

# Gray Steps

Version 2024-02-20



**Gray Steps** is neither a real CV sequencer nor a trigger sequencer. Gray Steps offers functions of both stepping devices.

Main features:

- Steps count can be set from 1 to 256.
- Step trigger by CV or push button.
- Single or cyclic step sequence.
- Restart by CV, push button and automatically.
- Step direction can be reversed by CV, by toggle button and automatically.
- Digital outputs for new cycle start and Gray Code bits.
- Analog outputs for step number, code number and code value.

## Table of content

Gray Steps.....	1
1 Basics.....	3
1.1 (Standard) Gray Code.....	3
1.2 Balanced Gray Code.....	4
1.3 Name conventions in this document.....	4
2 Gray Steps' main functional blocks.....	4
2.1 Code generator.....	4
2.2 Step control.....	4
2.3 Restart control.....	5
2.4 Reverse control.....	5
2.5 Cycle control.....	6
2.6 Analog outputs.....	6
2.7 Digital outputs.....	6
2.7.1 New cycle output.....	6
2.7.2 Binary code.....	7
3 Examples.....	8
3.1 Start# + Count < Code number, forward.....	8
3.2 Start# + Count < Code number, backward.....	8
3.3 Start# + Count > Code number, forward.....	9
3.4 Start# + Count > Code number, backward.....	9
4 Controls and connectors.....	10
4.1 Code generator.....	10
4.2 Code index range.....	10
4.3 Step.....	11
4.4 Restart control.....	11
4.5 Reverse control.....	12
4.6 New cycle output.....	13
4.7 Analog outputs.....	13
4.8 Binary code outputs.....	14
5 Oscillograms.....	15
5.1 Analog outputs.....	15
5.2 Binary outputs.....	19
6 Links.....	20

# 1 Basics

## 1.1 (Standard) Gray Code

In 1953 physicist Frank Gray has got a patent on his invention of a continual binary code. Goal of this code was error reduction with mechanical switches. While in regular binary code several bits can change from one value to it's next at same time, in Gray Code just one bit changes between two neighbored values.

Though today electronic part replace mechanical switches, Gray Code is still used abroad modern devices, for example in absolute rotation encoders. Signal transmission systems as DBV-T and cable-TV provide automatic error correction with help of Gray Code.

Here both codes are shown. Changed bits are marked bold.

	regular binary code	Gray Code	decimal value
bit value:	8 4 2 1	8 4 2 1	
decimal number			
0	0 0 0 0	0 0 0 0	0
1	0 0 0 <b>1</b>	0 0 0 <b>1</b>	1
2	0 0 <b>1</b> 0	0 0 <b>1</b> 1	3
3	0 0 1 <b>1</b>	0 0 1 <b>0</b>	2
4	0 <b>1</b> 0 0	0 <b>1</b> 1 0	6
5	0 1 0 <b>1</b>	0 1 1 <b>1</b>	7
6	0 1 <b>1</b> 0	0 1 <b>0</b> 1	5
7	0 1 1 <b>1</b>	0 1 0 <b>0</b>	4
8	<b>1</b> 0 0 0	<b>1</b> 1 0 0	12
9	1 0 0 <b>1</b>	1 1 0 <b>1</b>	13
10	1 0 <b>1</b> 0	1 1 <b>1</b> 1	15
11	1 0 1 <b>1</b>	1 1 1 <b>0</b>	14
12	1 <b>1</b> 0 0	1 <b>0</b> 1 0	10
13	1 1 0 <b>1</b>	1 0 1 <b>1</b>	11
14	1 1 <b>1</b> 0	1 0 <b>0</b> 1	9
15	1 1 1 <b>1</b>	1 0 0 <b>0</b>	8
0	<b>0</b> 0 0 0	<b>0</b> 0 0 0	0

## 1.2 Balanced Gray Code

In the list above you can see, that in *standard Gray Code* lower bits change much more frequent than higher bits. In a *balanced Gray Code* that frequency is almost the same for each bit. There are rather complicated mathematical algorithms to create balanced codes for a distinct number of bits.

## 1.3 Name conventions in this document

In order to prevent you from getting confused, meaning of used terms will be explained here.

#	the index or position of an element
analog output	an output jack, that provides a variable CV
code	a distinct number, that is content of a Gray Code set
count	total number of something
cycle	execution of all steps until restart
digital output	an output jack, which provides either 0 or 5 volts
gate	a CV pulse with a distinct duration, that is used to represent an actual state ON or OFF
reverse	stepping backwards
step	a momentary sequencer state
trigger	a very short voltage pulse, that is used to initiate an action; duration normally one sample (= 1/48.000 or 1/96.000 sec)

## 2 Gray Steps' main functional blocks

### 2.1 Code generator

In this module *standard Gray Codes* are generated with a simple algorithm. This algorithm is usable in code generator *bits* and *codes* modes. For *balanced* mode a list with 256 fixed codes is stored in the module.

Though a Gray Code with  $b$  bits can contain less than  $2^b$  codes, a code list must contain an even number of codes in order match the Gray Code rule.

A code list is reproducible. Each time you generate a code list, it will contain same codes in same order, when total number of codes is equal.

### 2.2 Step control

Step control has to manage step operation, including counting steps forwards or backwards, stopping at cycle end, automatic restart.

## 2.3 Restart control


This control prepares module for a cycle start at next recognized step. A restart may be initiated anytime by pressing the *restart push button* or a *CV trigger pulse*.

Please note: When *start code #* or *code count* was changed to very different values within a rather short interval at rather high step frequency, **Gray Steps** may stop stepping. That is normal behavior. After getting restarted, Module will continue normal work.

Two controls define how restart will work:

Automatic restart button 

- When it is toggled OFF, Gray Steps runs for one cycle and stops it's operation after last step (= count).
- When it is ON, an automatic restart will happen at next step pulse and start next cycle.

Restart mode slide switch  defines sequencer state after a restart appeared.

- "0": After a restart sequencer stays on a virtual step #0 until first step signal is received. So even very first cycle consists of complete *count* steps.
- "1": A restart pulse sets sequencer counter to step #1. This is the way, many sequencers react on a reset signal. Result is that very first cycle will be finished already after *count* - 1 step pulses.

Please note: When actual step direction is reverse, there will be "(count)" instead of "1", because cycle will begin with last step #.

## 2.4 Reverse control

This module block has to manage step direction. There are three signals, that can reverse actual step direction:

- Reverse toggle button
- Reverse CV input
- Automatic reverse (*see cycle control*)

Each of these signals can force a direction reverse at same time. So step direction will be forwards, when number of reverse signals is even and backwards, when number of reverse signals is odd.

## 2.5 Cycle control

Cycle control has to detect first step of a cycle.

- When stepping forward, step # 1 initiates a cycle start.
- When stepping backward, step # (count) initiates a cycle start.

At each cycle start cycle control sends a signal to *reverse control* and to digital *new cycle output*.

## 2.6 Analog outputs

There are analog outputs:

- step #
- code #
- decimal code

As default these outputs deliver a normalized CV from 0 to +5 volt. This means 0 volt at minimum value and +5 volts at maximum value.

Knobs allow CV scaling within a wide range.

- Gain knobs (yellow): -10 .. +10, default: +5
- Offset knobs (orange): -10 .. +10, default: 0

A double click on a knob will set it to it's default value.

While on step #0, analog outputs are 0, independent of knob settings.

## 2.7 Digital outputs

### 2.7.1 New cycle output

It sends a signal, when a new cycle starts. That is at first step in forward direction forward or last step in reverse direction. Kind of signal depends on gate/trig slide switch position.

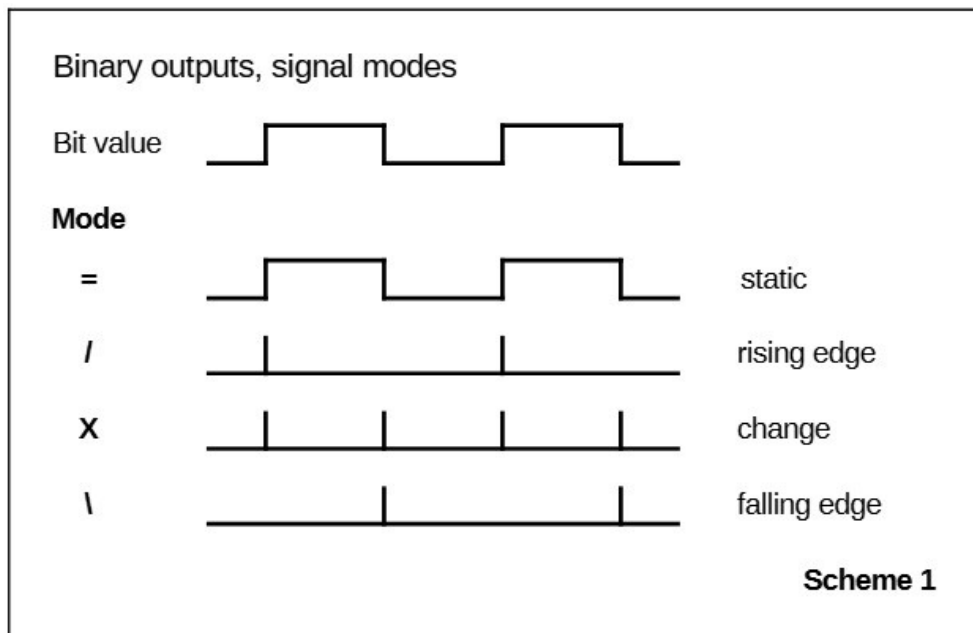
- "gate": Output is ON (+5 volt) during whole first cycle step.
- "trig": Output sends a very short 5 volt trigger pulse at beginning of first cycle step. Duration can be set from 1 to 1,000 scans.

## 2.7.2 Binary code

Here you can see a group of eight output jacks, one for each bit. These outputs are ordered from bit 7 (left) to bit 0 (right).

For each output:

- Area around output jack is colored according actual state:
  - dark grey: This bit is out of defined bit width and not active.
  - light grey: The bit is within defined bit width but actually invalid, because sequencer is in state 0.
  - blue: Actual bit value is OFF
  - light yellow Actual bit value is ON
- Small label above output jack displays actual signal mode. (see Scheme 1)



For all outputs:

Duration of a trigger pulse (for modes /, X, \) can be set to distinct number of sample counts.

Scans	Duration @ 48,000 sps (samples per second)
1	0.02 msec
2	0.04 msec
5	0.10 msec
10	0.21 msec
20	0.42 msec
50	1.04 msec
100	2.08 msec
200	4.17 msec
500	10.42 msec
1000	20.83 msec

Trigger pulse length is the same for all digital outputs.

# 3 Examples

## 3.1 Start# + Count < Code number, forward

This is the simplest case. Code count is not higher than number of codes from start# to last code in code list. First step of a cycle uses code start#.

(Scheme 2)

Relationship between parameters, start # + count < codes, reverse = off

bits	4																		
codes	16																		
count	8																		
start #	5																		
reverse	off																		
code #	-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
code	-	0	1	3	2	6	7	5	4	12	13	15	14	10	11	9	8		
dec value	(0)	0	1	3	2	6	7	5	4	12	13	15	14	10	11	9	8		
step #	0					1	2	3	4	5	6	7	8						
current code #	7																		
current dec value	5																		
current step	3																		
cycle start at step #	1																		

Scheme 2

## 3.2 Start# + Count < Code number, backward

Second case differs from first one only by step direction. (Scheme 3) Please note, that cycle starts with highest step #8. So first step in cycle does not use code at start# 5.

Relationship between parameters, start # + count < codes, reverse = on

bits	4																		
codes	16																		
count	8																		
start #	5																		
reverse	on																		
code #	-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
code	-	0	1	3	2	6	7	5	4	12	13	15	14	10	11	9	8		
dec value	(0)	0	1	3	2	6	7	5	4	12	13	15	14	10	11	9	8		
step #	0					1	2	3	4	5	6	7	8						
current code #	7																		
current dec value	5																		
current step	3																		
cycle start at step #	8																		

Scheme 3



### 3.3 Start# + Count > Code number, forward

Now highest code index is already reached at step #5. Instead of code index #1 following step #6 continues with code #9.

Relationship between parameters, start # + count > codes, reverse = off

bits	4																		
codes	16																		
count	8																		
start #	5																		
reverse	off																		
code #	-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
code	-	0	1	3	2	6	7	5	4	12	13	15	5	10	11	9	8		
dec value	(0)	0	1	3	2	6	7	5	4	12	13	15	5	10	11	9	8		
step #	0									6	7	8	1	2	3	4	5		
current code #	14																		
current dec value	11																		
current step	3																		
cycle start at step #	1																		

Scheme 4

### 3.4 Start# + Count > Code number, backward

As in 3.2 described, cycle starts with highest step #. In this case it is with code #11, just before start# 12.

Relationship between parameters, start # + count > codes, reverse = on

bits	4																		
codes	16																		
count	8																		
start #	5																		
reverse	on																		
code #	-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
code	-	0	1	3	2	6	7	5	4	12	13	15	5	10	11	9	8		
dec value	(0)	0	1	3	2	6	7	5	4	12	13	15	5	10	11	9	8		
step #	0									6	7	8	1	2	3	4	5		
current code #	14																		
current dec value	11																		
current step	3																		
cycle start at step #	8																		

Scheme 5

## 4 Controls and connectors

### 4.1 Code generator



In *bits* mode a list will be generated with all Gray Codes that are possible with that number of bits. With small '+' and '-' buttons you can chose 2 to 8 bits.



It's rather hard to create balanced Gray Codes. That's why a fixed list of 256 codes is stored in the module. Number of bits and codes cannot be changed.



In *codes* mode user can define, how many codes are to be generated. Because of Gray Code principle the number must be even. Values from 4 to 256 are valid. Number of bits results automatically from codes number and is not adjustable.

### 4.2 Code index range



With *start knob* one can chose the code index, the sequencer should begin with (in forward direction). Every value from one to total number of generated codes is valid.

*Count knob* lets you set total number of steps for one cycle. Every value from one to total number of generated codes is valid.



There is a second method to set start index. When a cable gets connected to *start input jack*, digital display moves and frees place for two other knobs.

- *Gain knob* (yellow) multiplies input voltage by -5.0 to +5.0. (default: 0)

- *Offset knob* (orange) adds an offset of -5 to +5 volt.

Resulting value will be limited to match the 1 to (codes) interval.



Same is for count. When a cable gets connected to *count input jack*, digital display moves and frees place for two other knobs.

- *Gain knob* (yellow) multiplies input voltage by -5.0 to +5.0. (default: 0)

- *Offset knob* (orange) adds an offset of -5 to +5 volt.

Resulting value will be limited to match the 1 to (codes) interval.

### 4.3 Step



Sequencer goes to next step, when it gets triggered. That can be done by pushing the step button or by a CV trigger. Duration of CV trigger pulse doesn't matter. As trigger a rising voltage is seen, that exceeds +2.5 volt.

LED gets lit for a short time to signalize a received trigger pulse.

### 4.4 Restart control



By pushing the red marked button, sequencer gets prepared for operation and step counter gets unlocked. With slider switch on "0" position, step counter goes to virtual step #0 and will wait for first step trigger.

When (count) steps are done, internal step counter gets locked.



Instead of getting locked, sequencer will automatically do a restart at next step trigger, when (red marked) *auto restart button* is toggled ON.

**Please note: Activation of auto restart does not unlock a locked step counter.**



If slider switch is on 0 right side, an external restart trigger (manually or by CV) sets step counter to step #1 (forward) or to step (count)# (reverse).

## 4.5 Reverse control



Reverse label field gets orange as long as current step direction is backward (reverse).

Step direction can be reversed, when orange push button is toggled on.

Direction can also statically reversed by an input voltage higher than +2.5 volt. Red LED will be lit as long as this voltage is high enough.

Until now *cycles slider switch* was not mentioned yet. We can imagine that it was on position OFF and didn't matter.

Now we shift it to "3" position. This will activate a cycle counter. Grey area below slider marks count of cycles that will be done in one direction before direction will be reversed automatically.

Orange circle marks actual cycle. The picture shows third of three cycles.

At first step of next cycle direction changes to reverse and cycle counter gets reset to 1.

Please note that small number at slide switch on the top changed from 1 to 4. That is because step count was set to 4 and so step #4 is first step of actual cycle.

## 4.6 New cycle output



*New cycle output* provides a signal when a step cycle begins. With slide switch in *gate* position output voltage is +5 volt as long as sequencer is in step #1 (forward) or (count)# (reverse). LED is on at same time.

When switch is in *trig* position, output sends only a short trigger pulse.

A right mouse click on trig label opens a menu. Now click with left mouse button on one of the menu lines to select a pulse duration.

**Please Note:** This selection is also valid for binary code outputs. (see later)

## 4.7 Analog outputs



Here are the analog outputs. Output voltage is normalized to each specific quantity.

Example for code count = 5:

step 0	0 volt
step 1	1 volt
:	:
step 5	5 volt

Example for start code# = 7:

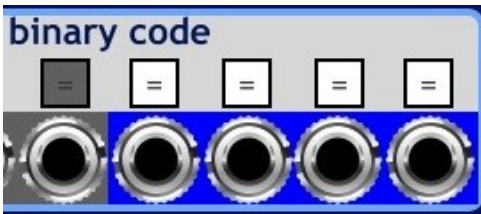
code #7	0 volt
code #8	1 volt
:	:
code #11	5 volt

- Yellow *gain knobs* can be adjusted from 0 to 10. Default is 5.

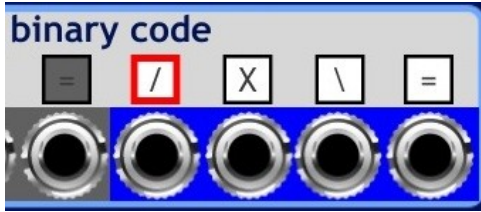
- Orange *offset knobs* have a range from -5 to +5. Default is 0.

**While sequencer is on step #0, all analog outputs are set to 0, independent of knob settings.**

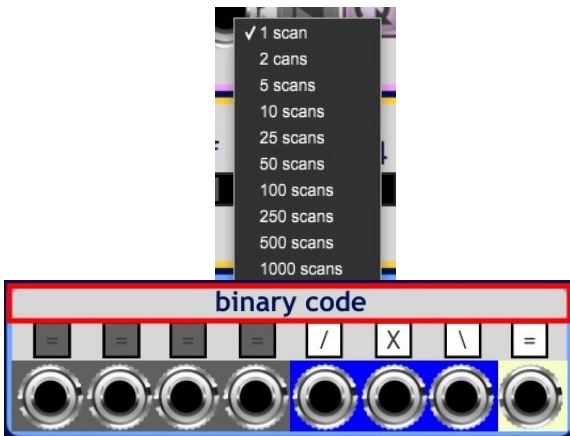
## 4.8 Binary code outputs



On the bottom of module front there are eight digital outputs, one for each bit of a code. On the most right side is lowest bit 0, on most left side highest bit 7.



Small label above each output show it's actual mode. When mouse cursor is on a label, a red border appears. That signalizes that you can change output mode with a left click on it. (see 2.7.2)



When mouse cursor in on grey area above the small labels, a red rectangle appears. With a right click on it you will get a menu for trigger pulse duration. You can select a line with a left click on it.

Please Note: This selection is also valid for new cycle output. (see above)

# 5 Oscillograms

In this chapter some oscillograms should illustrate how Grain Steps works.

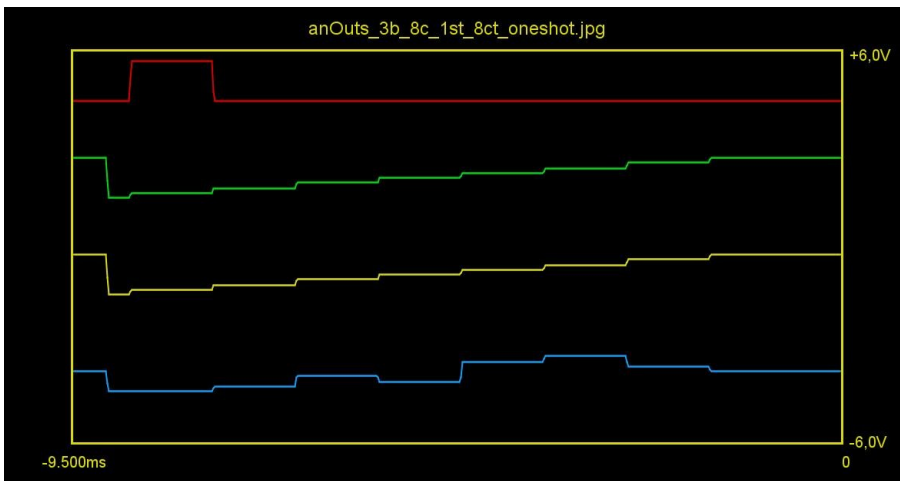
Common setup:

bits	3
codes	8
external restart to	0

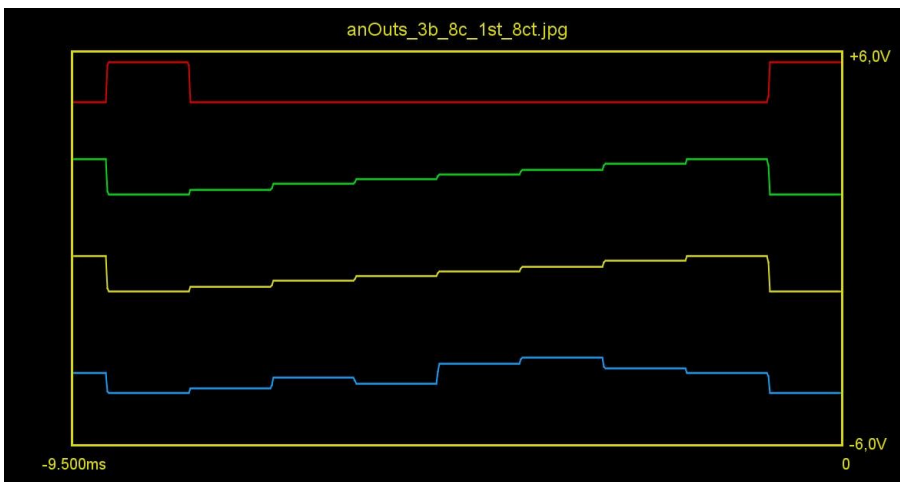
## 5.1 Analog outputs

Following oscillograms show signals (from top to bottom):

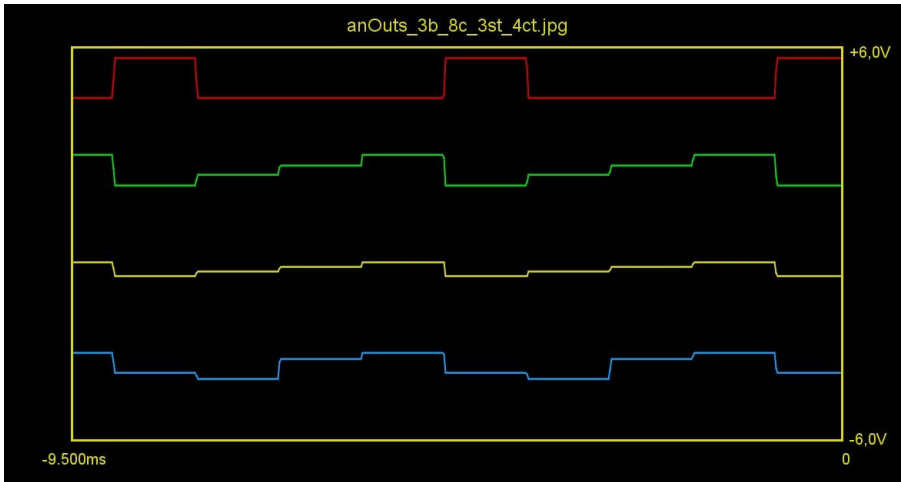
new cycle output	red
step # output	green
code # output	red
decimal value	blue



start#	1
count	8
auto restart	off

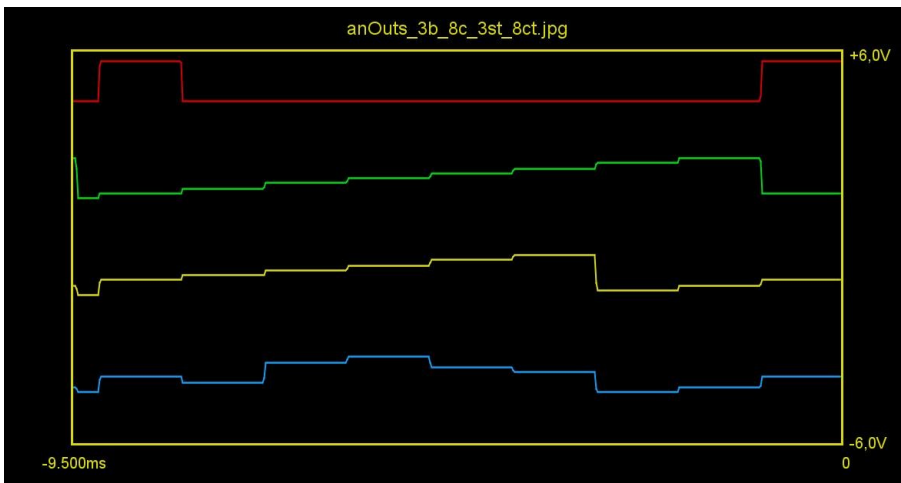


start#	1
count	8
auto restart	on



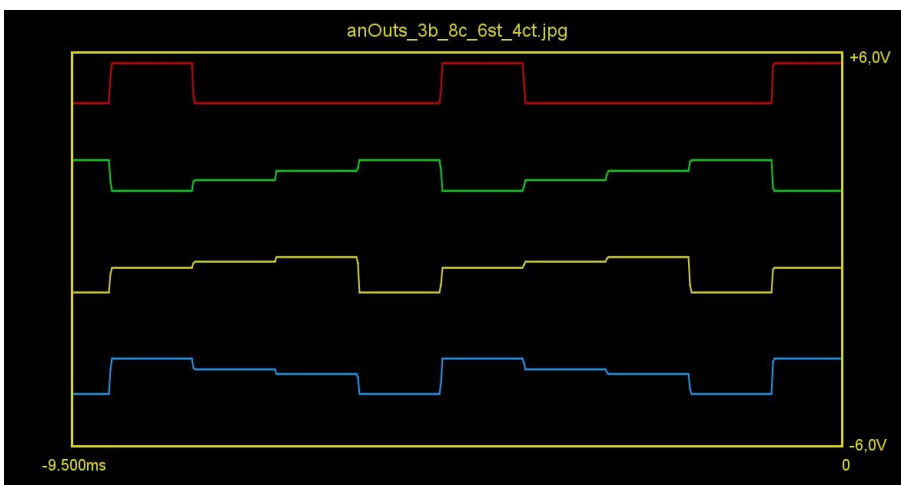
**start#**        3  
**count**        4  
**auto restart** on

**code series:**  
 3 4 5 6/3 4 5 6/ ...



**start#**        3  
**count**        8  
**auto restart** on

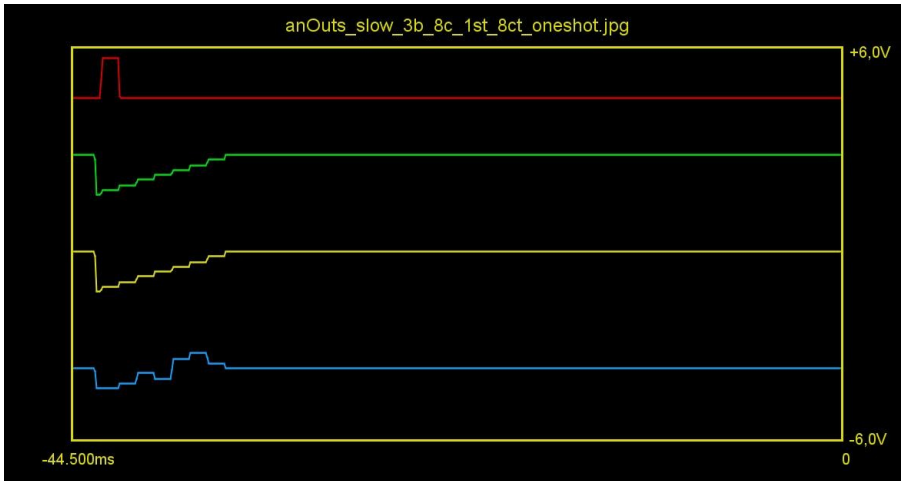
**code series:**  
 3 4 5 6 7 8 1 2/3 4 ...



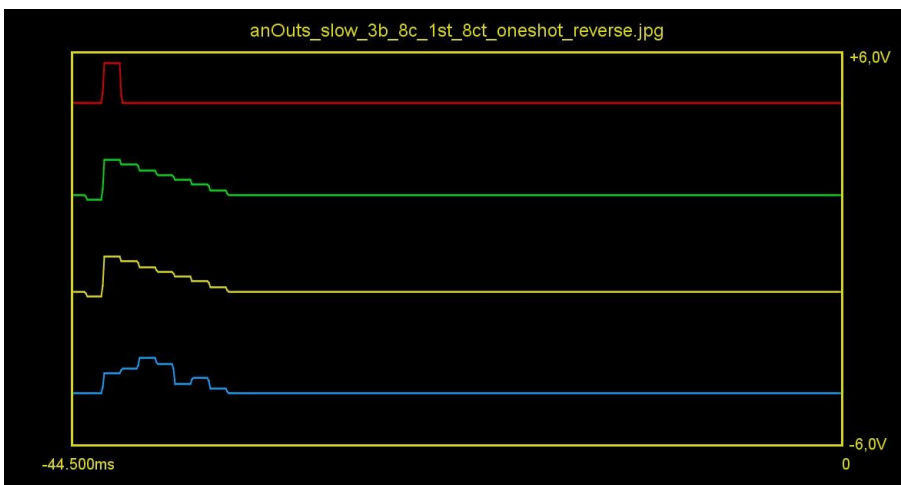
**start#**        6  
**count**        4  
**auto restart** on

**code series:**  
 6 7 8 1/6 7 8 1/...

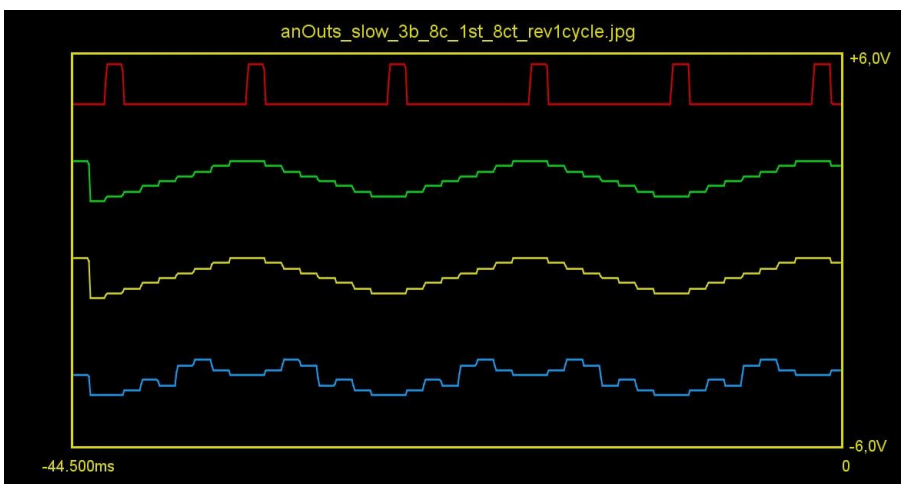




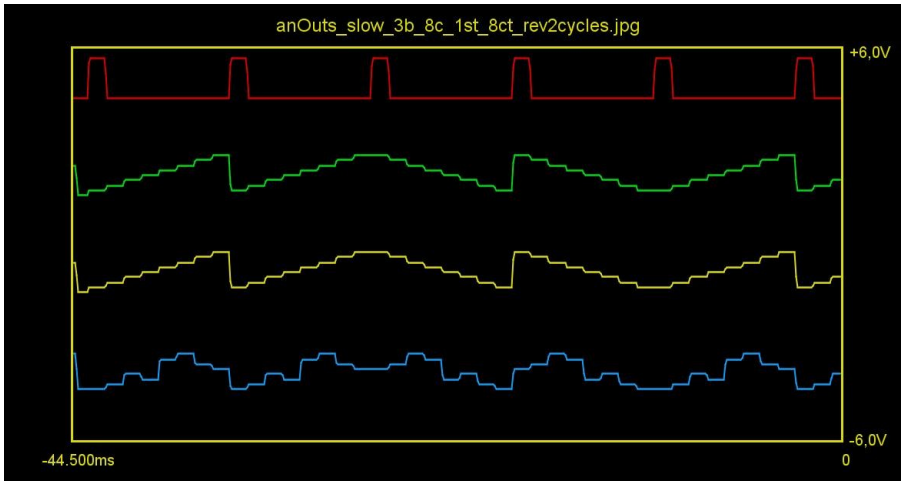
**start#**        **1**  
**count**        **8**  
**auto restart**   **off**  
  
**auto reverse**  
**cycles**        **off**



**start#**        **1**  
**count**        **8**  
**auto restart**   **off**  
**reverse**        **on**  
  
**auto reverse**  
**cycles**        **off**

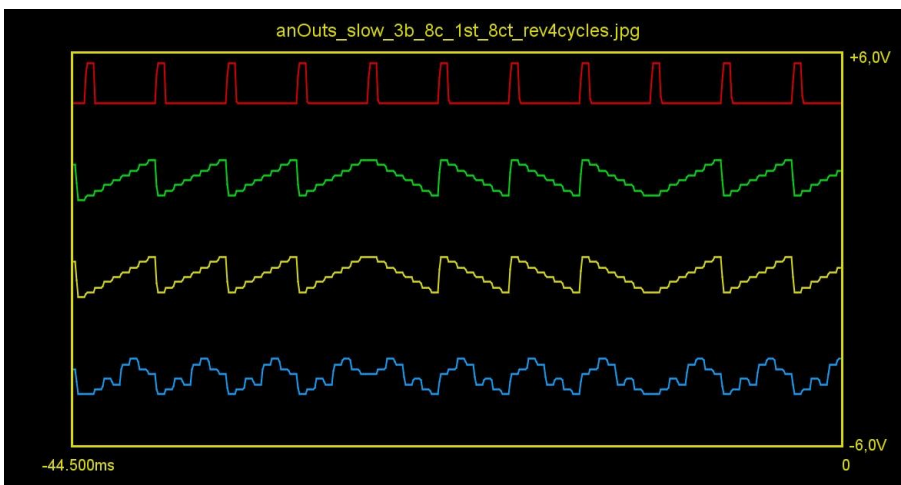


**start#**        **1**  
**count**        **8**  
**auto restart**   **on**  
  
**auto reverse**  
**cycles**        **1**



**start#**        **1**  
**count**         **8**  
**auto restart**   **on**

**auto reverse**  
**cycles**        **2**



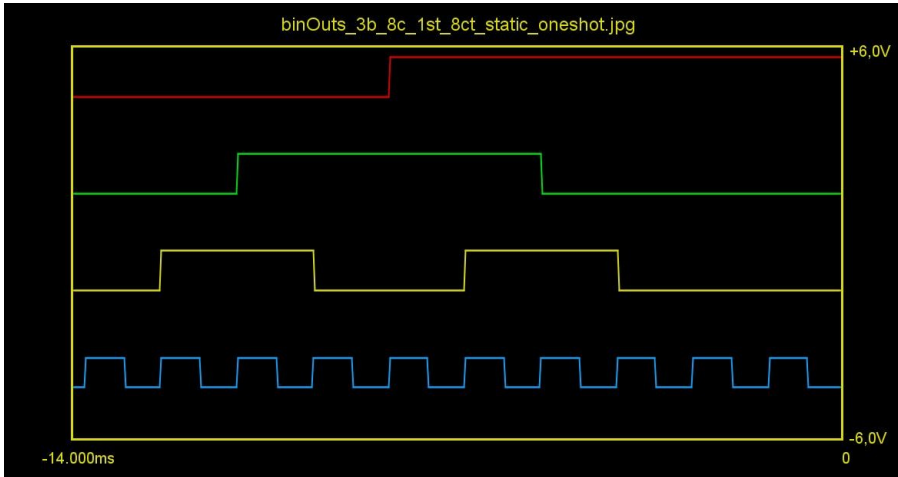
**start#**        **1**  
**count**         **8**  
**auto restart**   **on**

**auto reverse**  
**cycles**        **4**

## 5.2 Binary outputs

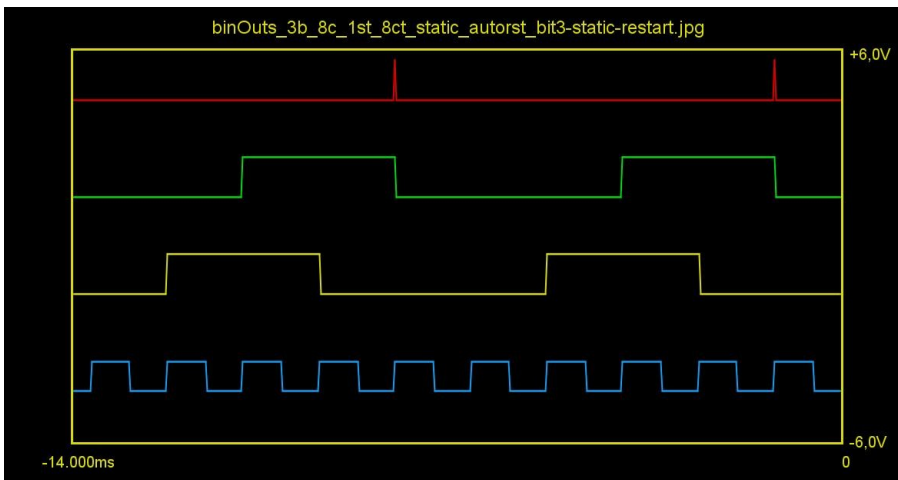
Following oscillograms show signals (from top to bottom):

bit 3 output	red
bit 2 output	green
bit 1 output	red
step input	blue



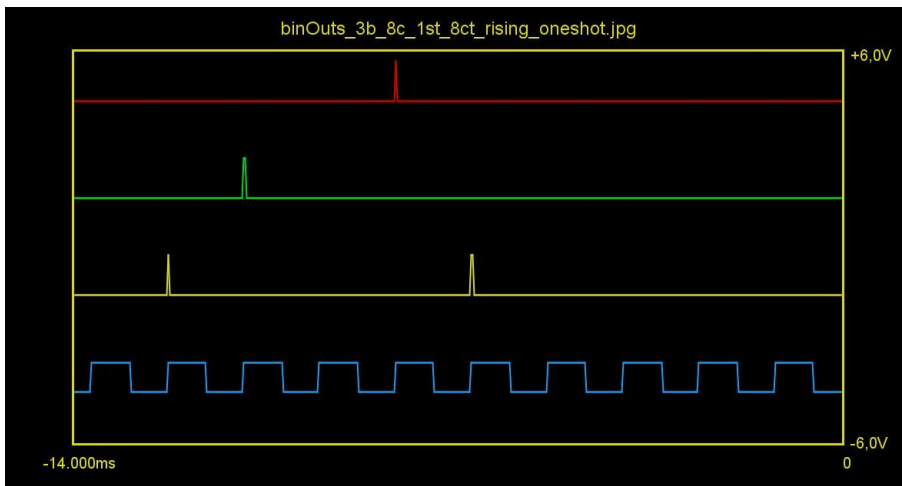
start#	1
count	8
auto restart	off

output modes	
bit 3	=
bit 2	=
bit 1	=



start#	1
count	8
auto restart	on

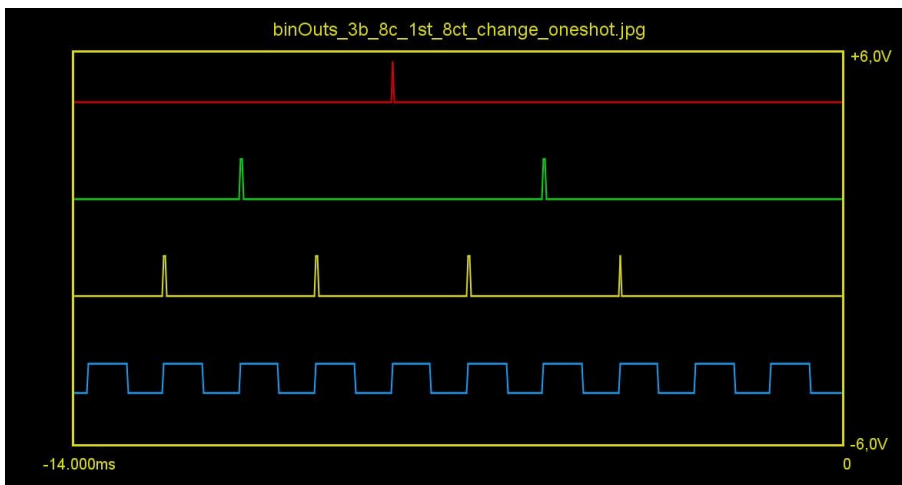
output modes	
bit 3	/
bit 2	=
bit 1	=



**start#**        **1**  
**count**         **8**  
**autorestart**   **off**

**output modes**

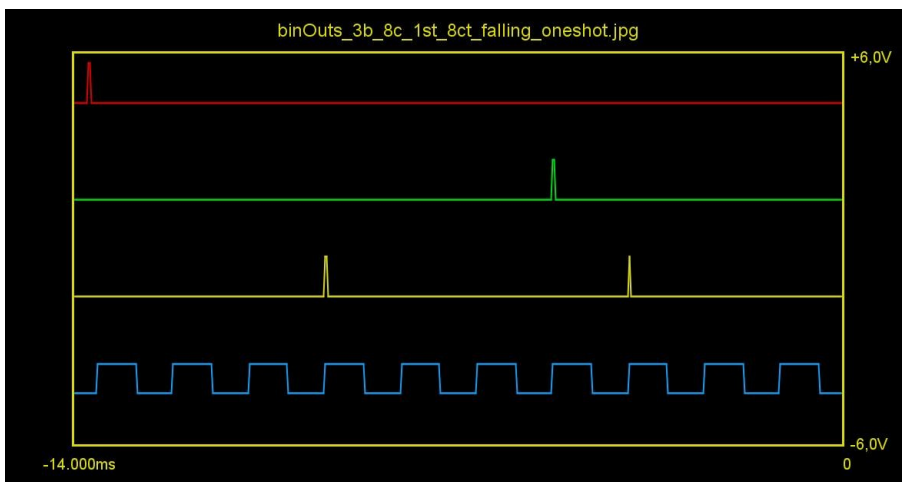
**bit 3**         /  
**bit 2**         /  
**bit 1**         /



**start#**        **1**  
**count**         **8**  
**autorestart**   **off**

**output modes**

**bit 3**         X  
**bit 2**         X  
**bit 1**         X



**start#**        **1**  
**count**         **8**  
**autorestart**   **off**

**output modes**

**bit 3**         \  
**bit 2**         \  
**bit 1**         \

## 6 Links

You can find example presets and other PDF manuals at:

<https://p-moon-modules.de>