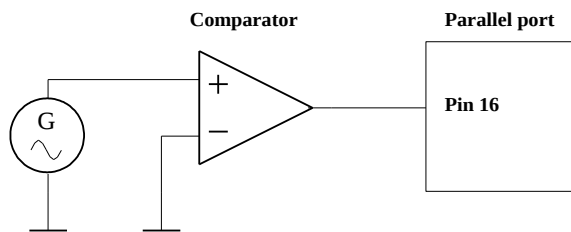


## LinuxCNC - HAL

### Generating a signal on a parallel port pin

Example: Pin 16 of parallel port 0 should deliver a 5kHz signal.

HAL block diagram:



The signal generator gives us an analog signal that is translated to a digital one (TRUE / FALSE) by using a comparator.

To interactively start HAL, type `halrun` in a terminal.

You get a prompt `halcmd`:

Load the components:

```
loadrt siggen
loadrt comp
loadrt constant
loadrt hal_parport cfg="0"
```

The constant is necessary for the second comparator input, it will be set to 0:

```
setp constant.0.value 0
```

(To set a duty cycle different from 50%, this constant could be changed)

The parallel port is configured as number 0, the first parallel port, normally with the address 0x378.

Define one or more threads. These define the available timing resolution. Often it is a good idea to define one thread for the fast and one for the slower operations:

```
loadrt threads name1=fast period1=50000 name2=slow period2=1000000
```

Here we have defined a fast thread updating its components every 50000ns = 50µs and a slow one with a timing of 1ms. (We need only the fast one)

The connections are done with the net command:

```
net s siggen.0.sine => comp.0.in1
net n constant.0.out => comp.0.in0
net a comp.0.out => parport.0.pin-16-out
```

The net names (s, n, a) are arbitrarily chosen.

For all components the thread for their updating must be defined with the `addf` function:

```
addf comp.0 fast
addf parport.0.write fast
addf siggen.0.update fast
```

Now the signal generator is ready to run. **It must be started with the HAL command `start`**

All components now run with their default values.

To set the frequency of the signal (or other parameters), use the `setp` command:

```
setp siggen.0.frequency 5000
sets f= 5kHz
```

### ***Debugging commands***

```
show comp    show components
show pin     show pins (virtual inputs and outputs of the components)
show param   show parameters like frequency, amplitude etc.
              ( R = readable, RW = readable and writable)
show funct   functions executed by the real time thread(s) like .update, .write ...
show threa d show threads, with indication on the ability for floating point operations
```

### ***Debugging instruments***

HAL meter: `loadusr halmeter`

HAL scope: `loadusr halscope`

HALscope must be linked to a thread! (first dialog)

If no trigger condition is set, the Trigger FORCE button forces a trigger, so the waveforms are displayed.

### ***Troubleshooting***

To unlink a false connection:

```
unlink <pin-name>
```

### ***Save a session***

```
save all saved.hal
```

### ***Exit HAL***

```
exit
```

### ***Restoring a session***

```
halrun -I -f saved.hal
```