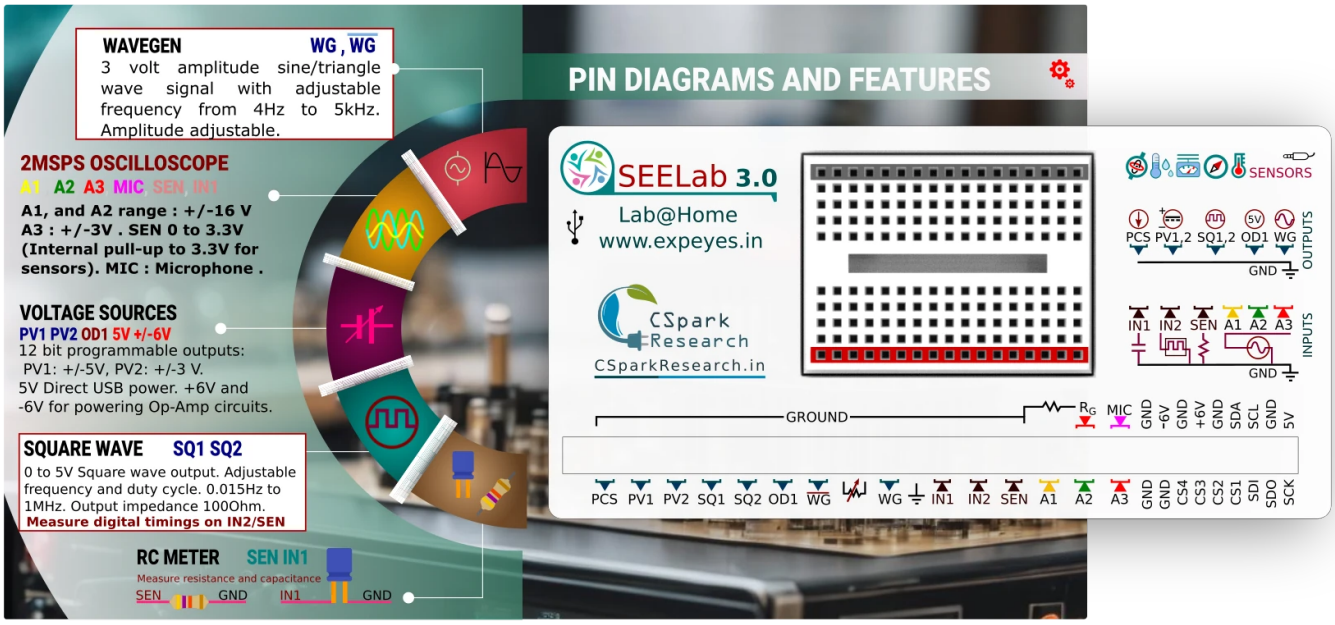
Waveform Generators



set_sine : Set Sine Wave Frequency for WG, WGbar

set frequency of sine wave on WG. Restores waveform type to sine if some other shape had been previously set. WGbar output will also output a sine wave which is 180 degrees out of phase with WG at all times.

parameter	description
frequency	4 to 5000 . Freq in Hz

invokes `p.set_wave(freq, 'sine')` under the hood.

```
p.set_wave(100) # 100 Hz sine wave on WG
```

---

set_wave : Set Frequency and type of WG waveform out

set frequency of wave on WG. Also sets waveform type to ‘sine’/’tria’.

parameter	description

parameter	description
frequency	4 to 5000 . Freq in Hz
type	‘sine’ or ‘tria’

```
p.set_wave(freq, 'sine')
```

set_sine_amp : Set Sine Wave Amplitude

Set the amplitude of the waveform output on WG

parameter	description
value	2 1x amplitude (3.3V)
	1 1V
	0 100mV

```
p.set_sine_amp(2) #3.3 V amplitude. +/-3.3V swing
```

load_equation : Load an arbitrary shape to WG using an equation

`p.load_equation(self, function, span=None, **kwargs)`

Load an arbitrary or preset waveform to the WG waveform generator output.

parameter	description
function	‘sine’ load sine wave
	‘tria’ triangle wave
	a python function of the form lambda x:expression(x)

```
p.load_equation('tria') # Changes waveform shape to triangle.
```

Use a python function: First two terms of the fourier expansion of a square wave

```
from matplotlib import pyplot as plt
import eyes17.eyes
p = eyes17.eyes.open()
# Connect WG to A1

def f1(x):    # First 2 terms of the fourier expansion of a square
wave.
        return sin(x) + sin(3*x)/3

p.load_table(f1,[0,2*np.pi]) #Evaluate from 0 to 2*pi
p.set_wave(400) # Set the frequency

#Measure the set waveform and study it.
x,y = p.capture1('A1', 500,10)
plt.plot(x,y)
plt.show()
```

---

set_sq1 : Set Square Wave Frequency for SQ1

`set_sqr1(self, freq, duty_cycle=50)`

set frequency of square wave on SQ1.

parameter	description
frequency	0.02 to 100000 . Freq in Hz
duty_cycle	0 to 100. default 50

Set a 1KHz square wave (0 to 5V) output on SQ1 with 10% duty cycle.

```
p.set_sq1(1000,10)
```

---

set_sq2 : Set Square Wave Frequency for SQ2

`set_sqr2(self, freq, duty_cycle=50)`

set frequency of square wave on SQ2.

> **Warning**
>
> This will disable the sine wave output on WG. invoking `set_sine` will restore the sine wave and disable this.

parameter	description
frequency	0.02 to 100000 . Freq in Hz
duty_cycle	0 to 100. default 50

```
p.set_sq2(1000)
```

Fourier Transformation demo

Connect WG to A1, and SQ1 to A2

```
import eyes17.eyes
p = eyes17.eyes.open()

from matplotlib import pyplot as plt
from eyes17 import eyemath17 as em

p.set_sine(1000)
p.set_sqr1(500)
t,v, tt,vv = p.capture2(5000, 20)   # captures A1 and A2

plt.xlabel('Freq')
plt.ylabel('Amplitude')
plt.xlim([0,10000])

#0.001 is to convert 20uS to mS units
xa,ya = em.fft(v,20*0.001)
plt.plot(xa,ya, linewidth = 2, color = 'blue')

xa,ya = em.fft(vv, 20*0.001)
plt.plot(xa, ya, linewidth = 2, color = 'red')

plt.show()
```