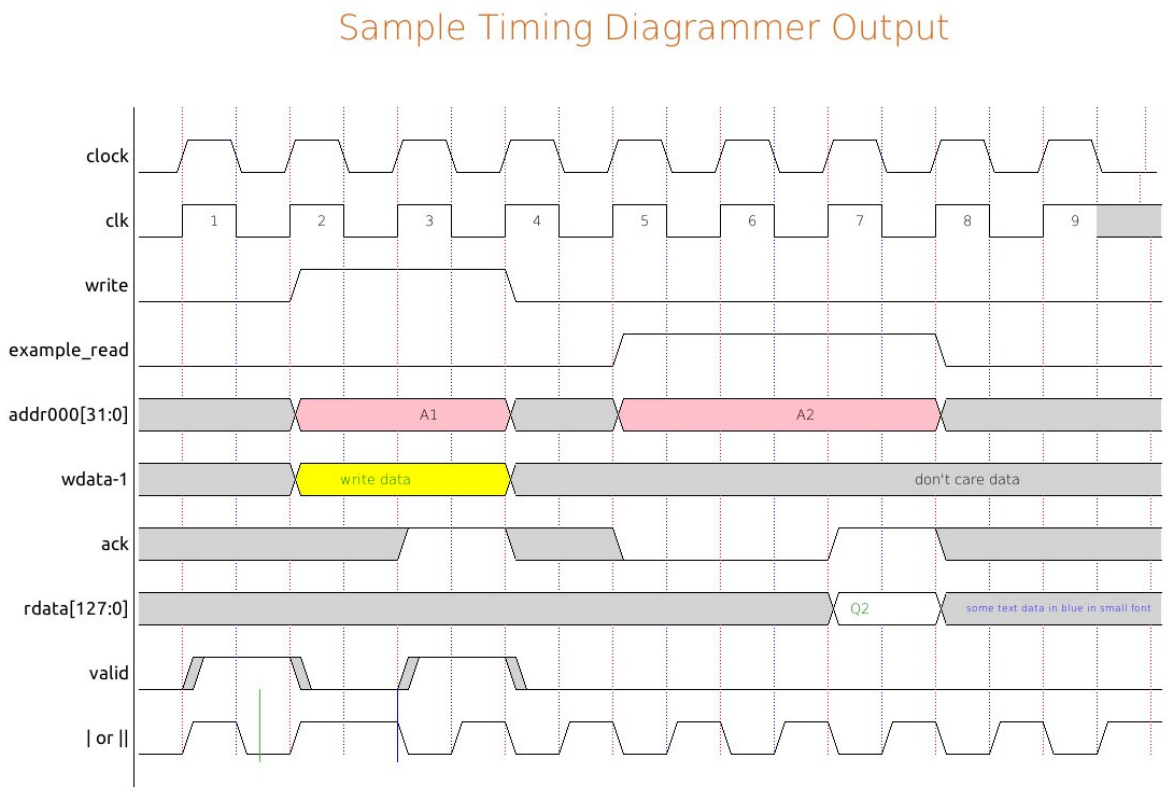
Timing Diagrammer

Timing Diagrammer is based on the Waves Timing Diagram Editor and similar in concept. The commands are not backwards compatible with Waves version 1. The source code has been completely re-written in Python-3 using Qt5 toolkit.

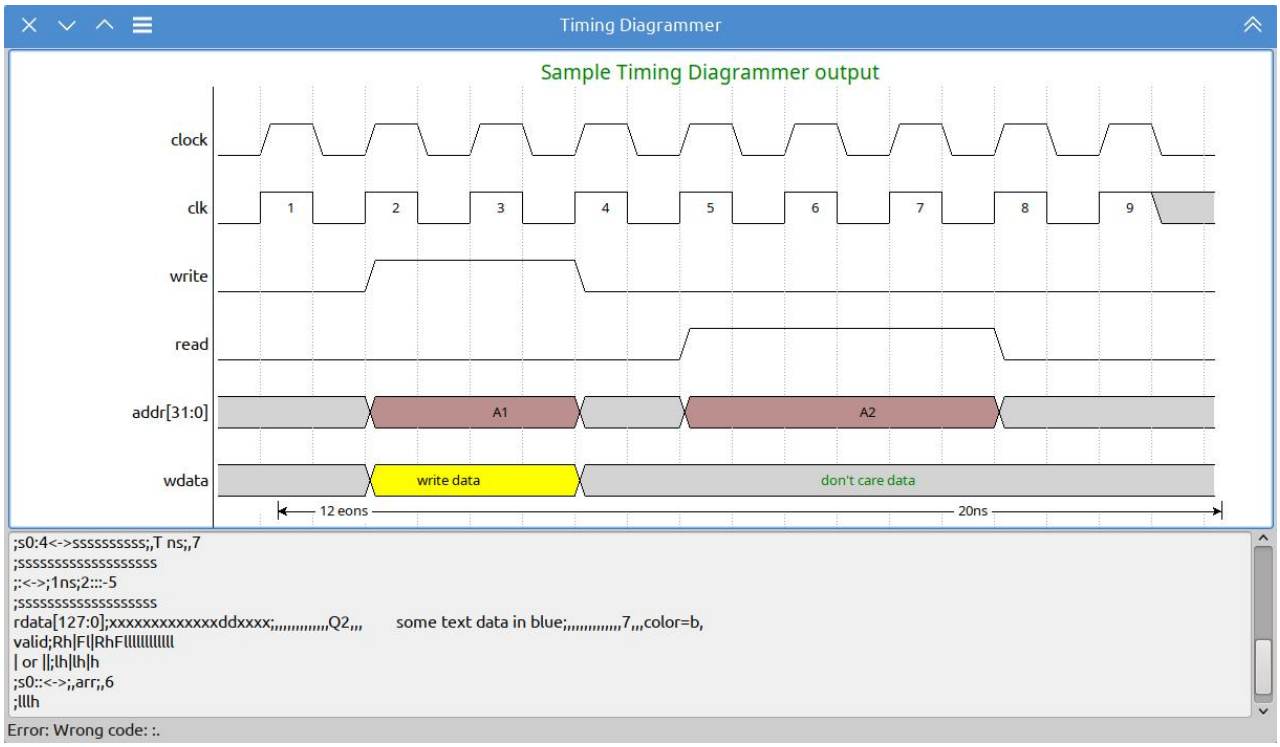
Like Waves, *Timing Diagrammer* retains its inspiration from Wavedrom, the JSON-based timing diagrammer.

The output from Timing Diagrammer is significantly superior to that from Waves, see below. The command set of *Timing Diagrammer* has been enhanced.



1. Timing Diagrammer Window

Timing Diagrammer has a unified waveform (canvas) window where the waveform is displayed and an editor window where commands are typed in.



2. Timing Diagrammer Canvas

The *Timing Diagrammer* canvas, like *Waves*, is divided into equal-sized vertical grid that can be thought of as a *half* clock period. Each signal will be described by a new line of command in the source entry window. Each waveform command adds a piece of the waveform for one grid, *i.e.*, a half-clock period, except the clock (periodic pulse) commands, *p* and *P*.

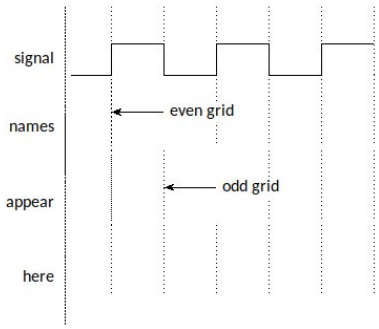


Figure 1: *Timing Diagrammer* Grid

A *Timing Diagrammer* command consists of four sections, each separated by a colon. This is shown in Figure 2.

signal name;*waveform command*;*wave annotations*[*annotation parameters*],
wave annotations[*annotation parameters*],…

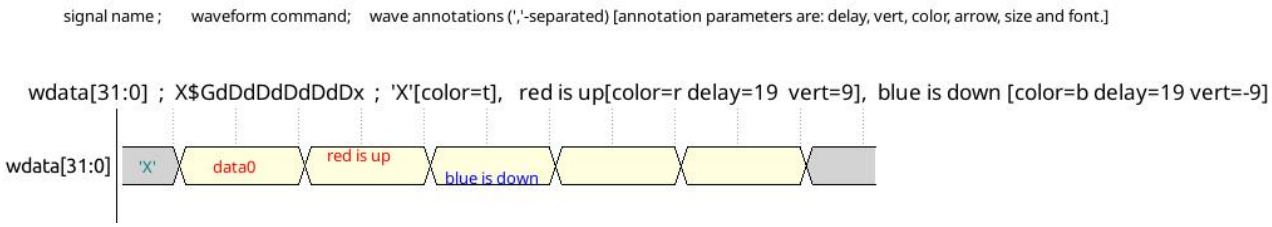


Figure 2: Command Format

3. Timing Diagrammer Signal Names

Signal names are the first part of the semi-colon-separated command string. The signal name can be of any length and is right justified. All characters including comma 'brackets, '[' and ']', hash/pound '#' and semicolon ';' are allowed. Signal names are optional. The first version of the program *Waves* did not allow colon in the signal names.

4. *Waveform Commands*

Waveform commands form the first part of the command string. The *Timing Diagrammer* commands are shown in Figure 3.

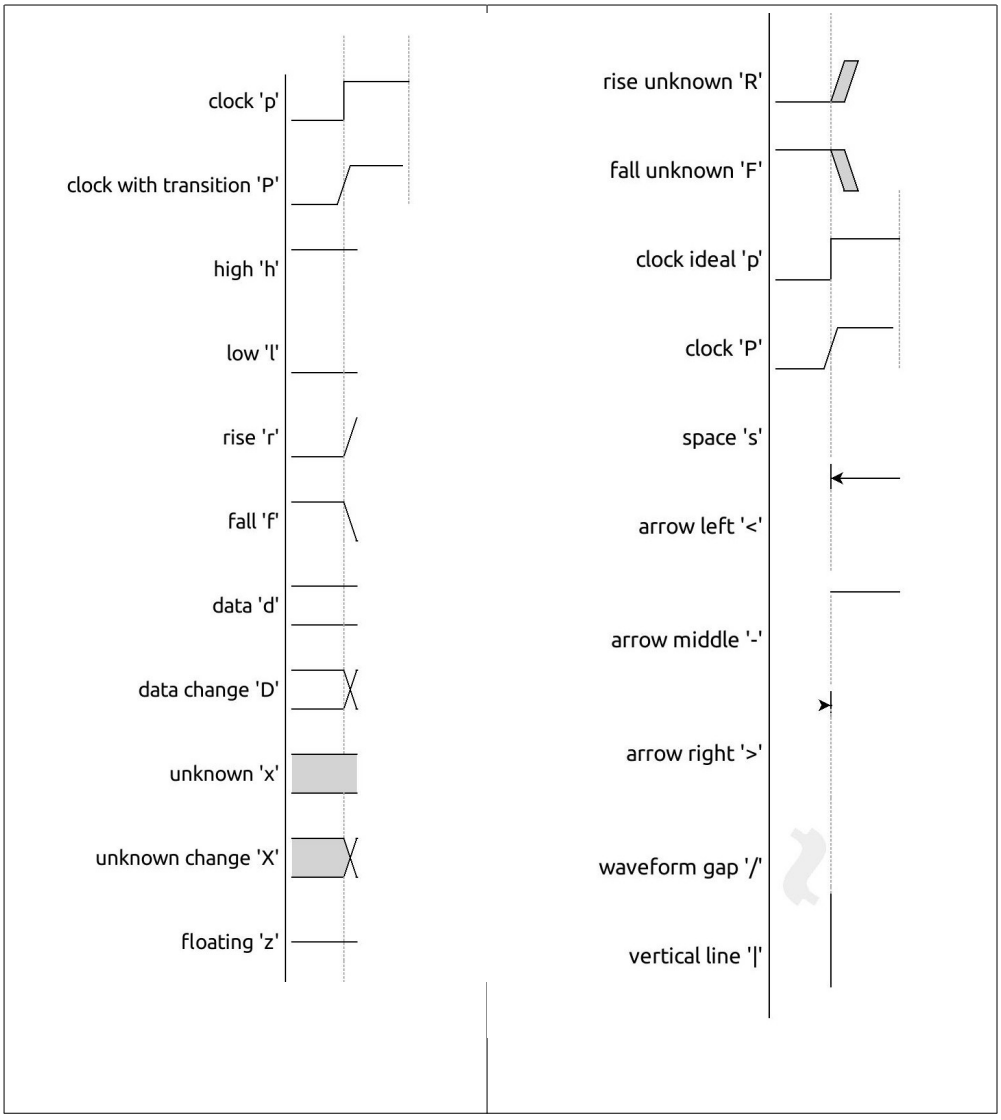


Figure 3: Timing Diagrammer Commands

5. *Waveform Metacommands*

Timing Diagrammer metacommands can appear in the waveform command and change the way shown in Table 1:

#	This indicates a comment till the end of the line.
# !	Waveform directives (global – executed before all waveform commands, or inline – executed in sequence of commands)
;	Command separator.
,	Grid separator for annotation and annotation parameters – indicates the grid where the annotation starts and which the parameter refers to.
\	When occurring as '\#', '\;' or '\,', will insert that literal character in the annotation string or signal name.
$	Color specifier. The next character is the color code. See the next table

Table 1: *Timing Diagrammer* Metacommands

5.1. *Waveform Colors*

Waveform colors codes are shown below. The color names and colors are found in the this W3 page.

'a'	'antiquewhite'	'D'	'deepskyblue'	'O'	'orchid'	'w'	'white'
'A'	'aquamarine'	'g'	'mediumaquamarine'	'p'	'pink'	'W'	'wheat'
'b'	'light blue'	'G'	'springgreen'	'P'	'peachpuff'	'x'	'black'
'B'	'bisque'	'l'	'lavender'	'r'	'rosybrown'	'y'	'light yellow'
'c'	'lightcyan'	'm'	'mistyrose'	's'	'snow'	'Y'	'yellow'
'C'	'cornflowerblue'	'k'	'khaki'	't'	'thistle'	'z'	'plum'
'd'	'skyblue'	'o'	'orange'	'v'	'violet'	'Z'	'cornsilk'

6. Waveform Annotations

6.1. Data Annotation

The data annotation section starts after third semicolon in the command. Each comma-separated string is associated with a grid. The annotation parameters are separated by space or spaces. The parameters are :

delay=value width=value color=value font=value size=value center=value vert=value

Parameter	Description
delay	Moves the text to the right by *value* (1/9th part of a half-clock period)
width	The maximum width of the text is *value* pixels, after which the text wraps (left-aligned)
color	Text color. See text color code table below.
font	Font face name.
size	Font size.
center	'Y' or 'y' indicates text is center, else it will be aligned to the left.
vert	Vertical displacement of the text in pixel (TODO)

Recognized color keyword names. The color names and colors are found in the this W3 page.

'a'	'aqua'	'g'	'green'	'O'	'olive'	'V'	'fuchsia'
'A'	'aquamarine'	'G'	'goldenrod'	'p'	'deeppink'	'w'	'darkslategrey'
'b'	'blue'	'i'	'indigo'	'P'	'purple'	'W'	'saddlebrown'
'B'	'brown'	'l'	'limegreen'	'r'	'red'	'x'	'black'
'c'	'chocolate'	'm'	'maroon'	'R'	'firebrick'	'y'	'mediumvioletred'

'C'	'crimson'	'M'	'magenta'	's'	'slateblue'	'Y'	'gold'
'd'	'darkblue'	'k'	'khaki'	't'	'teal'	'z'	'navy'
'D'	'darkred'	'o'	'orangered'	'v'	'darkviolet'	'Z'	'darkorchid'

7. Advanced Waveform Commands

- Time arrows
- Flexible arrows
- Flexible text

7.1. Time Arrows

Waves supports arrows on the time axis as shown in Figure 4 that can be optionally annotated (see section on Data Annotation). The position of the annotation is from the first gridline on the left. Please note that since the annotation is associated with the gridline where the left arrowhead starts, it must be after the second comma.

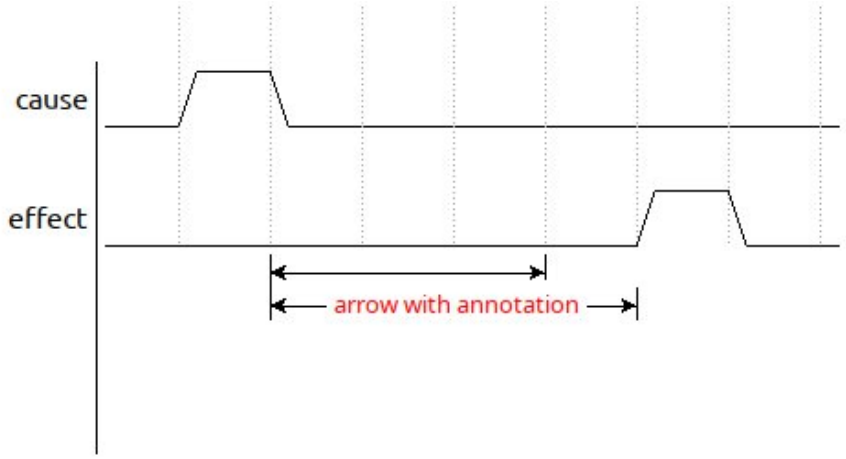


Figure 4:Waves Time Arrows

Code for waveform above

```
cause;rfllllll
effect;llllrfl
;ss<-->
;ss<--->;,,arrow with
annotation;,,delay=13 color=r
```

The time arrow commands are shown in Figure 5.

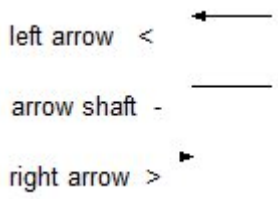
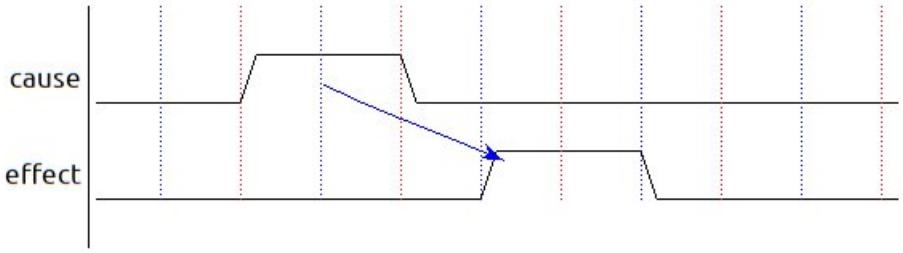


Figure 5: Time Arrow Commands

7.2. *Direction ('Cause-Effect') Arrows*

A cause-effect arrow starts with a '+' (before or after the '**;**') followed by the canvas (x, y) coordinates of the start point and then the end point.



Code for waveform above

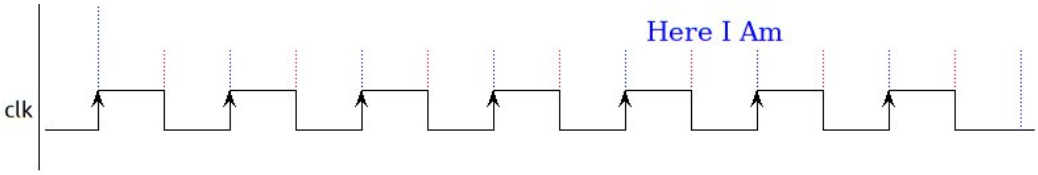
```
#!color dirarr b
cause;llhhllllll
effect;lllllhhlll
;+247 739 359 786
```

The coordinates can be automatically placed on the editor line:

- On a new line in the editor, add a '+' as the first character
- Press the Control key and double click on the start point of the arrow on the canvas.
  - A blue circle marks the point (tail of the arrow)
  - The coordinates are added in the editor
- Press Control and double click on the end point (arrow head). A straight arrow is now drawn from the start to the end point.
- If a curved arrow is required, press Control and double click on the point towards which the arrow should be curved.

7.3. *Place-anywhere Annotation*

A place-anywhere annotation is specified by a '^' character followed by the canvas (x, y) coordinates at which the text is placed.



Code for waveform above

```
clk;cccccccl
;^572 555 Here I Am
```

The coordinate can be automatically placed.

- On a new line in the editor, add a '^' as the first character
- Press the Control key and double click on the start point of the arrow on the canvas.
  - A blue circle marks the point (start position of the text annotation)
  - The coordinates are added in the editor
- Type the required annotation after the coordinates in the editor window.

8. Legal