Osmo Lindeman's

DICO

by Erkki Kurenniemi

Introduction

DICO (also DIGO or LDCO) is a synthesizer Finnish media art pioneer Erkki Kurenniemi designed and built for composer Osmo Lindeman in 1969. As the name suggests it is a digitally controlled oscillator. DICO is easily identified on Lindeman's electronic music catalogue.

DICO has a digital memory that is used to store a 12 steps long sequence. For each step, the internal state of the instrument is stored using 10 bits: Four bits for note, three bits for octave, two bits for note length and one bit for selecting the output channel.

DIR-DICO is an eurorack remake of Erkki Kurenniemi's DICO designed by Jari Suominen as part of 'DIMI is Reborn' project. The original user interface has been copied as closely as possible. As the main control panel of the original DICO is built for a 3U rack, the panel is mostly a 1:1 copy of the original. Width of the panel has been rounded up to the closest HP unit (40HP). The jack panel and optional MIDI port panel are placed on the right side of the control panel (both 4HP).

There are two changes to the control panel. First one is the addition of indicator LED to show when the sequencer is stepping. Other one is the switch in the middle. In the original DICO, the switch has two positions: run and edit. Edit mode allows using buttons to advance and start/stop the sequencer.

In DIR-DICO the switch has three positions. Down position is the original edit-mode. Middle position is used for tweaking instruments settings. The position up is the slave mode, where either MIDI clock or incoming trig signals will control the sequencer. In all positions the start and the step button works as expected.

Output and input section differ from the original. Also DIR-DICO adds mode to tweak additional parameters. This mode is accessed by setting the switch on the panel to the middle position (which does not exist in original DICO). You can keep the sequencer running while entering the settings-mode and while you are editing the parameters.

Controls

- **Step/Stop button:** This button will advance the sequencer to the next step when the sequencer is not running. If pressed when the sequencer is running, the sequencer will stop.
- **Start button:** This will start the sequencer.
- **Mode switch:**
    - Original DICO[1]:
        - Position up: play mode, sequencer will be running, step/start buttons disabled.

[1] Also DIR DICO built for Helsinki University Museum Liekki

        - Position down: edit mode, when accessed, sequencer will stop, step/start buttons enabled.
    - DIR-DICO:
        - Position up: slave mode, either MIDI beat clock or external CV pulse will drive the sequencer[2].
        - Position middle: setting mode, see DICO settings.
        - Position down: master mode, internal clock will be driving the sequencer.
- **Tempo knob:** Sets the tempo range from roughly 1 step per second to 200 steps per second.
- **Pitch knob:** When turned fully clockwise, DICO will be in tune. Turning this knob down anti-clockwise will starve the oscillators, causing pitch to drop. Note that the pitch of four analog oscillator circuits will drop different amounts causing intervals between them to fall out of tune with each other.

Hardware inputs / outputs

Patch panel from top to bottom:

1. Audio channel #1 out (CH=1)
2. Audio channel #2 out (CH=0)
3. Trig/gate #1 out
4. Trig/gate #2 out
5. Trig/gate #3 out
6. CV out
7. Trig/gate in #1
8. Trig/gate in #2

- MIDI in and out through header and ribbon cable. Should be compatible with Expert Sleepers MIDI breakout (not tested).

DICO memory

- Original DICO uses 10 bits of data per step:
    - 4 bits for note value (“blue bits”) D, C, B, A
    - 3 bits for octave range D1, D2, D3
    - 2 bits for note length DB, DA:
        - 00 = Eighth
        - 10 = Quarter
        - 01 = Half
        - 11 = Whole
    - 1 bit for output channel CH:
        - 1 = Channel #1
        - 0 = Channel #2
- On the panel interface there are two more bits that are not used. Could be that the idea was to allow 8 outputs (that would be logical compared to design solutions of Sähkökvartetti and Andromatic), but for some reason only implemented two.

[2] See setting opcodes 0100, 1101 and 1110.

- As DIR-DICO firmware is using 8-bit uC, there is actually 16 bits of memory per step. Although the interface still only has 12 bits (of which only 10 bits are used).

Memory mapping

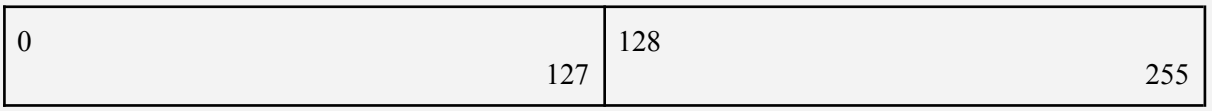
The memory of the original DICO can contain 12 steps. On DIR-DICO the memory is expanded and can potentially contain 256 steps. The memory allocation is following.

Memory will always contain 256 steps (steps from 0 to 255) of data. This memory space is divided into blocks depending how many sequences will be stored (setting 0111). Sequences will be filled starting from the beginning of the block and don't need to fill one block fully if sequence length is shorter than block size. If a sequence is longer than one block, it can read/write from the memory space of different blocks. It is up to the user to set sequence length correctly.

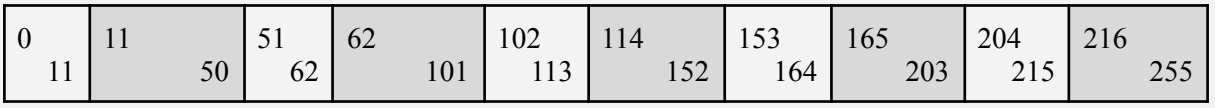
Data in memory is not rearranged when block count is changed. Changing block count will only change how data in memory is interpreted. Step counter of all sequences will start from 0 and run until the last step. The logic of how the space is allocated for one sequence is illustrated in the examples below:

One sequence, max length 256 steps without overlap.

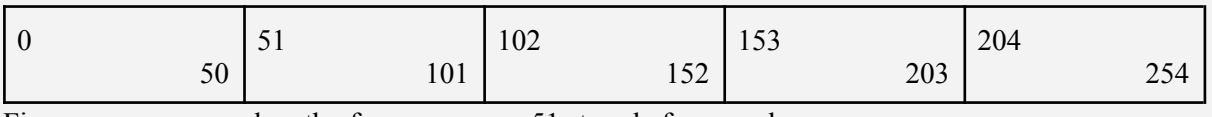
Stock DICO, one 12 step sequence, remains of the memory unused.



Two sequences, max length of one sequence 128 steps before overlap.



Five 12 step sequences, sections between sequences unused.



Five sequences, max length of one sequence 51 steps before overlap.

DICO settings

DICO settings can be accessed by setting the mode toggle switch to the middle position. Sequencer will stay fully operational when in toggle mode which makes it possible to tweak the settings when the sequencer is running[3]. Settings will be stored to EEPROM memory when exiting the edit mode.

Settings

Settings have been divided into two groups. First group are settings that might be useful to tweak during performance. The second group are settings you probably don't want (especially accidentally) tweaked during performance. Some settings have been combined under the same opcode to make space for new opcodes.

opcode[4][5]	*description*	*data*[6]	
0000	Current step.	0 - The last step of the sequence	r/w
0001	Last step of the sequence.	0 - 255 **(default 11 B0000 1011)**	r/w
0010	Clock pulses per step[7]	1 - 255 *(default 12)* In order for step to equal: B0001 1000 = 24 (4th note) **B0000 1100 = 12 (8th note)** B0000 0110 = 6 (16th note) B0001 0000 = 16 (8th triplet) Original DICO would use[8]: B0000 1000 = 8 *Pulses per quarter (PPQ) is always 24.*	r/w
0011	Step increment	0 - 255. Default **1**.[9]	r/w
0100	Tempo pot position display.	0 - 255	**r**
0101	Indicator MSB Tempo clock LSB	Indicator see 1000 **(default step 0101)** Tempo Clock: B0000 Hz (0-255) B0001 BPM (367-262)	r/w

[3] Some parameters could be useful to tweak live.
[4] Blue bits
[5] **Bolded** opcodes/values are at least partially implemented.
[6] Yellow bits
[7] MIDI sync clock thingie. Each step is divided into multiple pulses. Gate signals will be generated with pulse ticks.
[8] This number is not optimal for external synchronisation. However, only pulse counts divisible by eight will generate exact gate signals of the original DICO.
[9] Will fold after the last step. Folding can be used for both inverted and pseudo random playback.

0110	Current sequence	0 - The last sequence	
0111	Last sequence	0 - 255[10]	
1000	CV Gate output #1 MSB CV Gate output #2 LSB	**B0000 Off** **B0001 Gate** **B0100 Channel** **B0101 Step clock** **B0110 Pulse clock** **B0111 Reset** **B1000 Run/Stop** **B1100 Trig In #1** **B1101 Trig In #2** Gate Channel #1 Gate Channel #2 B0011 B0010	r/w
1001	CV Gate output #3 MSB CV output #4 LSB	Gate output see 1000 CV output: **B0000 Off** **B0001 Sequencer step (0-255)** **B0010 Note (blue bits)** B0011 Note (channel #1) **B0100 Note two octaves** B0011 Channel 1 pitch B0101 Note (blue bits) B0110 Yellow bits? **B0111 Tempo pot (0-255)** Normalised step? Blue bits as one byte? Two LSB bits? (normalised somehow)?	r/w
1010	MIDI input	**MIDI channel 0 - 15 Bxxxx cccc** **(0 is the 1st MIDI channel).** **Bxxx1 xxxx** OMNI mode Bxx0x xxxx Step after note received Bx0xx xxxx Enable note data in **B1xxx xxxx Beat clock in enable**	r/w
1011	MIDI output	**MIDI channel 0 - 15 Bxxxx cccc** **(0 is the 1st MIDI channel).** MIDI thru mode: **Bxx00 xxxx** Off Bxx01 xxxx Full	r/w

[10] As memory size is limited, the amount of sequences in the memory can and will limit the maximum size of one sequence.

		Bxx10 xxxx Only input channel Bxx11 xxxx All but input channel Bx0xx xxxx Enable note data out **B1xxx xxxx Beat clock out enable**	
1100	Gate/Trig input # 1 MSB Gate/Trig input # 2 LSB	**B0001** Step clock **B0010** Pulse clock **B0011** Next sequence B0101 Gate (“ring modulator”) **B0100** Reset **B1000** Run/Stop[11] **B1001** Start/Stop[12] **B1110** Vibrato[13] downwards **B1111** Vibrato upwards	r/w
1101	Random flags (1 = enable)	**Bxxxx xxx0** Reset on start **Bxxxx xx0x** Reset on stop **Bxxxx x0xx** One shot **Bxxxx 0xxx** Last step[14] **Bxxx1 xxxx** Enable gate **Bxx1x xxxx** Enable channel[15] **Bx1xx xxxx** Gate open on stop B1xxx xxxx Iterate sequences on end	r/w
1110	Divider B ratios	MSB (0-15) when bit B is on LSB (0-15) when bit B is off (default 4th 4/3 **B0100 0011**)	r/w
1111	Divider A ratios	See 1110 (default 3th 2/3 **B0010 0011**)	r/w

Synchronisation

Depending on the position of the *mode switch* either internal clock generator or external sync signal is used to drive the sequencer. If external source is selected, either suitable input signals on CV inputs (1101 and 1110) or MIDI beat clock (0100) input has to be enabled[16].

[11] Whether the sequencer will be rewinded back to step 0 on START (default DIN sync behaviour) depends on the flag setting ‘Reset on Start’. Run/Stop-signal is held HIGH when the sequencer is running, otherwise it is held LOW.
[12] This signal is a short pulse during both start and stop, otherwise the signal is held LOW. Note that quite commonly run/stop-signal is called start/stop. Start/stop signal is available for instance on CSQ-600 Sequencer.
[13] Vibrato is implemented within the divider routine. This makes it discrete binary shift up or down. Vibrato also gets wider, higher the pitch of the oscillator is. Further, when the vibrato bit is on, the scale enters microtonal dimension.
[14] Using step increments other than 1 might require the last step of the sequence to be interpreted differently. 0 - last step is the last step of the sequence (as you would normally consider). 1 - The last step is the step before the first step of the sequence.
[15] Output will always be the same as bit CH off, which would be channel #2.
[16] Both external synchronisation sources can be used concurrently, for whatever the reasoning to do this might be.

Input signals

All signals but the pulse clock can be used in both, master and slave modes.

The most common synchronisation method would be to use an external pulse clock signal with either run/stop or reset signal.

- Run/Stop
    - When held HIGH, the sequencer will be running. Meaning either internal pulse clock, external pulse signals or incoming MIDI beat clock will be used to drive the sequencer.
    - Playback will continue on the next pulse after RUN/STOP rising edge.
    - Similarly playback will stop and incoming pulses will be ignored after falling edge.
    - Works like MIDI Continue, does not rewind the sequencer to the first step by default. Rewinding can be forced with setting flag (0101) B0000 000X.
    - This signal will give similar results than what using start and step/stop buttons will do.
- Reset
    - This will reset the sequencer to the first step. Next pulse will be the downbeat of the first step (0).
    - Works the same as the MIDI Start command. Although RESET can be used when the sequencer is running.
    - Holding step/stop and start buttons down together will generate this signal internally.
- Pulse clock
    - Will advance the sequencer by one tick.
    - Similar behaviour as receiving MIDI Beat Clock messages.
    - Will be ignored unless the *mode switch* is on the position **up**.
- Step clock
    - Will advance the sequencer to the downbeat of the next tick.
    - In order to generate gate signals either internal or external pulse clock is also required. As the gate is always open on downbeat, if step clock input is used without pulse signals, gate control bits are bypassed and the gate is always open.
    - If step clock input is selected for either input channels, pulse clock signals will not advance the sequencer to the next step after pulse count per step has been reached.

Note mapping

On original DICO manual, C4 / 261Hz correspond to bit combination B0000 110 (XX X XX). This also maps to midi note 60.

Counting up from here, B0000 000 would be C7 / 2093Hz / MIDI# 96.
Counting downwards, B0000 111 would be C0 / MIDI# 12

With standard tuning / divider settings, lowest note would be B0010 / G-1 / MIDI# 7.
With standard tuning / divider settings, highest note would be B1101 / E8 / MIDI# 112.

With standard tuning / both dividers set to 1, the highest note would be H10 / MIDI# 143 (which of course does not exist)?
With standard tuning / both dividers set to 1, the lowest note would be D4?

In mk2, CV out (#4) will give out voltages from 0V-10.6V (C-1 - G#9). Future revision probably will be implemented according to Doepfer specs, unipolar, 0V to C3 (allowing easier tuning etc).

Calibration

Tuning the oscillator

Oscillator is tuned with four trimmers at the very left side of the panel. Set bits B and A to 0. Set octave bits to 110. Turn the pitch potentiometer fully clockwise. Each trimmer corresponds to one combination of the bits D and C. Starting from the top using traditional tuning used on original DICO:

- 00, tune to C4 261.63Hz
- 01, tune to D#4 311.13Hz
- 10, tune to F#4 369.99Hz
- 11, tune to A4 440Hz

Calibrating the gate

The gate offset is adjusted with two trimmers near the pitch knob. Easiest method is to create a loop with all steps on the channel that is adjusted. Then make all notes staccato (DB/DA bits off). Start the sequencer and turn the speed knob to an extremely fast tempo. Now look for a sweet spot where the low frequency brum is smallest. Sweet spot should be near the center position of the trimmer.

Calibrating CV output

CV calibration trimmer is located between PCB and panel and is facing down. You need to pull the module out from the frame in order to access it.

Calibrating pitch starve

There is a third trimmer behind the PCB near the gate trimmers. This is used to expand the range of the pitch knob. Typically you want to set it so that all oscillators (bits D and C) will only stop oscillating when the pitch knob is turned to CCW end. When tuned traditionally, DC setting 11 will select the oscillator that will stop oscillating last.