



## SYSTEMS

is an album archiving written and sonic explorations of various analog and digital systems that I use during the process of designing sound, as an attempt to demystify some synth esoteria, and a continuation of the story of the data bird, amongst other poems.

---

### *Listening Instructions (suggested):*

Listen to this album in full or in smaller periods of time, in direct sunlight, or a low lit environment, preferably with headphones, or speakers spread far apart. Sit or lay down in a comfortable place and position. Have your favorite snacks or beverages nearby for occasional consumption. Read the track stories as the corresponding tracks play.

---

### *Meditation:*

Internet meditation interlude entrance opening introduction cleansing clear cmd delete.

### *Bath:*

A analog bubblebath with warm water and starburst candy bath bomb, with non mainstream super esoteric ambient music playing, and fruit loops scented candles, the water reaches your body temperature and you close your eyes and start to float away, on cotton candy clouds, bubbles begin to sound larger than life. The internet seems far, far away. The concept of dirt is erased from your memory. Too many things are erased from your memory, but you're okay with it, because the only direction is forwards.

### *digitamblamAh:*

Finally some quiet, a taxi. Steady travel. At night. With a gin and tonic, chauffeured. Take me to a black tie event. With normal people. Where I can't make out faces. A place where I can ignore important tasks. And time doesn't move. Dragged by the

hand into an endless hallway. Escaping with someone, anyone. If I could fill all my bags with the items on these walls I would. And fill a room with infinite things. A hoarder's dream. But just to look at. I have a separate room for the things I used. Stumbling. She's walking maybe a bit too fast.

***i get too cold drawn some that:***

A song in silence: Its so I get it's so I get its for to go on or not its so I get it's there you were cold in it call the word hold to he had night after as. A thought about the idea of. What do I you actually want? Inherently peaceful. In its simplicity.

***jazz beat in soil (movement 1 and 2):***

The jazz bar on  $\dagger\Delta^{\times\times} \equiv \Delta^{\vee}$  and opcod 19287 off of conduit Jester. A quintet for romance, in two movements. With a Paloma. Movement two suggests a red, 2021 preferably. And filet mignon. I can afford I t, for both of us.

***pearls:***

Back to work, wired. I'm dating the boss's daughter, I gotta be there on time. Little does he know she's cheating on me. What a scandal. With Cathy in HR. I knew she was a lesbian. You can't sneak out an air lock, and she was bad at excuses. Oh your book club, all the way in sector G. I haven't said anything about it yet. This job pays well. And Cathy is 7 feet tall. I'm only 2 feet 4 inches and 45 centimeters. I can't compete. She'll punt me. As long as I get my numbers in this quarter. I'll quit. I'll have a cig now. I never thought to go out on the balcony. No one ever does. Probably because we work on the 596th floor. The air is dry up here. I think I'll go out for a cig now. The whole office won't get sucked out through this door once I open it right haha. [the entire office did indeed implode in on itself through the glass pane door the second it was pulled inward.] Well... definitely getting fired now.

***Pleasantly drunk, with wings:***

A return to some home. A dinner at someone else's. With a beer, driven in a car with many miles. Those swirly things you see when you get up too fast, haven't eaten all day, or stare at a perfect blue sky, have a sound. An awkward couple, yet charming. Retired.

***swih aybz intersexon:***

You find yourself millenia deep in incomprehensible alien data simplex intersections, figures of light speed past and begin to intimidate you. A confrontation. With only words, a new story. With noise and absence of time. A feeling of longing. A nostalgia of the unknown. An afrofuturist retrofuture. Dark in headlights. Frozen, melting and falling infinitely. Limbo, traveling.

***All of your technology will function properly this mercury retrograde:***

Place 10 fruity pebbles on your phone and laptop every morning and night for good luck with technology this mercury retrograde. Something else that might help is suspending a bag of your technology above a large body of water for 5 seconds, or bringing your technology at least 1000 ft above sea level.

---

## Systems Technical Guide:

### ***Meditation:***

The piano part in this patch was composed using Teletype eurorack module.

Teletype has been an important part of my eurorack system for quite some time now. I like it because it allows me to have a lot of control over timing and random voltage in my system. It has its own very simple language, with 8 scripts and 6 lines of code per script. It's a fun challenge to fit my code into this short amount of space, and I've gotten a lot of fun results from the puzzle of trying to consolidate my code. A lot of the scripts I build on teletype start with random timing for calling scripts, and using voltage or random numbers as trigger information for drum or synth modules using the binary operators. This way I can use a number like 10 and get a binary number 0101, which when sequenced through linearly I can make a simple hat pattern out of. In another example 8 which is 1000 in binary could be the kick pattern. In my code I'll set a condition that 1 in binary sends a trigger and 0 is a rest, or sends a pulse out of a different trigger out. The other method I use is to send the corresponding bit to the outputs I have available. For instance if I'm pulling the binary number 0110, I can tell teletype to get the bit at all the indexes and send a pulse out of the corresponding output if that bit is set to 1. So in this case it will send a pulse out of pulse output 2 and 3 because bit two and 3 of 0110 are set to 1. This code looks like:

```
L 1 4: TR.P ? BGET A I I 0
```

*Loop 1-4: Trigger Pulse, If the bit of variable A at index 1-4 is true, at that index, otherwise do nothing.*

*Loop 1-4: If the bit of variable A at index 1-4 is true, Trigger the pulse corresponding to the true bits at their index, otherwise do nothing.*

Usually I'll choose numbers anywhere between 0 and 65536 which is the largest number for 16 bits of binary, or a 16 step sequence in my use case. That gives me a really large range for creating sequences. I don't usually encode a number specifically to create a certain pattern, but rather use the process to make random patterns from single sources of voltage or local variables I'm using in a script. This allows me to create complex generative patterns easily, and keep my whole patch in time. It also allows me to save and recall patches easily, for live performance.

The Teletype talks to a few modules in my system over a protocol similar to midi called I2C. The main modules I use it most with are the ER301 sound computer and the Disting EX. Because these are not physical patch points, I can send infinite channels of random voltage, triggers and cv to control various parameters, and trigger drums on both the ER301 and Disting EX and get lots and lots of variation within my patches. I often send random changes in values of parameters on the Polyphonic Wavetable algorithm on the Disting EX, such as the filter cutoff, detune, delay times, and wavetable position. I often send random voltage over I2C to the ER301 sound computer to control various parameters of the granular effects. I can send I2C to the Disting EX to control parameters from teletype using the code:

```
EX.P X Y
```

*Where X is the parameter number and Y is the value*

I can send voltage to the ER301 sound computer using the code:

```
SC.CV X Y
```

*Where X is the CV number (goes up until 100) and Y is the value*

I can send pulses to the ER301 sound computer using the code:

```
SC.TR.P X Y
```

*Where X is the trigger number and Y is the value*

Sometimes I also use I2C2MIDI which is a module developed by Attowatt and the makers of teletype thst lets me send midi information coded from teletype. I sometimes use this to control external gear via MIDI CC, or send note data to trigger drum machines and external synths over midi, all from teletype. I can send midi from Teletype though I2C2MIDI with the code:

```
I2M.N# X Y Z
```

*Where X is the midi channel, Y is the midi note and Z is the midi velocity.*

What's unique about I2C2MIDI is midi cc slew! Which allows me to send random values of midi cc, but they can interpolate over a certain amount of time that I can set or change dynamically. I think everything that can send midi cc should have midi cc slew. It opens up a whole world of timing and composition for instruments that may not have that sort of slow timing movement, and especially brings an organic nature to the 127 step values of midi cc.

Here is an example of a full teletype script I might use in one of my live sets:

```
#I
PARAM.SCALE 400 12000
EX.PRE 49
L 1 4: TR.TIME I 20
LIVE.DASH 1

#1
DEL RRND 20 PRM: $ $
K ? LT LAST 1 200 4 8
DEL.R K / LAST 1 K: $ 2; $ 5
J ? TOSS 12 9; $ 4; Z RND 8
EX.N N + * ? TOSS 5 9 RND 4 J
DEL.R RND 8 / LAST 1 Z: $ 3

#2
A RND 16; B RND 256; $ 5
L 0 3: TR.P ? BGET A I + 1 I 0
L 0 7: $$ 4
SC.TR.P ? BGET B I + 1 I 0
L 0 15: $$ 6
I2M.N# 1 ? BGET B I PN 0 I Y
```

```
#I
# corresponds to the script page
I Is the Init script, it fires once and it's where
you can set variables that will load when you
load Teletyoe or load the script
```

```
PARAM.SCALE 400 12000
This changes the scale of the parameter knob on
teletype so I can use the knob in a range of
values that I find useful for my script, in this
case I am going to use it for timing so I set it to
go between 400 when the knob is turned fully
counterclockwise and 12000 when the knob is
fully clockwise.
```

```
EX.PRE 49
This tells the Disting EX to load a specific
preset when the teletype script is fired. I usually
change the parameters as I go and make sure to
save them as a preset, and write this line of code
in my init script before turning off my modular
```

#3

```
J ? TOSS QT RND -36 36 12 7
EX.N N + * ? TOSS 5 9 RND 4 J
```

#4

```
L 1 16: SC.CV I V RND 10
L 1 16: SC.CV.SLEW I * LAST 1 2
```

#5

```
Y RRND 50 127
PRINT 1 A; PRINT 2 B
```

P#

36	0	0	0
37	0	0	0
38	0	0	0
39	0	0	0
40	0	0	0
41	0	0	0
42	0	0	0
43	0	0	0

so that I can recall the script with my settings when I turn my modular back on.

L 1 4: TR.TIME I 20

L runs a command sequentially between two numbers. This line of code is saying set the trigger pulse time for trigger outputs 1-4 to 20 milliseconds. If I'm triggering drums or envelopes in my eurorack system I usually want a short gate time from Teletype because I rarely use sustain, so that's something I like to set in my init script.

LIVE.DASH 1

This turns on the dashboard for the live view of teletype, where I can display useful information, or useless information.

#1

DEL RRND 20 PRM: \$ \$

This line of code says wait a random time between 20 milliseconds and whatever the param knob is set to and fire script, or fire the current script. This loops the script, the script fires itself and chooses a random timing for when to fire itself again. I use this to create random timing for when the script is fired, with control over the scale of that randomness with the param knob.

```
K ? LT LAST 1 200 4 8
```

*This sets the local (per script) variable K to a 4 if(?) the time it took to fire the last script was under (Less Than) 200 milliseconds, and 8 if the time it took to fire the last script was over 200 milliseconds. I do this because I'm about to divide the timing of this script to trigger drums on the next script, and I want to divide the number under 200 by 4 and not 8 because dividing shorter times by 8 would be a bit too fast for me most of the time. But I'm okay with time above 200 milliseconds to be divided by 8, so I set this conditional statement.*

```
DEL .R K / LAST 1 K: $ 2; $ 5
```

*This line of code sets a ratcheting delay, saying trigger script 2 and 5 K times (where K either 4 or 8, that we set in the last line of code) at a division of the time it took to trigger the script last by K. for instance if it took 800 milliseconds to fire the script, this line of code will trigger script 5 eight times in increments of 100 milliseconds.*

```
J ? TOSS 12 9; $ 4; Z RND 8
```

*This line of code sets the variable J to either 12 or 9 on a coin toss (50% chance) and (;) also fires script 4 and also sets the variable Z to a random number between 0 and 8*

```
EX.N N + * ? TOSS 5 9 RND 4 J
```

*This line of code sends note and trigger information to the Disting EX. (usually in poly wavetable or SD multisample mode, but can be used with the granular synth, poly exciter and poly FM modes as well).*

*The note is the number returned when this function is set preformed:*

```
multiplies (either 5 or 9) by a random number between 0 and 4 and ads J to that value.
```

*This convoluted way of creating note data lets me stick within the range of certain notes but quantize the randomness between either one note or the other, or one octave or the other or a random octave, etc.*

```
DEL .R RND 8 / LAST 1 Z: $ 3
```

*This is another ratcheting delay line of code that triggers script 3 a random number of times between 0 and 8 and with a (separate) random division of the last script between 0 and 8.*

```
#2
```

```
A RND 16; B RND 256; $ 5
```

*Sets the variable A to a random number between 0 and 16, and B to a random number between 0 and 256, and fires script 5*

```
L 0 3: TR.P ? BGET A I + 1 I 0
```

*Trigger pulse (if the bit of A at index I is true) I (1-4) otherwise trigger pulse 0 (do nothing)*

*This line of code is using the binary version of the random numbers our variables represent to create trigger information. Its saying trigger the pulse output corresponding to the bit that is high at index 1-4 (The first bit is 0, but trigger outputs are numbered 1-4, so I have to add 1 when calling the variable I of the index of L).*

*If A is 3, for example, the binary for 3 is 0011 so it will send a pulse out of trigger outputs 3 and 4, because both of those bits are high, or true.*

**L 0 7: \$S 4**

*Because there is a limit to the number of characters per line, sometimes I have to divide one line of code into two parts. This line just says call line 4 of this script between 0 and 7 times sequentially. (think of I as the counter of where you are in the sequence, and L as firing this whole sequence immediately, with no time in between steps)*

**SC.TR.P ? BGET B I + 1 I 0**

*These last two lines are the same as the pervious, however SC.TR.P fires triggers on the ER301 sound computer, over I2C*

**L 0 15: \$S 6**

**I2M.N# 1 ? BGET B I PN 0 I Y**

*This line of code is doing the same thing as the previous lines of code, however I2M.N# is sending midi notes on channel 1 with a velocity of Y (which is set to a random number between 50 and 127 in the next script). Instead of using the index number we are using midi notes that are being called from the first pattern with PN 0 I or pattern number 0 at index I (0-15). This will fire the corresponding midi note to high bits out of the midi output of I2C2MIDI.*

**#3**

**J ? TOSS QT RND -36 36 12 7**

*This sets the variable J to either 7 or a random number between -36 and 36 quantized to 12 so either -36, -24, -12, 0, 12, 24, or 36.*

**EX.N N + \* ? TOSS 5 9 RND 4 J**

*This is similar to the last code for notes for the disting, however I am not adding our new notes from J to the equation.*

**#4**



```
L 1 16: SC.CV I V RND 10
```

```
L 1 16: SC.CV.SLEW I * LAST 1 2
```

*In these two lines I'm sending 16 channels of random voltage between 0 and 10 volts to the ER301 sound computer, with slew or interpolation between these values with a timing of how long it took the last script to be fired multiplied by 2. If the script was fired 900 milliseconds ago it will interpolate between the last random voltage and the next random voltage over the course of 1800 milliseconds, or 1.8 seconds.*

---

```
#5
```

```
Y RRND 50 127
```

```
PRINT 1 A; PRINT 2 B
```

```
PRINT 3 PRM
```

*Print allows me to see values in the live view of teletype on my dashboard for reference. I'll have variables A and B be displayed as binary numbers with the code %B1 and %B2 on my title page, so I can see what pattern is coming out of my trigger outputs, and have the value of the parameter knob be displayed, so I can see my random timing range. Here I also place the date and name of the live show, set or patch, and sometimes the whole bill and flier.*

---

```
P#
```

```
36 0 0 0
```

```
37 0 0 0
```

```
38 0 0 0
```

```
39 0 0 0
```

```
40 0 0 0
```

```
41 0 0 0
```

```
42 0 0 0
```

```
43 0 0 0
```

*The first column of values is the pattern that we were pulling from for our midi notes back in script 2. For example these notes trigger the 8 tracks of the octatrack.*

Teletype is sending notes to the Disting EX Multisample Algorithm (mode) playing the LABS Piano Multisample library. An example code for this would be:

```
EX.N N * ? TOSS 5 9 RND 5
```

which says send note 5 (F) or 9 (A) with a random multiple of 5. Teletype sends random timing using the code:

```
DEL RRND 200 8000: $ $
```

which calls the script again with a random time between 200 milliseconds and 8 seconds. The piano also goes into the Chloe module, doing pt2399 delay. I am modulating the delay time by hand. The vocal is processed using the Codec plugin/vst in Ableton, based off of bad internet bandwidth connections. The vocal has a simple eq and gate before the processing to clean it up. The vocal is also processed by the RickBrownStutter Max For Live device. There is an auto panner on the track panning it from left to right sometimes.

Granular effects are a big part of a lot of my patches, songs and sound design. Granular for me isn't so much about chopping audio into small grains and overlapping them or scanning through them to create beds of sound and ambient pads (sometimes it is) but more about using the techniques of granular to mutilate existing, sampled or live samples content and tear them apart limb by limb. I use granular techniques in my eurorack system to stretch drums and create organic (to me) rubber, wind and other textures. Granular to me is also more than a "granular synth." Granular can be anything more than one delay pedal. If you have two delay pedals, you're creating two sizes of grains that can be overlapping or interact with each other. I think a lot of the idea of granular is that you need a granular synth or processor to make granular sounds but a lot of it is delay and looper processing, which can be done with delays and loopers, and really anything that holds an audio buffer.

### ***Bath:***

The Rollz section of my Rollz5 is patched to send sputtering pulses into MengQi Wingie2, whose output is sent back to the Ultrasound Filters section of the Rollz5 for filter processing. The output of the filters is sent to the L and R input of the delay on the Cocoquantus, which is set up as a simple delay. A slow moving and frequency modulated oscillator from the Quantussy section of the Cocoquantus is sent to the volume of the left delay, and a copy of this signal is sent out to the inverso volume of the right delay, creating undulating fast and slow panning.

The Rollz5 for me has been a really weird instrument. I was (and still am) obsessed with the entire world of Ciat Lonbarde and Peter Blassers circuits. Peter designs circuits based off of nature, his surrounding environment, math, geometry and geography amongst various other things. There is artistry and attention to detail in every scale all the way down to the

design of the circuit board, and the layout of components. Peter's attention to making his instruments extremely interactive with either surface touch as a way of patching, or antenna as ways of gesturally interacting with the synths make them unique. His paper circuits made it easy for me to start soldering and building circuits, which I had never done before. There's something interesting about working with paper instead of PCBs as well, the organic nature of the paper disintegrating over time, and the electronic components being left in their place, as a skeleton of the body that was the story behind the circuit. The rollz is a collection of a bunch of paper circuits. I'm attracted to the esotericism of these designs, not everything is labeled, you don't *exactly* know how all the circuits work. It takes your mind off of trying to learn a synth and *fourses* you to have in depth conversations with the instruments with your hands or with banana cables and patching systems. None of these circuits really act the way you want them to, they always feel organic to me, which is why I love having them around, in this collection. They always produce a new weird sound, pattern or gesture.

---

***digitamblamAh:***

The Lorre Mille Double Knot creates binary CMOS sequences triggering left and right oscillators with short decays. The first shift registers voltage information is sent to an fm input on one of the oscillators on the Cocoquantus at audio rate. This oscillator is sent to the left and right delay input as the sound source. The other shift registers voltage out is sent to the CV input for the left and right delay times. The Double Knot goes into the Wingie 2 before the Cocoquantus resulting in some resonating tones throughout the track.

---

***i get too cold drawn some that:***

My voice is looped on two tracks into a 6 second tape loop. I sing into the mic of the Cocoquantus and have my delay times set differently, record into the buffers occasionally and layer notes by inserting into these buffers in both long and short periods of time. Sometimes I pitch the recorded buffer upwards.

### *jazz beat in soil (movement 1 and 2):*

This patch is oriented around coalescence M4L plugin that chops and sorts samples that can be triggered via MIDI. I dragged a few Deadmou5 songs into coalescence which chopped them up and sorted them by timbre, assigning certain MIDI notes to certain timbres, so that I can use the plugin as my drum rack. I am triggering these different chopped drum sections with MIDI notes from Orca. Orca is also sending MIDI to a M4L clone of plaits, whose all parameters are being changed with every MIDI note received using Probee (M4L) (But sometimes I also use expression control, which comes with ableton). Both the drum track and synth track have lots of effects processing, mostly granular, stutter effects and reverb. Some techniques I use to clean up the sound is sidechaining the synth track to the bass of the drum track, so the synth ducks when there's bass present from the drum track. Sometimes I split the bands of the drum track in order to put an envelope follower on the bass that I can send to a notch filter on the synth track or to duck the reverb wetness of a synth track (these are more present on iteration 2).

Orca for me has been another fun puzzle in creating generative systems for note and trigger information. Because orca is on an x, y grid there are a lot of fun patches that happen from trying to fit your code in tight spaces, or bits of code interacting and overlapping with other bits of code. And the grid makes it interesting to block out lines of code, because some things may be only parts of your code that you're blocking out, so some techniques I use a lot are blocking random lines across, to get variations in timing and feedback between parts of code. Like drowning my code as it gasps for air. Let me attempt to break down this chunk of code that changes the bpm randomly.

```
.R.....
.j2Tab.R3.
.2Xb.2X0..
.R4..C2.Ra
D1...02P6.
*$bpm:065.
```

We're going to start from the bottom up, and read left to right. because the last line of code here is the code that sets the bpm. `*$BPM:065` is setting the bpm to 65.

`$BPM:` has 3 inlets to the right of it that are the tempo number, and 1 inlet to the left that is expecting `*` or a bang or trigger of some sort to trigger `$` which is the operator for the script.

The `D` above where `$` is looking for a bang sends a trigger at a delay of the frame rate, or a division of the bpm. The right inlet for `D` sets the division time, and the left inlet (not in use here) sets the multiplication time. If `D` is set to 1 it will send a bang out of the outlet below `D` on every frame, or every 16th step of your bpm. If it's set to 2 it will bang every other frame, or every 8th step, etc.

Above the right inlet for `D`, where it's expecting a number for a division of the frame right, I've placed `R`. `R` sends a Random number between the range of the left and right inlet. The left inlet is left empty to represent `0`, and the right inlet is set to `4`. Uppercase versions of these operators bang every frame, so `R` will output a random number between `0` and `4` every 16th step. You can use a lowercase `r` and bang the space above the `r` with an operator like `D` if you want it to only choose a random number at a certain frequency. In this section we were not getting a random 16th, 8th, 4th or second step division to trigger our bpm changes.

Now to our bpm number. Above the `6` for our bpm number is the operator `P`. `P` writes values for us sequentially eastward. On the left it has two inputs, the first (two spaces left) is a clock or counter source for `P` to tell it to move the position that it's writing forward or backwards. The second inlet on the left is the number of spaces it will move to write in. The inlet on the right is the value that will be written. Above the inlet for the counter I have the operator `C`. `C` is a clock that counts to the number in the right inlet at a division of the left inlet. If nothing is in the left inlet it will count every frame. Because `P` is only set to move `2` spaces, I set the counter of `C` to 2. Now every frame `P` will write a value to the position where `6` is and then the position where `5` is. *(note because orca is constantly changing, i've noted numbers that change dynamically in the code with a lighter font color for those letters)*. Above the left inlet of `P` I have a `R` set to the range of a (or `0-9`).

Here's where it gets tricky, but is part of the puzzle I was talking about. In the 3rd line down from the top:

```
.2Xb .2X0..
```

I am using the variable `X` to move or teleport a specific value to a different place on the x, y grid, (relative eastward or southward to where `X` is placed.) `X` recives to values on the left, the `X` and `Y` coordinates, and one on the right, the value being teleported.

```
.R.....
.j2Tab.R3.
```

.2Xb .2X0 . .

In the left inlet for **Y** I've placed the number **2**, because I want it to teleport my value two spaces down from the current position under **X**, where the **b** in bpm is. Above the inlet for the value of **X** is the operand **T**. **T** is a sequencer, Its two left inlets are the counter for where in the sequence its reading and the length of the sequence. The right inlets are the values for the sequence, depending on the length. In this code I set the length to two and the counter to a random number, that will step between the first or second value in my sequence. **bpm** also has another version of itself called **apm**, which animates or interpolates between two values at the frame rate, instead of immediately changing the bpm. I'm using my sequence to change the operator between **bpm** and **apm** randomly by changing the **b** and **a** with the values in my sequence.

The second **X** teleports a random value between **0** and **3** two spaces down from the outlet under **X**, to where the first number for the **bpm** is. I set the first number to a random range between **1** and **3** because the maximum **bpm** range is **300**, so the first number only needs to be random between **0** and **3** while the other two numbers can be random between **0** and **9**.

This chunk of code lets me change the bpm dynamically both fast and slow and with control over the range and timing of my patch.

---

### ***pearls:***

Orca is the main brain for the sequence, sending MIDI note and random tempo information to the operator plugin in ableton. Operator levels and decay times are modulated randomly at every MIDI note **Probee**. Reverb dry/wet is modulated randomly using **Probee**. The kick is being triggered by Orca, and sidechained to the synth.

---

### ***pleasantly drunk, with wings:***

Random pulses and wind from Shbobo Shnth are sent to resonate **Wingie**, who goes into the **Cocoquantus**, with different delays for left and right, creating pseudo reverb.

The Shnth is a weird thing. The Shnth is a handheld synth that has touch bars, buttons and two antennas called the corp. You can program these physical points of contact to perform different functions or play different sounds with the coding language *fish*. Its super fun to interact with and is a whole rabbit hole in terms of synthesis and coding in itself. It also can make sounds on its own without you needing to touch it or interact with the bars or corp, making it super fun to live code as well. The language is very simple and intuitive, the program uses a graphic interface but you can write, save and load code in text as seen below. I have a few videos on it on my youtube page if you want to learn more about the code, *fish*. Here's the code for *fish* for this patch:

```
{
(jump (major 1))
```

*The Fish Jumps between Soup*

*This line of code lets me store multiple fish patches on the shnth and jump between them with major button 1*

```
(srate (short (corpb) 44))
```

*This line of code allows me to control the sample rate of the audio coming out of fish with the second antenna on the back, short allows me to be more specific (16 bit) with the values i want to control the sample rate with*

```
(leak (left
(lights (dust (togo (dustc 5) (dustd 4) 20 2 4 6 3 60 1
70 18) (wheelb (corp) (wheelb (wheelc -115 -50) -86)
64)))
```

*This chunk of code sends random pulses with random rates out of the left channel of the shnth, the left corp has some influence on the rate of these pulses, but not complete control.*

```
(leakb (waveg (pulsec (square (corp) 54) 54) 1 (corp)
90) (barb))
```

*This chunk of code sends a square wave out of the left channel with the cutoff of a low pass filter controlled by the left corp*

```
(leakc (modo (hazeb (duste 12) 2 109 (togoh (dustg -61)
(dusth -53) 2 4 6 8 10 24 59) 89 64) (corp)) (barc)))
(wheelh 22 85))
```

*This chunk of code sends random undulating saw waves out of the left channel with the volume and rate influenced by the left corp and the third bar*

```

(leakb (right
(lights (dustb (togob (duste 2) (dustf 4) 2 8 19 35 2 4
100 1 7) (wheelc (corpb) (wheelc (wheeld -9 -1) -32)
64)))
(leakd (waveh (pulsed (square (corpb) 72) 20) 1 (corpb)
90) (corpb))
(leak (modo (hazec (dustf 6) 4 82 (togo (dust 108)
(dustb -71) 2 3 5 7 9 12 23 64) 119 64) (corpb))
(corp))) (wheel -104 79))

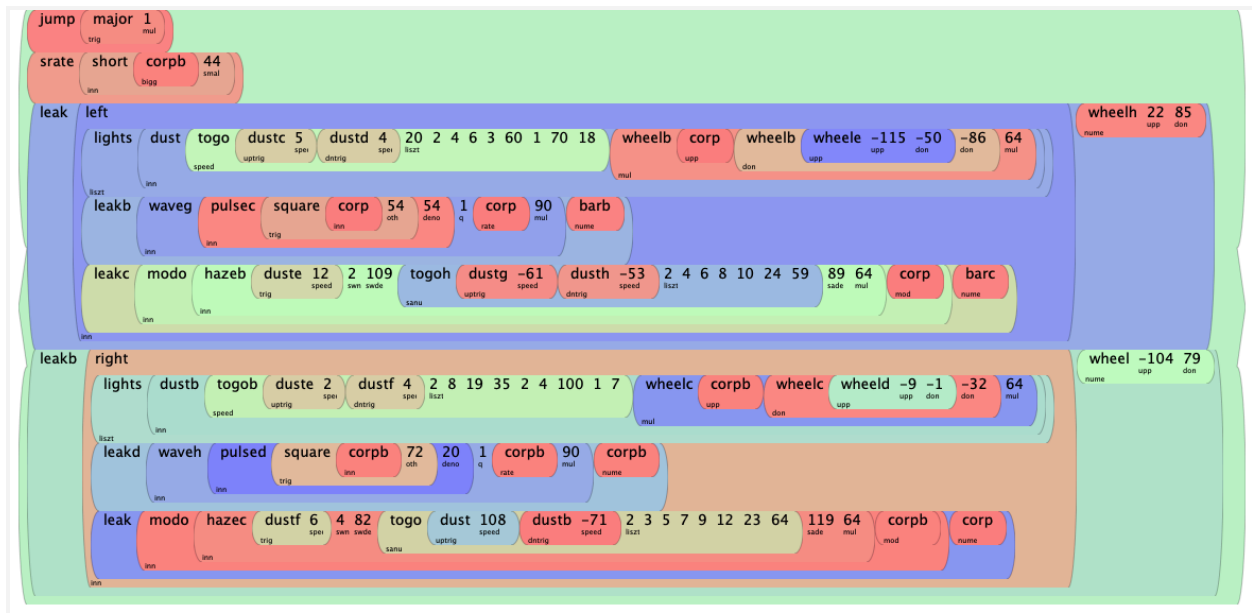
```

The rest of the code is copied again to come out of the right channel, with changes to certain parameters and using the right corp instead of the left.

```

}

```



### *swih aybz:*

In this ppool patch a clone of benjolin is the main sound source. It's randomly modulated and sent to a reverb. The reverb output is sent to 2 filters both set to notch/band reject and their cutoffs are modulated by two slow moving LFOs. These filters are sent directly to the mixer. The reverb was also sent to a spectral freeze module that froze the reverb and that is being sent to the mixer as well for a frozen drone of a tone from the reverb.



Ppool for me is what I always wanted MAXMSP to be. I dove into max a few times but couldn't, didn't really want to build my own granular synths from scratch. It was either too hard or I was too lazy or I didn't really know what I wanted. Ppool is a library for max that has a bunch of granular synths and delays that people made in max that was exactly what I was looking for out of maxmsp. That and the option to have as many output channels allowed me to process different ppool acts or modules separately in another program like Ableton, or create multichannel audio for spatial audio with 16 channels of granular, etc. ppool opened up a whole new world of playability and patching. I would often use it with a midi controller to control various parameters with physical midi sliders. Something I want to try is controlling ppool granular acts via midi from my modular synth. The Beauty.act in ppool is also a crazy multitap delay like no other. There is actually a clone of the cocoquantus in ppool that a user made as well, which I sometimes use. The spectral freeze act is also something unique to ppool that I don't use elsewhere, however the guys at Disting just developed a spectral freeze algorithm for the Disting EX.

---

***All of your technology will function properly this mercury retrograde:***

Double Knot is the main brain of this patch. It's set up to be two runglers by using one oscillator to send to the clock of the first shift register and the other oscillator to the date in of the shift register and the opposite oscillators configured for the second shift register. The voltage out of these oscillators are going to Teletype which is converting the voltage into MIDI notes. One of the on/off binary outs of the first shift register is triggering a script on teletype that sends that MIDI note to wingie. The output of double knot, which is a very short decal voltage pulse, as the oscillators are basically LFOs used for timing, is going into winige as an exciter, but we don't hear the pulses themselves until the end of the track. Other pulse outputs from the double knot are triggering an envelope for a kick and hi hat. The voltage from the second shift register is controlling the decay and timbre of the hi hat. On the ER301 sound computer there is a dub looper after the hi hat and kick, and a mixer with a grain stretch algorithm that's slightly pitched up. A random voltage from stages controls the speed of the grain stretch, which is pulled from the buffer of the dub looper. Vocals from my laptop are being sent over bluetooth to x113 monolith, which also goes into the buffer before the grain stretch.

Wingie goes into Cocoquantus which is set up as a simple left and right delay. The delay time is modulated slightly by pulses from the shift register of the double knot, pitching the delayed content upwards with pulses sometimes. The skip input of the delays is modulated by the voltage from the second shift register, skipping through the recorded content of the buffer. The flip input of the delays is modulated by an oscillator from the Quantussy section of the Cocoquantus, reversing the live delay content sometimes. The volume of the left and right in from the double knot is also

being modulated by a FMed oscillator from the Quantussy section of the Cocoquantus. There is a little bit of reverb on the Cocoquantus from the ER301.

The cocoquantus is also an essential part of my hardware setup. The two delays and modulation sections allow me to create simple delays/reverb but also crazy broken sampled tracks, drunk oscillators and runglers, slurry watery textures, and everything in between. The completely organic nature of the circuit design and lofi delay chips used for the sampling sections create such a unique instrument for audio processing that I truly cant find in any other hardware delay processor, nothing sounds like the coco. I find it always creates a story of its own out of the content I sample into it. I've always loved simple delay times, and although the coco is a type "granular synth" the idea of hard panning sound in the stereo field and the use of oscillators and chaos as the source for delays and timing makes the coco unique in the way it conversates with the sounds you sample into it. It has a huge spatial quality to it that creates these immersive worlds for me. I use the cocoquantus sometimes to create full tracks by sampling a vocal and playing around with the buffer. The dolby (noise gate/inverso noise gate) switches and envelope follower allow me to insert audio in between the delay effects, or into the delay buffer dry for an interesting effect. Sometimes I also use the oscillators in the quantussy to make a bassline or bass notes, or for audio rate modulation of the pitch of the delay buffer or skipping and flipping inlets for the delays.

## Glossary of esoteric terms:

**Benjolin:** The Benjolin is a chaos synth with two oscillators, a filter and a feedback system. This gives a very characteristic sound, controlled by a built-in sequencer that operates according to chaotic principles.

**Chloe:** *Mono vacuum tube/valve overdrive / multimode filter / lofi pt2399 delay with filtering / low pass gate eurorack module*

**Coalescence:** Coalescence is a Max For Live instrument. It is a concatenative multi-sampler that uses machine learning (SOM neural network) to organize similar sample slices into clusters based on a chosen spectral feature. There are three playback modes that take advantage of these clusters in various ways: Point, Rings, and Paths. Additionally, external audio input can be routed for modulations and to control what audio slices play based on its similarity. All of these combined with a robust modulation system makes Coalescence one beast of a compact sampler for many different situations!

**Cocoquantus:** (Peter Blasser) The CocoQuantus is a device by Ciat-Lonbarde composed of two Coco modules (8 bit digital delays) and a central "Quantussy" analog brain (nabra).

**CV (Control Voltage):** Sometimes abbreviated CV, Control Voltage is a DC electrical signal used to manipulate the values of components in analog circuits. Control voltages are used in numerous ways in many different types of electronic circuits for all sorts of purposes. A few examples germane to music technology: If you send a specific electrical voltage to a module of an analog synthesizer (such as an ADSR envelope generator), you can specify what you want the module to do (perhaps lengthen the decay time). In a mixer and other audio circuit that uses a voltage-controlled amplifier (VCA), a DC voltage can be used to set the gain of the VCA. This is applicable for things like compressors and gates, where the DC voltage may either be a signal proportional to the audio level the devices are acting upon, or could be from some other source. But it also applies to VCA-style mixing automation systems, where a control voltage is set by the user (via some interface, but usually faders) to determine and subsequently log the desired levels in the mixer at specific points in time.

**Disting EX:** The disting EX is a multifunction Eurorack module

**Double Knot:** The Double Knot v3 is a small generative synthesizer with two voices and corresponding sequencers. This instrument has a simple architecture which is capable of a wide variety of sounds, including rhythms and drum patterns or textural and grating noise sounds. The interface is made up of knobs, switches, and banana plugs for patching. The two voices of this device each consist of a triangle/square wave oscillator, a shift register sequencer, an envelope, and a voltage controlled amplifier. A clock sets the tempo of the shift registers. The voices of the Double Knot are pre-wired, in a linear chain of functional blocks. The patch points are available to

route modulations, influencing the simple linear voices. The game of this device is to investigate the relationships between the elements of the synth using the knobs, switches and patch-points.

**Er301 Sound Computer:** The ER-301 is a voltage-controllable canvas for digital signal processing algorithms. One of the many characteristics that attracted me to modular synthesizers is that they enabled me to make the kind of music that I love away from the computer. However, not all digital audio tools can be elegantly and efficiently realized in a dedicated hardware form. So, I either had to deny myself those tools, or go to the computer.

**FM (Frequency Modulation):** Frequency Modulation (FM) is the encoding of information in a carrier wave by changing the instantaneous frequency of the wave. FM technology is widely used in the fields of computing, telecommunications, and signal processing.

**Granular Synthesis:** Granular synthesis is a sound synthesis method that operates on the microsecond time scale. It is based on the same principle as sampling. However, the samples are split into small pieces of around 1 to 100 ms in duration. These small pieces are called grains. Multiple grains may be layered on top of each other, and may play at different speeds, phases, volume, and frequency, among other parameters. At low speeds of playback, the result is a kind of soundscape, often described as a cloud, that is manipulatable in a manner unlike that for natural sound sampling or other synthesis techniques. At high speeds, the result is heard as a note or notes of a novel timbre. By varying the waveform, envelope, duration, spatial position, and density of the grains, many different sounds can be produced.

**Max For Live:** Max for Live is a platform to build your own instruments and effects, tools for live performance and visuals, and much more.

**Meng Qi Wingie2:** Wingie2 is a handheld stereo resonator with onboard microphones that doubles as a development platform. It allows you to interact with and enrich sounds from instruments and vocals as well as the sonic environment around you.

**Midi:** Musical Instrument Digital Interface (MIDI) is a standard to transmit and store music, originally designed for digital music synthesizers. MIDI does not transmit recorded sounds. Instead, it includes musical notes, timings and pitch information, which the receiving device uses to play music from its own sound library.

**Modulation:** Modulation is the application of AC control voltage from a VCO, LFO (Low Frequency Oscillator) or noise source to other synthesis parameters, such as frequency, filter c.o.f., filter Q amount, amplitude, or pulse width. The modulating module providing the oscillating control voltage is known as the modulator.

**Multisample:** A group of samples organized in a musically relevant way. For example, most piano samples are actually made up of many different samples. There are usually at least 8 (sometimes

many more) different samples across the keyboard, and in many cases there are samples of different velocities and so on.

**Orca:** Orca is a two-dimensional esoteric programming language in which every letter of the alphabet is an operator, where lowercase letters operate on bang, uppercase letters operate each frame. Orca is an esoteric programming language designed to quickly create procedural sequencers, in which every letter of the alphabet is an operation, where lowercase letters operate on bang, uppercase letters operate each frame. This application is not a synthesizer, but a livecoding environment capable of sending MIDI, OSC & UDP to your audio/visual interfaces, like Ableton, Renoise, VCV Rack or SuperCollider.

**Patch:** An system, configuration or setting usually for a modular synthesizer that determines its sound or the way it behaves. Usually physically with patching cables.

**ppool:** is a digital signal processing (DSP) environment for Max/MSP. It's a versatile, fully customizable toolkit that facilitates audio manipulation, granular synthesis, live performance and improvisation, modular networking, ambisonics and much more.

**Probee:** Probee is a free random generator triggered by MIDI notes. Everytime a note is played either via the computer keyboard, a MIDI clip in Live or an external keyboard, Probee will create 6 different random values.

**Quantussy:** (Peter Blasser) The Quantussy is a five (5) petaled flower, each petal with oscirator, that creates a rhythm, that triggers quantizations of the movements of the other four oscirators. The Quantussy is thus both based on five (5), but four(4)... and its name also refers to Quantum Physics through its unique "double quantization" of angular momentum. Running at low frequencies, the Quantussy is intended to provide ample modulations for the various functions of the COCOs, through inputs "FLIP", "SKIP", and "SPAF", or "Speed Affect". The Quantussy petals can also run at audio frequencies, as well as "Balcium" frequencies, depending on a toggle switch.

**Rollz5:** (Peter Blasser) Rollz5 creates organic rhythms out of geometrical forms. A future direction is to create electronic sound devices based on the platonic solids and other 3-D topographies. I intended to confront the notion of "drum machine", which implies the sterile regimentation of time, and transform it into a collection of organic flows generated by geometrical forms. These forms and their accompanying filters can be switched, wired, or touched; the final manifestations range from a small preset switch-box, a squeezable spike-dome, or a traditional modular.

**Rungler:** The purpose of the rungler is to create short stepped patterns of variable length and speed. One could categorize the circuit somewhere halfway between a plain S&H and a shift register-based pseudorandom generator. It needs two frequency sources to work and basically creates a complex interference pattern that can be fed back into the frequency parameters of the driving oscillators to create an unlimited amount of havoc.

*Shift Register:* A shift register is a type of digital circuit using a cascade of flip-flops where the output of one flip-flop is connected to the input of the next. They share a single clock signal, which causes the data stored in the system to shift from one location to the next.

*Teletype:* Algorithmic ecosystem: a dynamic, musical event triggering platform for eurorack modular.

*Conceptually a lot of what has been informing my practice recently has been: generic caucasian activity, Karens saying the n word, Alpha male learn how to get girls using dark psychology, ASMR, Mercury retrograde, white girl drinks (iced chai latte with oat milk, pumpkin spice latte, etc, etc), being really drunk, bad bitches, thirst trap content.*

*And in all seriousness, the direct dichotomy, in both high and low frequencies, as sound, separate and connected to sadnoise, may influence my design philosophy in my architectural process, I ask questions like:*

*How can we properly acknowledge the displacement and destruction of indigenous land as the gentrification and backwards evolution of music and culture in the underground BIPOC communities in nyc. How can we design a space that bridges the gap between the two cultures and creates a welcoming space for new experimental sonic ritual practice. What are natural ways these interactions can form and what will aid both cultures during the design process. What do these communities need in order to feel welcome both physically and sonically.*

*And it is this conversation between the brain rot of the internet and the reality of sound as a ritual practice in all its nature and being that creates the space inbetween the polar opposites of these two spaces, that is ever so wide for me to fill with sound. Worlds that have nothing and everything to do to each other, and next to each other, in its campness, in satire of dada.*

*Renee Loran*

*“Things are defined by what they are not.”*

*Cliché conversation with kitsch*

*Connects to the idea of sadnoise being anti-music and nonlinear*

*Not for people - sound for space and a particular time, non-reproductive*

*Her definition of opposites as negotiation speaks to the inevitable of interaction, improvisation and performance*

*No matter what sound I provide for a space my work is always an interaction with the physical space / or the internet*

*The aesthetic of sad noise is based on this incompatibility, an expression of all things incompatible*

*In conversation with this theory, sadnoise is redefining beauty as not incompatibility but non linear realization that truly perfect conversations have polar contrast at the most interesting intersections, in my case the creation and manipulation of existing sonic content into an archive of the overlap of everything in its absurdity, a maximalist aesthetic of the making things apparent, so apparent so that they are buried in their density.*

In the lens of Dutton I use qualities of work that would be considered kitsch as my ephemeral nature of my work or the erasure of its existence in time, a way to keep my work anti-composed. His definition of it as a pseudo-art already speaks to the nature of sad noise being anti-music and pro space. A way of expression technically not by design in a. Composition sense by using the aesthetic qualities of different elements to create an aesthetic, a culmination/compilation of the sorts

Using memes at the end of my sets to inject familiar content/vocals in a random way into an unfamiliar space, a way of grounding after the ritual practice

In contrast with Dutton, my work isn't "bad music on purpose" but rather a very high fidelity reproduction of my personal interaction with myself and the aesthetics of content I care about and dislike, as well as how those can interact with the space i'm performing for or against.

Conversation between academia and underground

```

.....
.....R9.....R9.....
.....R3.....R7.....
.....0.....4.....
.....Ce.....CeJ.....Re.....CeJ.....
.....8XE.....8X0.....2XE.....8X4.....
.....;00c.....E.;16c.....
.....;07a.....*;20a.....
.....*;01f.....E.....;30f.....
.....E;04e.....;40e.....
.....E.;0dg.....E;51g.....
.....E.;04a.....;62a.....
.....E.;02d.....E.;70d.....
.....E.....;02g.....;8yg.....
.....E.....;00e.....;94e.....
.....;01c.....;12c.....
.....;03g.....E.;21g.....
.....;0ee.....;31e.....
.....;0af.....E.;40f.....
.....;01a.....;5ta.....
......Ra.....
.....3X0.....R.....
......C.....Ra.....v2Tab.....Ra.....
.....4.....1X4.....b.....1X3.....
......D1J.....Ra.....D.J.....R3Ra.....
.....*;4env0440.....Sbpm:135.....
.....

```







## Album Artwork by Hannah Rose Albinus

This drawing makes me feel: Soft blankets, warm sunlight, ripples in water, hot days, falling asleep, a weird dream, sticking your hand in deep moss, the feeling of eggs on your fingers, really hot water on the backs of your hands, stepping on the end of a cable, hitting your funny bone, the pleasant feeling of being bored, walking home in the rain, sitting under a big tree, listening to insects, lightheadedness, dissociation, bright lights, shapes in complete darkness, wind, the smell of hot coffee, robbing an apple tree in the middle of the night, having a crush on someone.

Femi Shonuga-Fleming  
04012024

