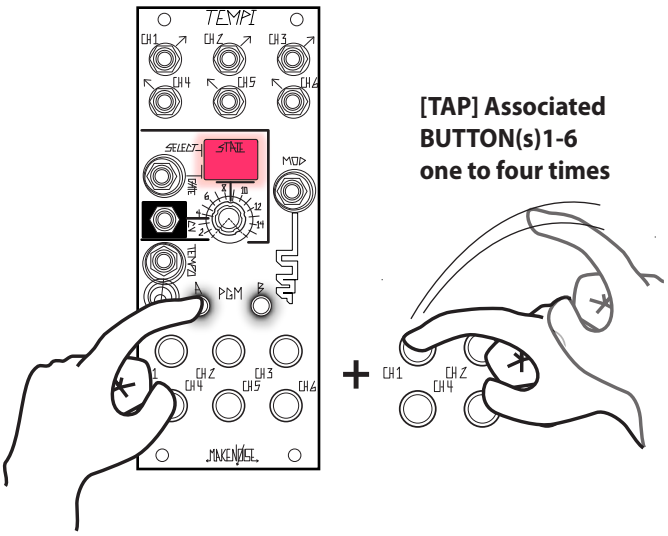
Next, **PHASE adjustments** are performed in a similar way to the **MULTIPLIER/DIVISOR** settings, using **Coarse** and **Fine PHASE decrements/increments.**

To begin **{PHASE}** ProGraMming,
[PRESS] PGM_A and PGM_B



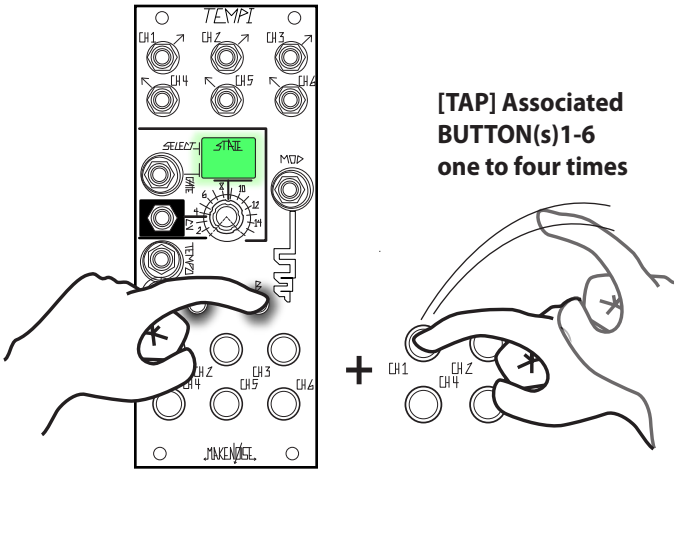
Once in **{PHASE}** ProGraMming Page:

For **Coarse PHASE decrement:**
[HOLD] PGM_A



The user **[HOLDS] PGM_A** and **[TAPS]** CHannel **BUTTON(s)1-6**, one to four times in order to ProGraM the associated CHannel OUTput(s). **Coarse PHASE adjustments** offsets the **PHASE** of the CHannel's OUTput by one cycle of the **Leading TEMPO**. The **STATE LED <LIGHTS RED>** to indicate a **Coarse PHASE decrement** has been made.

For **Coarse PHASE increment:**
[HOLD] PGM_B



The user **[HOLDS] PGM_B** and **[TAPS]** CHannel **BUTTON(s)1-6**, one to four times in order to ProGraM the associated CHannel OUTput(s). **Coarse PHASE adjustments** offset the **PHASE** of the CHannel's OUTput by one cycle of the **Leading TEMPO**. The **STATE LED <LIGHTS GREEN>** to indicate a **Coarse PHASE increment** has been made.

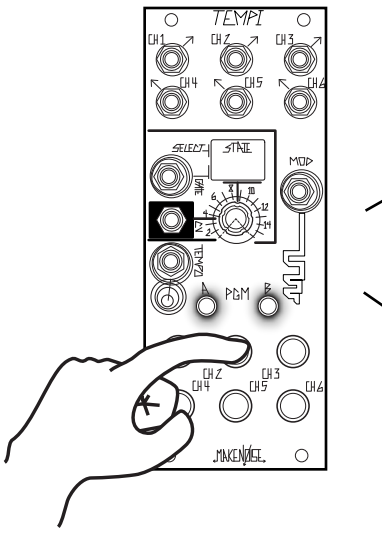
When programming Multiples of the TEMPO, Coarse PHASE adjustments can be very subtle. Here is why: if the CHannel is set to an integer Multiple (e.g. *** 4**), **Coarse PHASE** will have no audible effect, because the CHannel's Clocks are output at the same times at each pulse of the **TEMPO**. If the CHannel is set to a non-integer Multiple (e.g. * **4.25**), the CHannel's clocks will drift in and out of **PHASE** with the **TEMPO** over a period of several pulses of the **TEMPO**. Coarse **PHASE** of a Multiple in this situation will only affect the zero-center of this **PHASE**-shifting: a subtle effect indeed.

MACHINE ProGraMming {PHASE}: (cont'd)

While still in **{PHASE}** ProGraMming Page:

For **Fine PHASE adjustment:**

[HOLD] Associated BUTTON(s)1-6

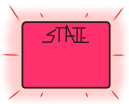


\+

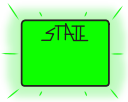
[PRESS] PGM_A to Fine PHASE decrement (i.e. lead AHEAD of **TEMPO**)

or

[PRESS] PGM_B to Fine PHASE increment (i.e. lag BEHIND **TEMPO**)



For **FINE PHASE decrement**, the user **[HOLDS]** the associated CHannel **BUTTON(s)1-6** and **[PRESSES] PGM_A**. The **STATE LED <FLASHES RED>** to indicate a **FINE PHASE decrement** has been made.



For **FINE PHASE increment**, the user **[HOLDS]** the associated CHannel **BUTTON(s)1-6** and **[PRESSES] PGM_B.** The **STATE LED <FLASHES GREEN>** to indicate a **FINE PHASE increment** has been made.

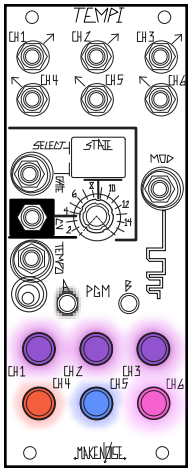
NOTE: You may FINE PHASE increment/decrement across the entire possible PHASE range. For CHannels ProGraMmed to be **DIVISIONS**, **Fine PHASE adjustments** offsets the **PHASE** of the CHannel's OUTput by one quarter of a cycle of the **Leading TEMPO**. Likewise, for CHannels ProGraMmed to be MULTIPLES, it offsets the **PHASE** of the CHannel's OUTput by one quarter of the selected MULTIPLE of the **Leading TEMPO**.

Coarse PHASE adjustments OVERRIDE Fine PHASE adjustments. In this way, **Coarse PHASE adjustments** return to INTEGRAL DIVISORS and MULTIPLIERS, for instance, 1:1.

{MUTE}

To **MUTE** a CHannel output, **[PRESS] PGM_A** to toggle **{MUTE}** ProGraM page to "ON." In order to exit the **{MUTE}** ProGraM page, **[PRESS] PGM_A**. The associated **PGM_A LED <ON>** indicates that the **{MUTE}** ProGraM page is active. While in **{MUTE}** page, **STATE** changes are ignored. The User then **[PRESSES]** the associated CHannel **BUTTON(s)1-6** in order to toggle the **Variable Clock** OUTput(s) to be **MUTE enabled** or **MUTE disabled**. When toggled to be **MUTE enabled,** the associated **BUTTON1-6 LED** is **<RED>** and **<FLASHES>** with each Rising Edge of each Clock in order to indicate the currently ProGraMmed **Variable Clock**. If toggled to be **MUTE disabled**, the associated CHannel **BUTTON1-6** is **<BLUE>**. If a CHannel is **MOD enabled** and **MUTE disabled**, the associated CHannel **BUTTON1-6** is **<PURPLE>**. Alternatively, if a CHannel is BOTH **MOD enabled** and **MUTE enabled** (see below), the associated CHannel **BUTTON1-6 LED(s)** is **<PINK>**.

The purpose of **{MUTE}** is to stop the Clock activity at the associated CHannel OUT(s) while still allowing the user to ProGraM desired changes. In this way, it is possible to ProGraM a new **Variable Clock** into a **MUTE enabled** CHannel. This is useful for cueing changes during live performance. Exit the **{MUTE}** page by **[PRESSING] PGM_A** in order to ProGraM changes to the **MUTED** CHannel by **MACHINE** or **HUMAN** ProGraMming.



MUTE enabled, MOD disabled = **<RED>**

MUTE disabled, MOD disabled = **<BLUE>**

MUTE disabled, MOD enabled = **<PURPLE>**

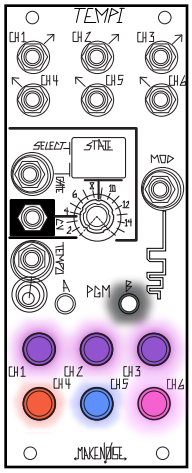
MUTE enabled and MOD enabled = **<PINK>**

NOTE: It is possible to {MUTE} ProGraM several CHannels at once. This makes it more intuitive to create interesting timing arrangements.

Please note: these changes ARE NOT STOREd UNTIL you STORE the STATE. Without running the STORE operation, changes ARE NOT held when the power is cycled. This is useful for improvising with alternate versions of a STOREd theme.

{MOD}

The **MOD** function allows you to ProGraM **SHIFT** or **RUN/STOP** behavior. To ProGraM the **MOD** function, **[PRESS] PGM_B** to toggle to **{MOD}** ProGraM page, indicated by the **PGM_B LED <ON>**. While in **{MOD}** page, STATE changes are ignored. In order to exit the **{MOD}** ProGraM page, **[PRESS] PGM_B** again. In order to toggle associated **Variable CLocK OUT**put(s) to be in or out of the **MOD** function, **[PRESS]** the associated CHannel **BUTTON(s)1-6**. CHannels in the **MOD** function are **<PURPLE>** and **<FLASH>** to indicate the currently ProGraMmed **Variable Clock**. If a CHannel is BOTH **MOD** and **MUTE enabled**, the associated CHannel LED appears **<PINK>**. If a CHannel is **MOD disabled**, it is either **<BLUE>** or **<RED>**. ALL CHannels will continue to **<FLASH>** to indicate currently-ProGraMmed **Variable Clock**.



MOD disabled, MUTE enabled = **<RED>**

MOD disabled, MUTE disabled = **<BLUE>**

MOD enabled, MUTE disabled = **<PURPLE>**

MOD enabled and **MUTE enabled** = **<PINK>**

{ProGraM EDIT}
[DOUBLE PRESS] PGM_B:

MOD SETTINGS

[PRESS] BUTTON4 for **SHIFT:**

CW	=	**<RED>**
CCW	=	**<BLUE>**
RANDOM	=	**<PURPLE>**
OFF	=	**BUTTON-5** selected

[PRESS] BUTTON5 for **RUN/STOP**

RUN/STOP	=	**<RED>**
RUN/STOP ALL	=	**<BLUE>**
ALT RUN	=	**<PURPLE>**
OFF	=	**BUTTON-4** selected

[PRESS] BUTTON-6

When **SHIFT** is active (i.e. **BUTTON-4 = <ON>**), sets **Jumbled SHIFT ON/OFF**; **<RED> = OFF**

When **RUN/STOP** is active (i.e. **BUTTON5 = <ON>**), sets behavior of the **MOD IN**put jack to be **Momentary** or **Toggled**; **<RED> = OFF**

While **HUMAN** or **MACHINE** ProGaMming, **MOD** remains **disabled** until ProGaMming is completed. This behavior may be used creatively in order to manually "freeze" the **MOD** functionality. In order to easily create interesting timing arrangements, it is possible to **MOD** ProGraM several CHannels at once.

The **MOD** function is determined by the ProGraMming of the **MOD SETTINGS** in the **{ProGram EDIT}** page. The **MOD IN**put affects the **TEMPI** according to the behavior specified. The two behaviors are **SHIFT**, which causes timing information to move between **MOD-enabled** CHannels at each new Gate, and **RUN/STOP**, which can cause some or all CHannels to **RUN**, **STOP**, and/or **RESET**. The **MOD LED** will **<FLASH MAGENTA>** to indicate a **MOD** function has been performed.

Please note: these changes ARE NOT STOREd UNTIL you STORE the STATE. Without running the STORE operation, changes ARE NOT held when the power is cycled. This is useful for improvising with alternate versions of a STOREd theme.

{ProGraM EDIT}: SHIFT

The **SHIFT** FUNction trades the values for the **Variable Clock** OUTputs and is accessed from the **{ProGraM EDIT}** page (**[DOUBLE PRESS] PGM_B = PGM_B <FLASHES>**). For example, if CHannels 2, 3, and 5 are set to be **SHIFT**ed, on the Rising Edge of each Gate in the **MOD IN**put, the value of **Variable Clock** 2 is swapped with that of **Variable Clock** 5, while simultaneously replacing the value of **Variable Clock** 3 with that of the previous **Variable Clock** 2, allowing CHannel 5 to take on the original value of CHannel 3…. In other words, it behaves as a **Shift Register** for the **Variable Clock** parameter values, where at each **SHIFT** command received, the values are passed to the next active CHannel within the **{MOD}** Page. These changes always occur at the Rising Edge of the signal patched to the **MOD** INput so that when a Gate "HIGH" is read, the **MOD LED <FLASHES MAGENTA>** and the **SHIFT** of parameter values occurs immediately, thus allowing for self-patching the **TEMPI**.

Using **BUTTON-6** in **{ProGraM EDIT}**, **SHIFT** may be ProGraMmed to operate in the following two ways when the user navigates away from and back to a State:

- **Not-Jumbled SHIFT** is the Default and goes back to the last **STORE**d settings;
  BUTTON-6 LED = <OFF>
- **Jumbled SHIFT** returns to the values last utilized at that **STATE**;
  BUTTON-6 LED = <RED>

Disengage **JUMBLED SHIFT** when you want to easily return to a **STATE's** ProGraMmed **Variable Clocks** even after having **SHIFTed** any number of CHannels.**Note: MUTEs are not SHIFTed. One could emulate a SHIFTed MUTE by ProGraMming a large Division and SHIFTing.**

Tip: SHIFT is also useful for ProGraMming variations of **STATEs** where you want use the same **TEMPI** CHannel values, but on different CHannels:

1. Create your initial **STATE**.
2. **COPY** and then **PASTE** to another **STATE**.
3. **[PRESS] PGM_B** to enter **{MOD}** ProGraMming Page and set two CHannels to be **SHIFT Enabled** (**BUTTON1-6 = <Purple>**)
4. Send a Gate signal to the **MOD** INput to swap.
5. Set these two CHannels back to being **MOD disabled**.

Use this technique to trade CHannel values and create new **STATEs**.

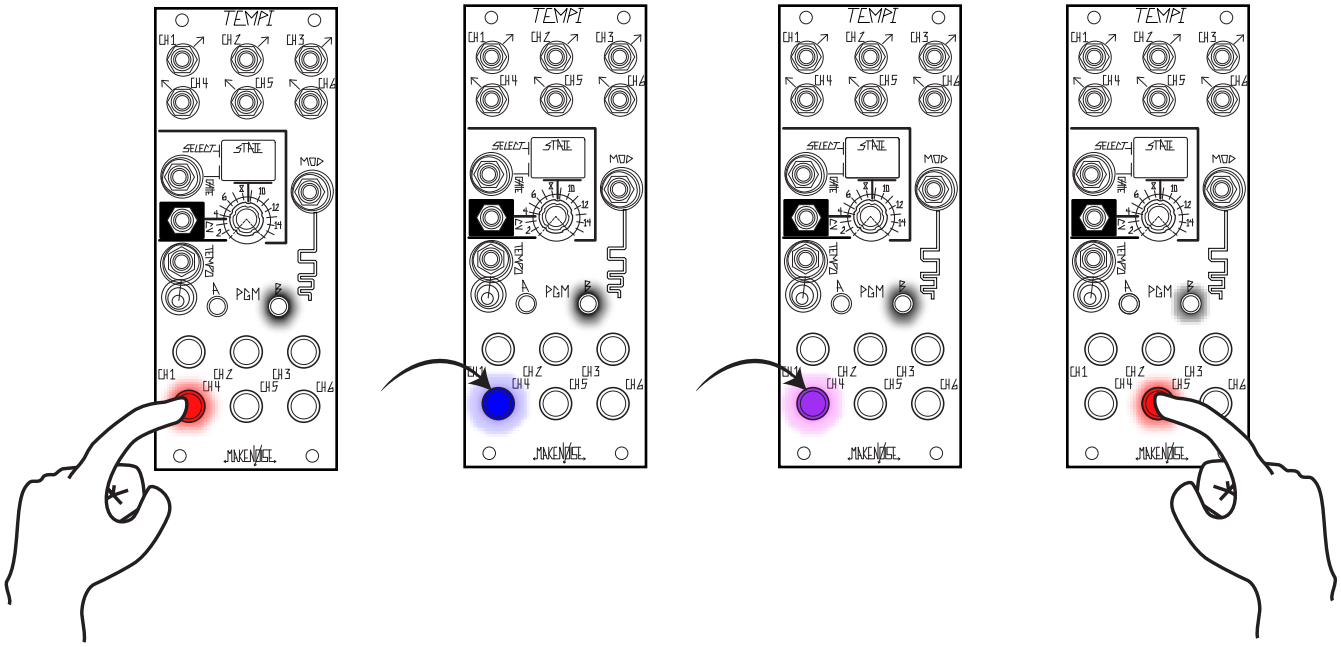
Using **BUTTON-4** in **{ProGraM EDIT}**, **SHIFT** may be ProGraMmed to operate in the following directions:

Clockwise, CW
BUTTON-4 = <Red>

Counterclockwise, CCW
BUTTON-4 = <Blue>

Random (aka RODENT Mode)
BUTTON-4 = <Purple>

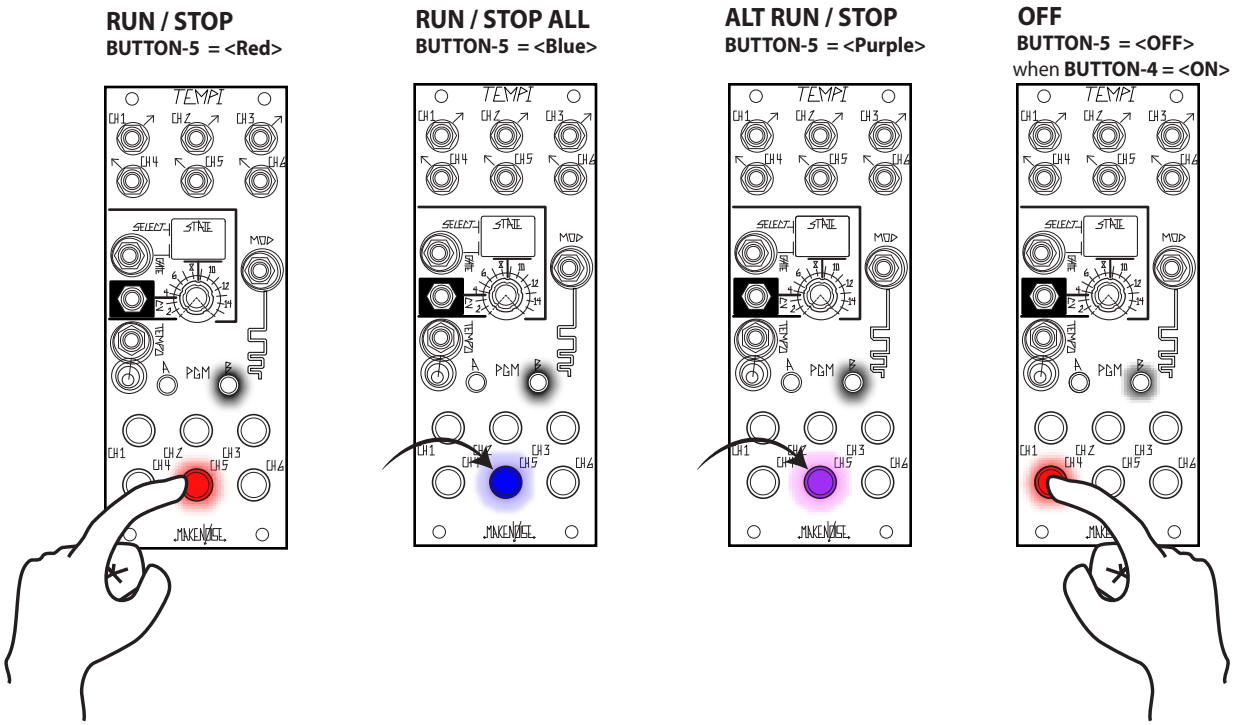
OFF
BUTTON-4 = <OFF>
when **BUTTON-5** is **[ON]**



{ProGraM EDIT}: RUN / STOP

The **RUN / STOP** FUNction allows for resting and/or resetting the **Variable Clock** OUTputs. When **RUN**ning, the **Variable Clock** OUTputs are **ON** and operate as ProGraMmed with a 50% Duty Cycle Clock signal. When **STOPped**, the **Variable Clock** OUTputs are resting at 0V, or **OFF**.

There are several types of **RUN/STOP** behaviors to choose from that are ProGraMmed using **BUTTON-5** in the **{ProGraM EDIT}** Page (**[DOUBLE PRESS] PGM_B = PGM_B <FLASHES>**):



RUN / STOP **BUTTON-5 = <RED>**

Makes CHannels ProGraMmed for **MOD = <Purple>**, **RUN** on **MOD <ON>**, and **STOP** on **MOD <OFF>**. CHannels not ProGraMmed for **MOD** (i.e. **<BLUE>** or **<RED>**) are not affected. The **MOD LED** indicates if the **MOD** is **<ON>** or **<OFF>**.

RUN / STOP ALL (aka James Cigler Mode) **BUTTON-5 = <BLUE>**

Makes all CHannels RUN on **MOD <ON>** and **STOP** on **MOD <OFF>**, reseting all CHannels on **RUN**. The **MOD LED** indicates if the **MOD** is **<ON>** or **<OFF>**. This behavior is good for creating compositional structures where you would like a Beginning or End. It is also nice when used with a MIDI to CV module to convert MIDI Note On/Note Off messages to Gate Signals that can be used to drive the **TEMPI**.

ALT. RUN / STOP (aka Robert A.A. Lowe Mode) **BUTTON-5= <PURPLE>**

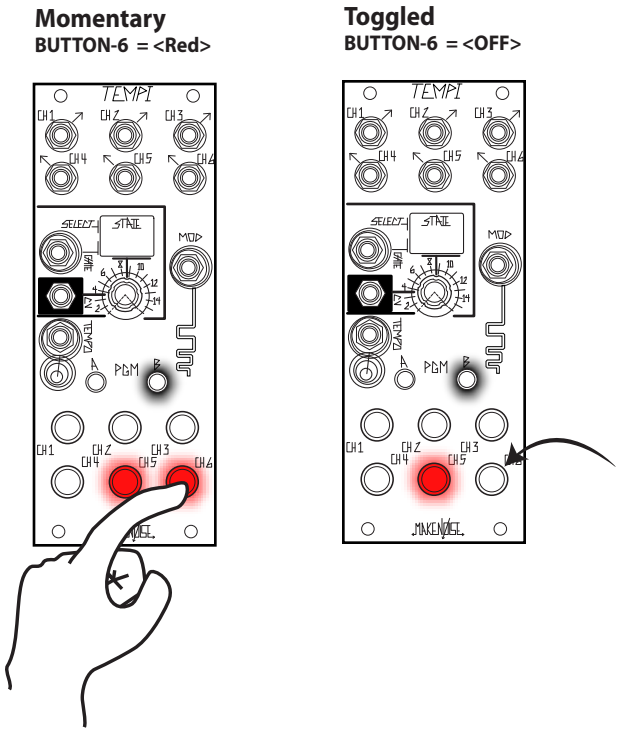
Makes CHannels ProGraMmed for **MOD = <PURPLE>**, **RUN** on **MOD <ON>** while CHannels not ProGraMmed for **MOD <BLUE>** will **STOP** on **MOD <ON>** and vice versa. The **MOD LED** indicates if the **MOD** is **<ON>** or **<OFF>**.

OFF **BUTTON-5 = <OFF>,** when **BUTTON-4** is **<ON>**

NOTE: The difference between RUN / STOP ALL and RUN / STOP with all CHannels ProGraMmed as MOD (<PURPLE>) is that RUN/STOP ALL is able to Reset the Variable Clocks.

{ProGraM EDIT}: RUN / STOP (cont'd)

Still in the **{ProGraM EDIT}** Page (**[Double Press] PGM_B = PGM_B <FLASHES>**), while **RUN / STOP** is active (i.e. **BUTTON-5 = <ON>**, any color), **BUTTON-6** sets the behavior of the **MOD IN** jack to be **Momentary** or **Toggled**.



Momentary **BUTTON-6 =<RED>**
Makes the **RUN / STOP** behavior mirror that of the **MOD Gate** INput. When the **MOD Gate** INput is "High," **MOD** is ON. Alternatively, when the **MOD Gate** INput is "Low," **MOD** is OFF.

Toggled **BUTTON-6 = <OFF>**
Makes the **RUN / STOP** behavior turn **<ON>** and **RUN** when **MOD Gate** INput goes "High" and not turn OFF and/or **STOP** until the **MOD Gate** INput goes from "Low" to "High" a second time. The **MOD LED <LIGHTS MAGENTA>** at the rising edge of the signal in the **MOD Gate** INput.

{STATE EDIT}: STATES

TEMPI stores up to sixty-four timing scenarios referred to as **STATES**, arranged into four **BANKS** of sixteen, with the **STATE LED** that indicates **BANK** by **<COLOR>** and **STATE** changes by **<FLASHING>**.

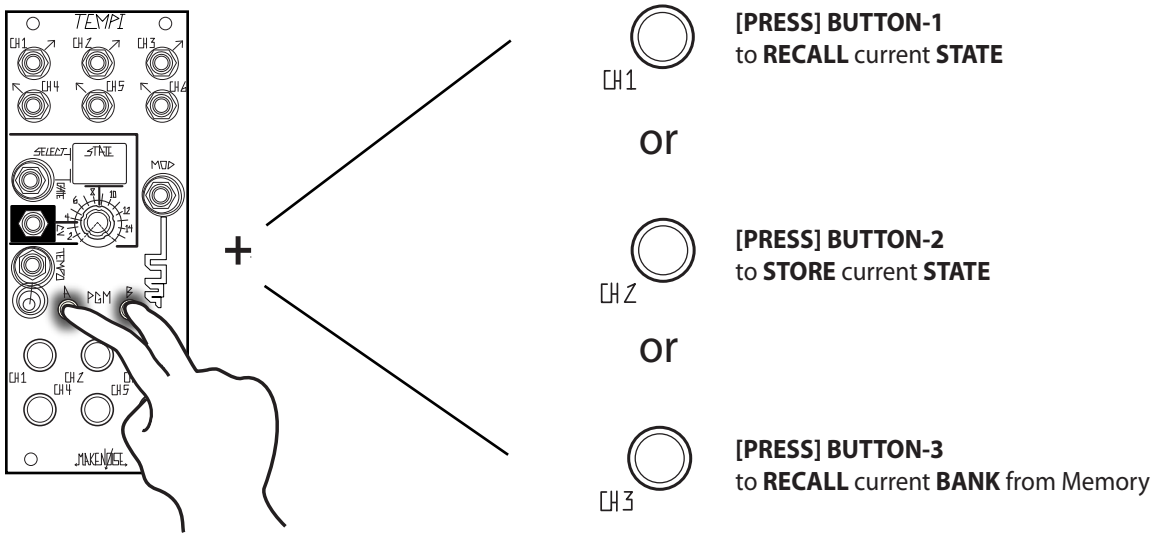
STATE is selected by either **STATE SELECT CV** or **GATE**, with **STATE SELECT CV** having top priority. The **STATE SELECT GATE** increments the **STATE** value at the rising-edge of each Gate. In other words, when a **STATE** change is read at **STATE SELECT CV**, that change is absolute and takes priority over **STATE SELECT GATE** changes.

There is a Gate INput for **STATE SELECT GATE** and a Combo Pot/CV INput for **STATE SELECT CV**. With nothing patched to **STATE SELECT CV** INput, the **STATE SELECT CV** Combo Pot operates as a Panel Control which manually selects the **STATE**. The **STATE SELECT CV** Range is 0V to 5V. The **STATE LED** above the **STATE SELECT CV** Panel Control **<FLASHES>** each time **STATE** is changed.

To **RESET** to **STATE 1**, patch a Gate to the **STATE SELECT** CV INput and set the Combo Pot/attenuator so at **GATE HIGH**, **TEMPI** goes to the last **STATE** you would like to use (**STATE 8**, for example). Next, patch a Clock or Gate to the **STATE SELECT GATE** INput and step through **STATES 1** through **8**. Send a **RESET GATE** to the **STATE SELECT CV** INput and at **GATE LOW,** you will return to **STATE 1**. In this way, it is possible to sequence any number of **STATES** linearly from **STATE 1** to **STATE *n*** before **RESET**ting.

While **{MUTE}, {MOD}, {HUMAN}, {MACHINE},** or **{PHASE}** ProGraMming are active, **STATE** is held and **DOES NOT reflect STATE SELECT changes via CV or GATE UNTIL active ProGraMming is exited or completed.** This behavior may be used creatively in order to manually "freeze" the **MOD** functionality or **STATE SELECT** modulation.

To **STORE** or **RECALL** a **STATE**:
[HOLD] PGM_A + PGM_B



Note: {STATE EDIT} settings are saved on STATE STORE.

{STATE EDIT}: COPY, PASTE, and MUTATE STATE

Use **STATE SELECT** Panel Control in order to audition and select a **STATE** to **COPY**. Note: it is important that you first remove any modulation source controlling **STATE SELECT** and/or **MOD** in order to accurately choose the desired **STATE**. When you find the desired **STATE**, **[HOLD]** both **PGM_A + PGM_B** and **[PRESS] BUTTON-4** to **COPY**. After a **STATE** is **COPIED**, you may then decide to **PASTE** an exact duplicate or a **MUTATEd** version of the **COPIED STATE**.

Once **COPIED**, use the **STATE SELECT** Panel Control in order to audition and select a **STATE** into which you would like to **PASTE**. **Keep in mind, you will be overwriting what is STOREd in the currently-SELECTED STATE!** To finalize the **PASTE** operation, **[HOLD] PGM_A** and **PGM_B** and **[PRESS] BUTTON-5**.

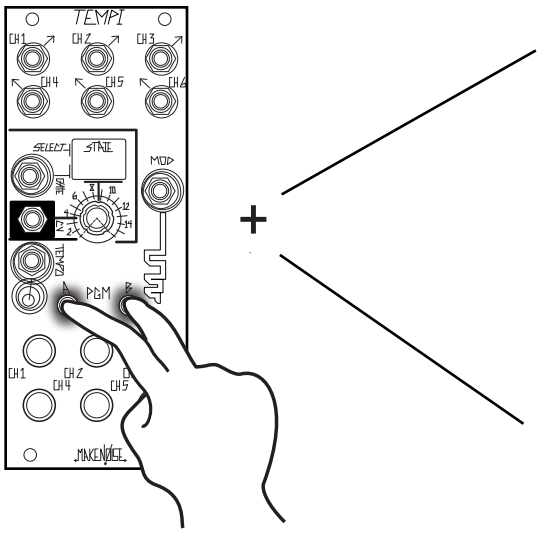
Alternatively, once a **STATE** has been **COPIED**, you may use the **STATE SELECT** Panel Control in order to select a **STATE** over which you would like to paste a **MUTATEd copy**. **MUTATE is very similar to the PASTE operation, so keep in mind, you will be overwriting what is STOREd in the currently-selected STATE!**

When you find the desired **STATE** to overwrite, **[HOLD]** both **PGM_A + PGM_B** and **[PRESS] BUTTON-6** in order to paste a **MUTATEd** version of the previously-**COPIED STATE**. **MUTATE** works by deviating to a small degree from the values that are already ProGraMmed within a **COPIED STATE.** To create more dramatic and alarming Mutations, simply, repeat the **COPY and MUTATE** procedures again… and again. **Note: is possible to COPY, PASTE, and MUTATE several times to quickly and easily ProGraM a STATE or BANK.**

Please note that these changes ARE NOT STOREd UNTIL you STORE the STATE or BANK. Without running the STORE operation, changes ARE NOT held after power is cycled. This is useful for improvising alternate versions of a STOREd theme.

To **COPY, PASTE,** or **MUTATE** a **STATE**:

[HOLD] PGM_A + PGM_B



+

[PRESS] BUTTON-4
to **COPY** a **STATE**

or

[PRESS] BUTTON-5
to **PASTE** a previously-**COPIED STATE**

or

[PRESS] BUTTON-6
to **MUTATE** a previously-**COPIED STATE**

{BANK EDIT} Settings

In order to enter **{BANK EDIT}** ProGraMming page, **[DOUBLE PRESS] PGM_A. PGM_A <FLASHES>** in order to indicate **{BANK EDIT}** page. To exit the **{BANK EDIT}** ProGraMming page, **[DOUBLE PRESS] PGM_A.**

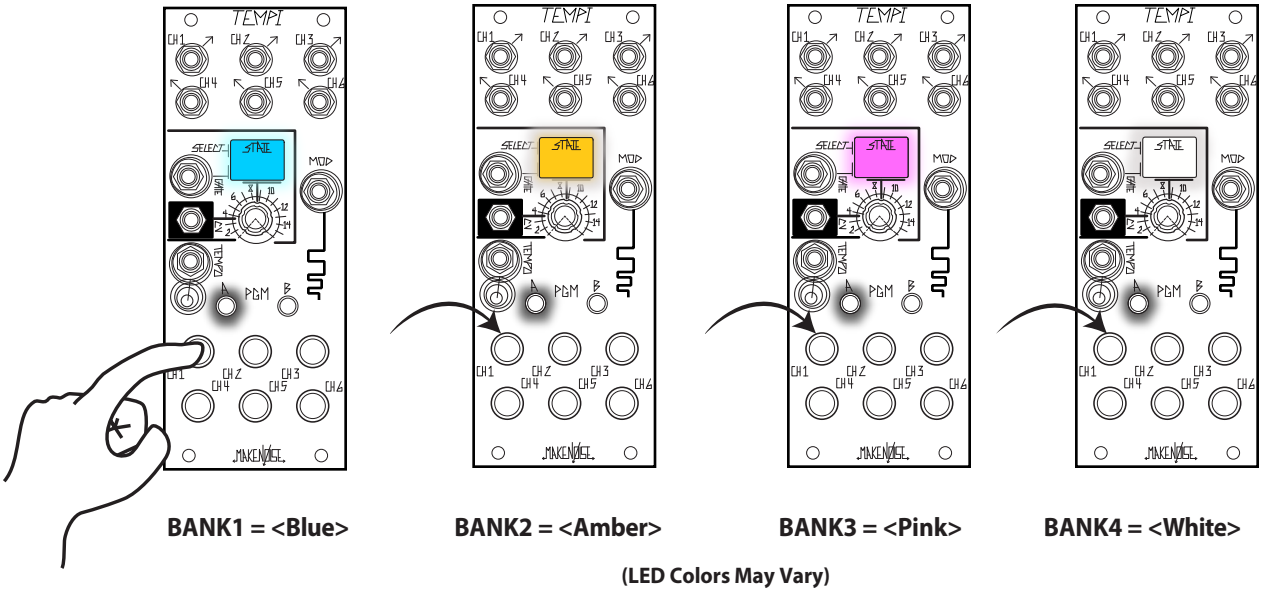
{BANK EDIT}
[DOUBLE PRESS] PGM_A:

- **[PRESS] BUTTON-1** to **SELECT BANK**
- **[PRESS] BUTTON-2** to **STORE ALL BANKS** and current **TEMPO**
- **[PRESS] BUTTON-3** to **FOLLOW SELECT & TEMPO Busses**
- **[PRESS] BUTTON-4** to **COPY** selected **BANK**
- **[PRESS] BUTTON-5** to **PASTE** over selected **BANK**
- **[PRESS] BUTTON-6** to **MUTATE** over selected **BANK**
- **[PRESS] PGM_A** to Exit **{BANK EDIT}**

SELECT BANK:

In order to **SELECT BANK**, within the **{BANK EDIT}** ProGraMming page (to access, **[DOUBLE PRESS] PGM_A = PGM_A <FLASHES>**) and **[PRESS] BUTTON-1** for sequential selection of **BANK1, BANK2, BANK3**, or **BANK4**.

The **STATE LED** color reflects **BANK** change as follows:



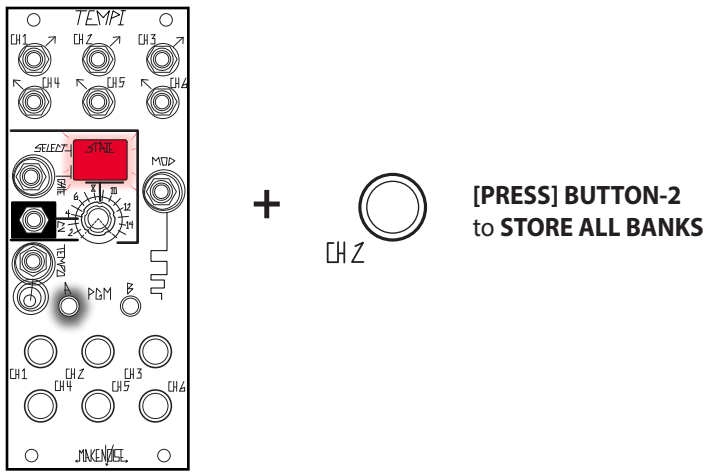
BANK1 = <Blue> **BANK2 = <Amber>** **BANK3 = <Pink>** **BANK4 = <White>**

(LED Colors May Vary)

STORE ALL BANKS:

In order to **STORE** all contents of all **BANKS** (i.e. **ALL STATE**s and **Leading TEMPO**), within the **{BANK EDIT}** ProGraMming page (to access, **[DOUBLE PRESS] PGM_A = PGM_A <FLASHES>**), **[PRESS] BUTTON-2**. The **STATE LED <FLASHES RED>** in order to indicate **STORE** complete.

[DOUBLE PRESS] PGM_A



[PRESS] BUTTON-2
to **STORE ALL BANKS**

{BANK EDIT} Settings (CONT'd)

COPY, PASTE, and **MUTATE BANKS:**

Still in the {BANK EDIT} (**[DOUBLE PRESS] PGM_A = PGM_A <FLASHES>**) page, in order to **COPY**, **[PRESS] BUTTON-1** to select the **BANK** to **COPY**. When you find desired the **BANK**, **[PRESS] BUTTON-4** to **COPY**.

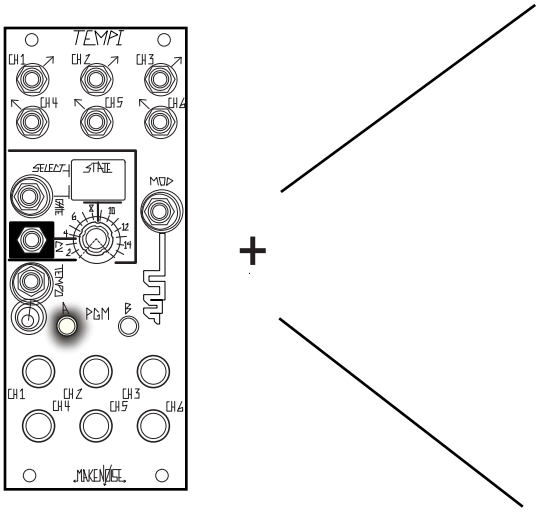
In order to **PASTE**, **[PRESS] BUTTON1** to select the **BANK** into which you would like to **PASTE**. **Keep in mind: you will be overwriting what is STOREd in the selected BANK!** Lastly, **[PRESS] BUTTON-5** to complete the **PASTE** operation.

MUTATE works by deviating to a small degree from the values that are already ProGraMmed within a **COPIED BANK**. In order to **MUTATE** a **COPIED BANK**, first **[DOUBLE PRESS] PGM_A** to migrate to the **{BANK EDIT}** ProGraMming page. Next, **[PRESS] BUTTON-1** to select the desired **BANK** to **MUTATE** paste. When you find the desired **BANK**, **[PRESS] BUTTON-6** in order to paste a **MUTATEd copy** . To create more dramatic and alarming mutations, simply repeat the **COPY** and **MUTATE** procedures again… and again.

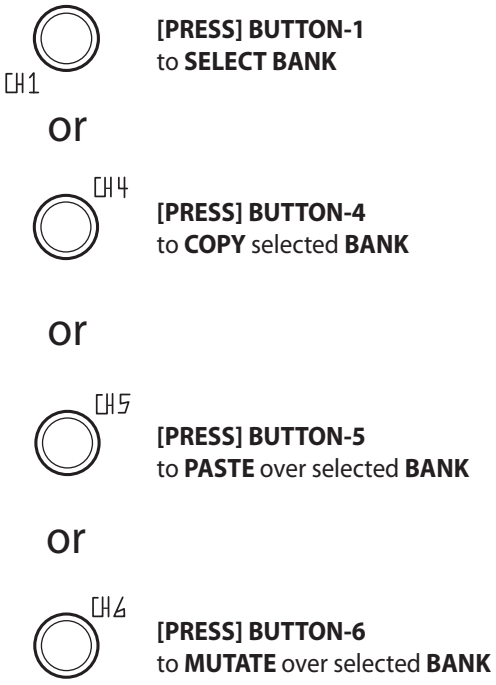
Note: it is possible to **COPY, PASTE,** and **MUTATE** several times to quickly and easily ProGraM a **BANK**. **MUTATING** a CHannel that is ProGraMmed with a 1:1 MULTIPLE or DIVISOR **Variable Clock** value will have very little effect.

To **COPY, PASTE,** or **MUTATE** a **BANK**:

[DOUBLE PRESS] PGM_A



\+



[PRESS] BUTTON-1
to **SELECT BANK**

[PRESS] BUTTON-4
to **COPY** selected **BANK**

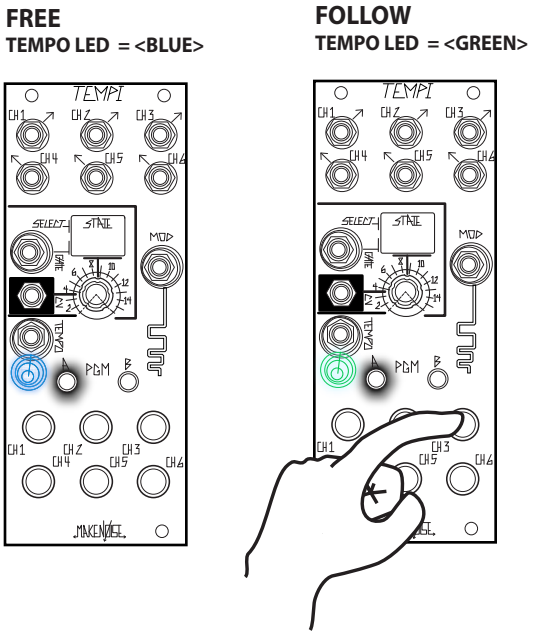
[PRESS] BUTTON-5
to **PASTE** over selected **BANK**

[PRESS] BUTTON-6
to **MUTATE** over selected **BANK**

{BANK EDIT} Settings (CONT'd)

FREE / FOLLOW

In order to set the **TEMPI** to **FOLLOW** the **TEMPO** and **SELECT Busses**, **[PRESS] BUTTON-3**. The **TEMPO LED** is **<GREEN>** to indicate **FOLLOW**. **[PRESS] BUTTON-3** again to set **TEMPI** to be **FREE** of the **TEMPO** and **SELECT Busses**, indicated by the **<BLUE> TEMPO LED**.



While set to **FOLLOW**, the **TEMPI RECEIVES** messages from the **SELECT Bus** that help determine the **STATE** and a Clock from the **TEMPO Bus** to determine the **Leading TEMPO**. **IT DOES NOT TRANSMIT messages**.

The Panel Controls for **STATE** and **TEMPO** are still useful when the **TEMPI** is set to **FOLLOW**. The **STATE SELECT** CV and GATE allow for offsetting from and modulating around the **STATE** that is selected by the **SELECT Bus**. The **TEMPO Bus** may be completely overridden by patching a Clock to the **External TEMPO** INput, thus allowing for an independent **Leading TEMPO**.

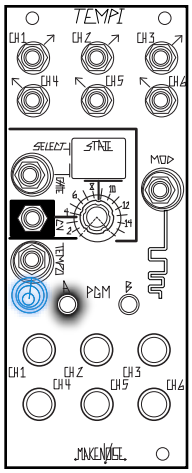
Please note: changes ARE NOT STOREd UNTIL you go to {BANK EDIT} ProGraMming page and [PRESS] BUTTON-2 to STORE ALL BANKS. Without running the STORE ALL BANKS operation, changes are held ONLY UNTIL the power is cycled. This is useful for improvising alternate versions of a theme.

{BANK EDIT}: SELECT BUS and TEMPO Buses

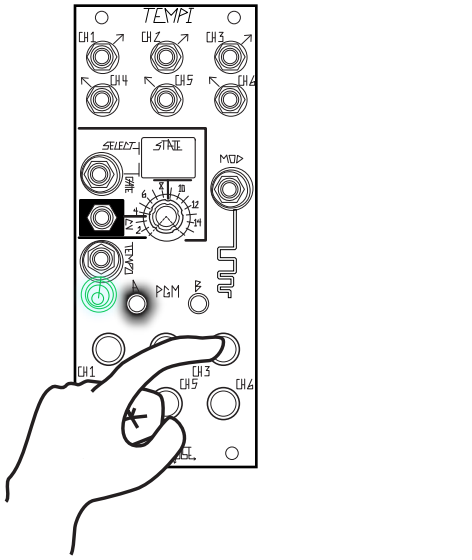
The **SELECT Bus** and **TEMPO Bus** utilize the often-inactive CV and GATE Busses in the Eurorack system in order to allow remote (i.e. patch-less) control of **STATE SELECTion** and **TEMPO** for **TEMPI** and other modules designed with this standard in mind. Any number of **TEMPI** on the same Bus Board **can be controlled** simultaneously by these signals, simplifying the process of macro-control over complex changes in a patch. It is worth noting, **TEMPI** only **RECEIVES** messages over the **SELECT/TEMPO Buses**. **IT DOES NOT TRANSMIT**.

To engage Bus Control, enter **{BANK EDIT}** page (**[DOUBLE-PRESS] PGM_A**) and **[PRESS] BUTTON-3**. When the **SELECT** and **TEMPO Busses** are enabled, the **TEMPO LED** will **<FLASH GREEN>** instead of **<BLUE>**. Now, the **TEMPI** will be controllable by Clock and **STATE SELECT** signals coming through these Busses.

SELECT and TEMPO Busses Disabled
TEMPO LED = <BLUE>



SELECT and TEMPO Busses Enabled
TEMPO LED = <GREEN>



Before engaging, make sure that no other modules are sending CV, Gate, or other signals (other than **SELECT** and **TEMPO** signals) through the Busses.

More information regarding **SELECT** and **TEMPO** Buses will become available as the standard is developed.

Tips and Tricks

• It is possible to ProGraM several CHannels at once. This makes it more intuitive to create interesting timing arrangements. This is true of **MACHINE**, **HUMAN**, **MOD**, **MUTE**, and **PHASE** ProGraMming.

• **Coarse PHASE adjustments** OVERRIDE **Fine PHASE adjustments**. In this way, **Coarse Adjustments** return to **INTEGRAL DIVISORS** and **MULTIPLIERS**, for instance, 1:1 ratio.

• **MUTE**s are not **SHIFT**ed. One could emulated a **SHIFT**ed **MUTE** by programming a large **DIVISION** and **SHIFT**ing. For example, this allows for turning off a voice within a patch or stopping a sequence.

• You must first **COPY** a **STATE** or **BANK** before you can **MUTATE** it.

• If **HUMAN** ProGraMming and you would like to use only INTEGER Divisions or Multiples, set to **Human Resolution 100%** (**[PRESS] Button-3** on **{ProGraM EDIT}** Page). This will ensure instantly perfect ProGraMming of 'four on the floor' beats, or something similar.

• If using **SHIFT**, and changing **STATES**, turn **JUMBLED SHIFT** ON to preserve **SHIFT**ed channels after leaving and returning **STATES**. This is useful for creating Live modifications to timing arrangements. This is also the default.

• If you would like to return **SHIFT**ed channels to their original positions, turn **JUMBLED SHIFT OFF** and then leave and return to **STATE**. This is useful for creation Live modifications to timing arrangements without messing up the original settings.

• The **STATE SELECT CV** and Panel Control override any **STATE** stepping that occurs. So if you stepped through **STATES** with **STATE SELECT** Gate and then changed **STATE SELECT** Panel Control, it immediately takes it back to the Panel Control (or CV) setting. For example, if set to **STATE 1** then **STATE STEP** six times with **STATE SELECT** Gate to **STATE 6** and then turn the **STATE SELECT** Panel Control to **STATE 2**, the next **STATE** would be **STATE 3**, as the knob position and **STATE SELECT** CV takes priority when moving.

• It is possible to ProGraM a new **Variable Clock** into a **MUTED** CHannel. This is useful for cueing changes during live performance.

• The **SHIFT Register** function and **MODded** CHannels do not activate without a Trigger or Gate patched to the **MOD** Gate Input. For **SHIFT**, at least two CHannels must be **MOD**-enabled. If only one CHannel is **MOD**-enabled it can only **SHIFT** with itself (i.e. no visible effect).

• **MUTATING** a CHannel that is ProGraMmed with a 1:1 DIVISOR or MULTIPLE value will have very little effect.

• If a **STATE** or **BANK** is **COPIED** and then **MUTATEd**, it is still possible to **PASTE** an **unMUTATEd** iteration of the associated **STATE** or **BANK**.

•To create more dramatic and alarming Mutations, simply, repeat the **COPY and MUTATE** procedures again… and again. In this way, it is possible to **COPY, PASTE,** and **MUTATE** for easily ProGraMming a **STATE or BANK.**

• For a stable **TEMPO**, once the **Leading TEMPO** is learned, unpatch the Clock from the external **TEMPO** INput. As such, to create Clock Jitter, pre-process your **External Clock** before it is patched to the **TEMPO** INput. Even as the **External Clock** rate changes, the **TEMPI's Variable Clock** OUTputs will remain synchronized as long as the **External Clock** remains patched to the **TEMPO** INput.

•Self-patching the **TEMPI's** OUTputs to the **MOD**, **STATE SELECT** GATE INput, and even **STATE SELECT CV** INput can be very useful. However, be aware that self-patching to the **TEMPO** input will result in a feedback loop that causes the module to quickly reach maximum or minimum **TEMPO** value (unless the CHannel patched to **TEMPO** INput is set to 1:1, in which case it will have no effect at all).

•To **RESET** to **STATE 1**, patch a Gate to the **STATE SELECT** CV INput and set the Combo Pot/attenuator so at **GATE HIGH**, **TEMPI** goes to the last **STATE** you would like to use (**STATE 8**, for example). Next, patch a Clock or Gate to the **STATE SELECT GATE** INput and step through **STATES 1** through **8**. Send a **RESET GATE** to the **STATE SELECT CV** INput and at **GATE LOW,** you will return to **STATE 1**. In this way, it is possible to sequence any number of **STATES** linearly from **STATE 1** to **STATE *n*** before **RESET**ting.

•**SHIFT** is also useful for ProGraMming variations of **STATES** where you want use the same **TEMPI** CHannel values, but on different CHannels. First, create your initial **STATE**. **COPY** and then **PASTE** to another **STATE**. Now, **[PRESS] PGM_B** to enter **{MOD}** ProGraMming page and set two CHannels to be **SHIFT Enabled** (**BUTTON1-6 = <Purple>**) and send a Gate signal to the **MOD** INput to swap them. Now, set these two CHannels back to not being **MOD enabled**. Use this technique to trade CHannel values and create new **STATES**.