

# **HAL for Dummies**

Centurion CNC Club

C.L. Churms

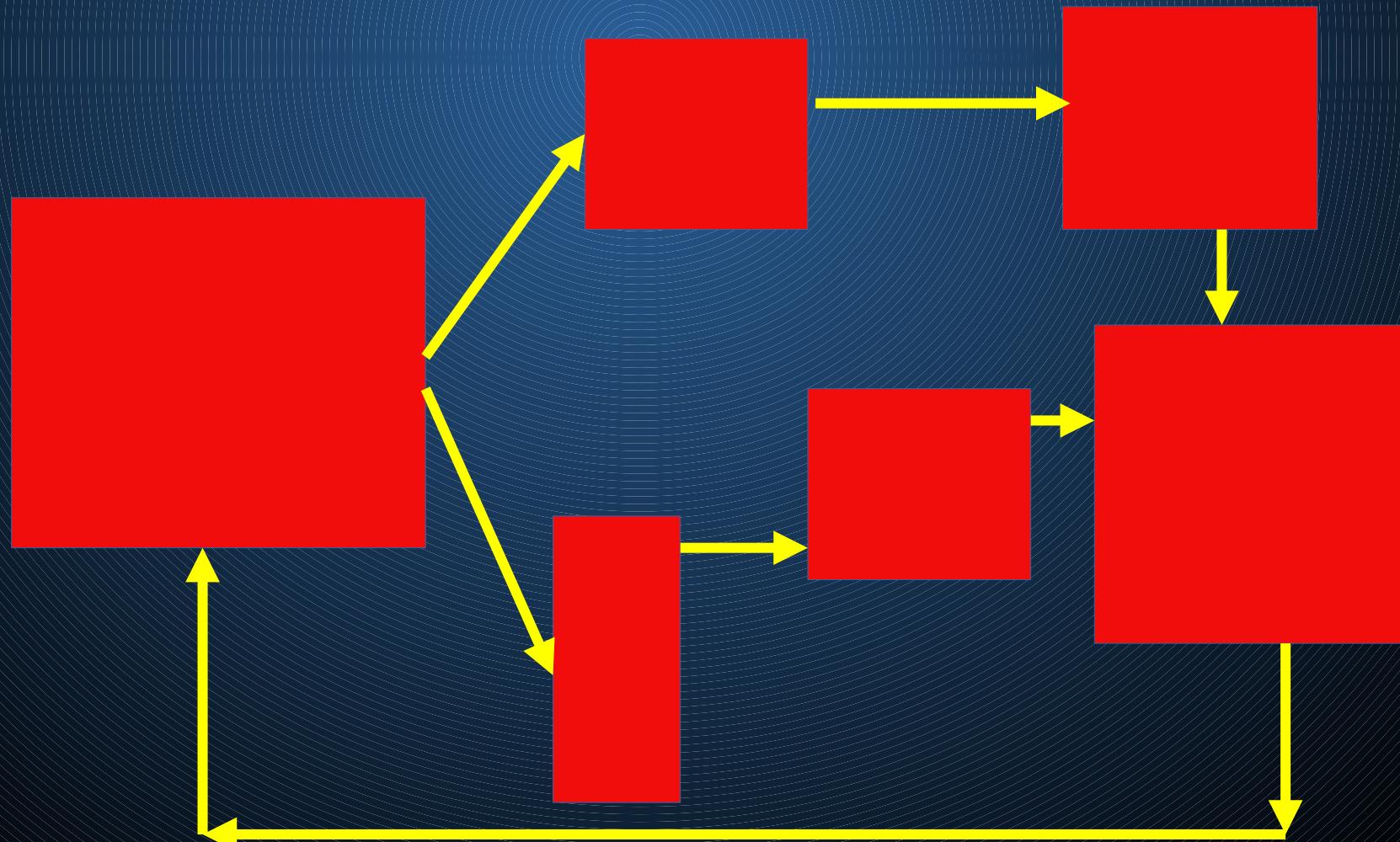
27/07/2021

# HAL is Relatively Simple (You must be kidding!)

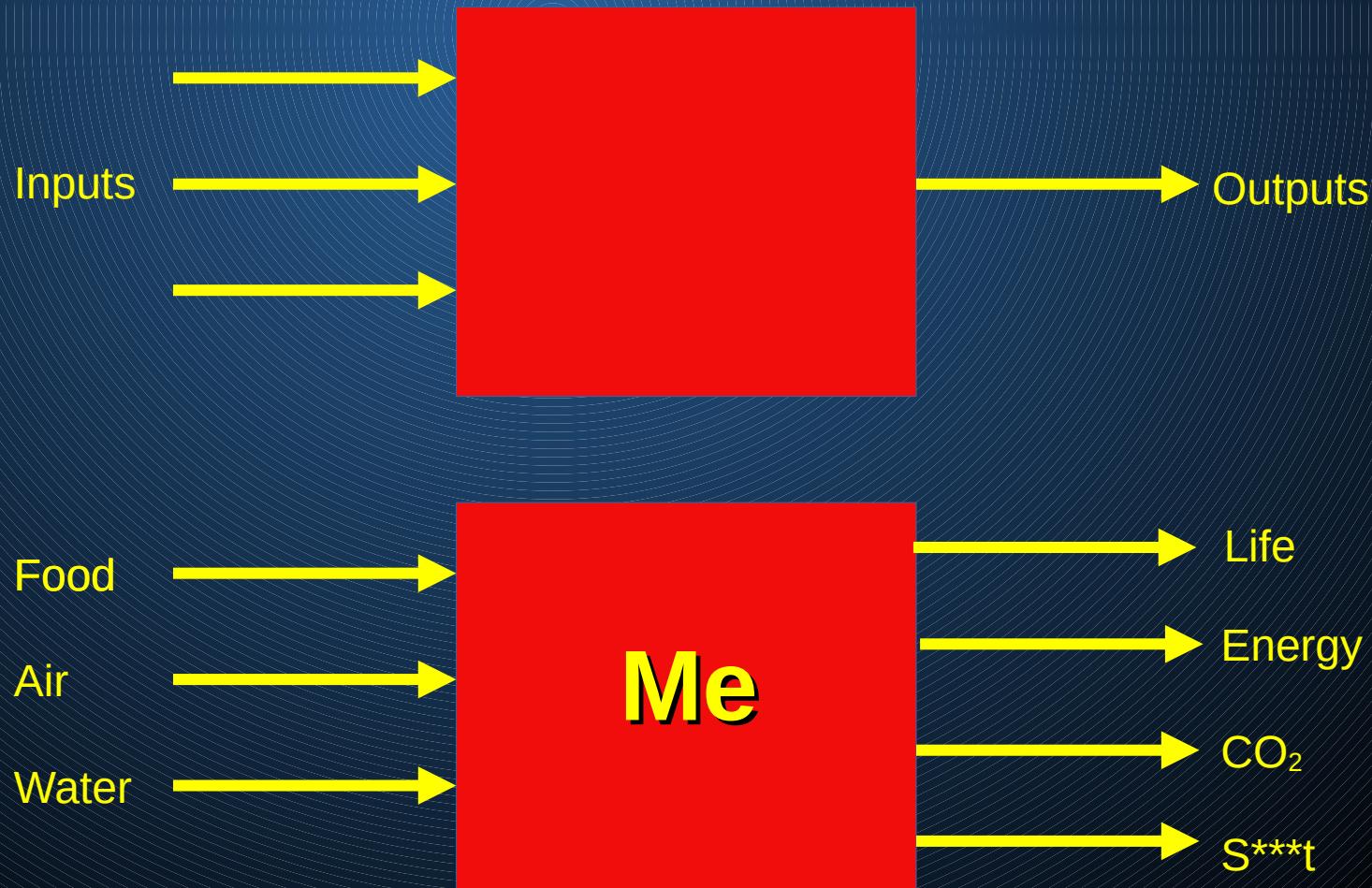
I believe the arrangement of the sections  
in the documentation  
makes it feel complex

I attempt to separate the basic operation of  
HAL from the more advanced (and confusing)  
sections of the documentation

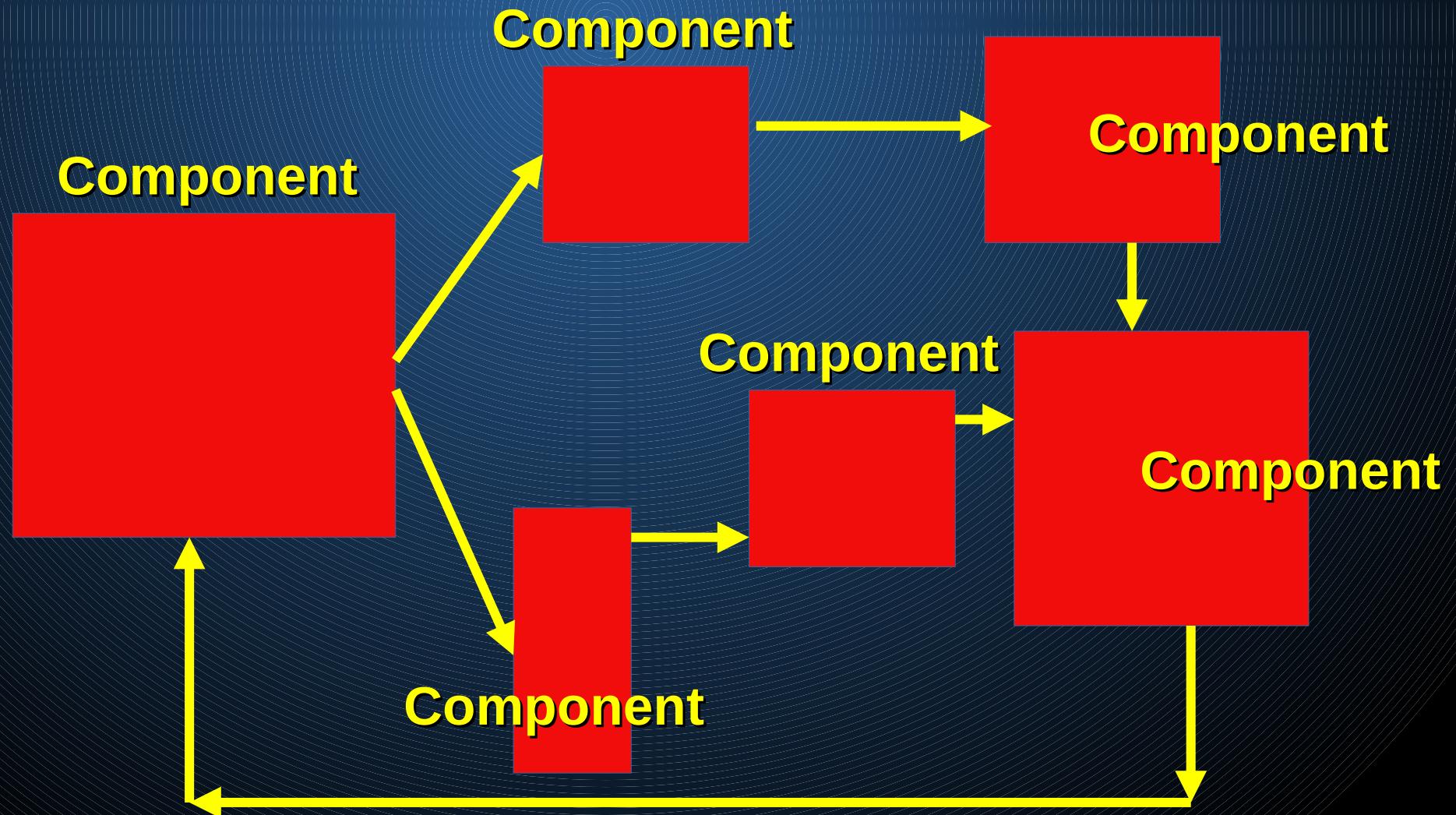
# Just joining of Black Boxes



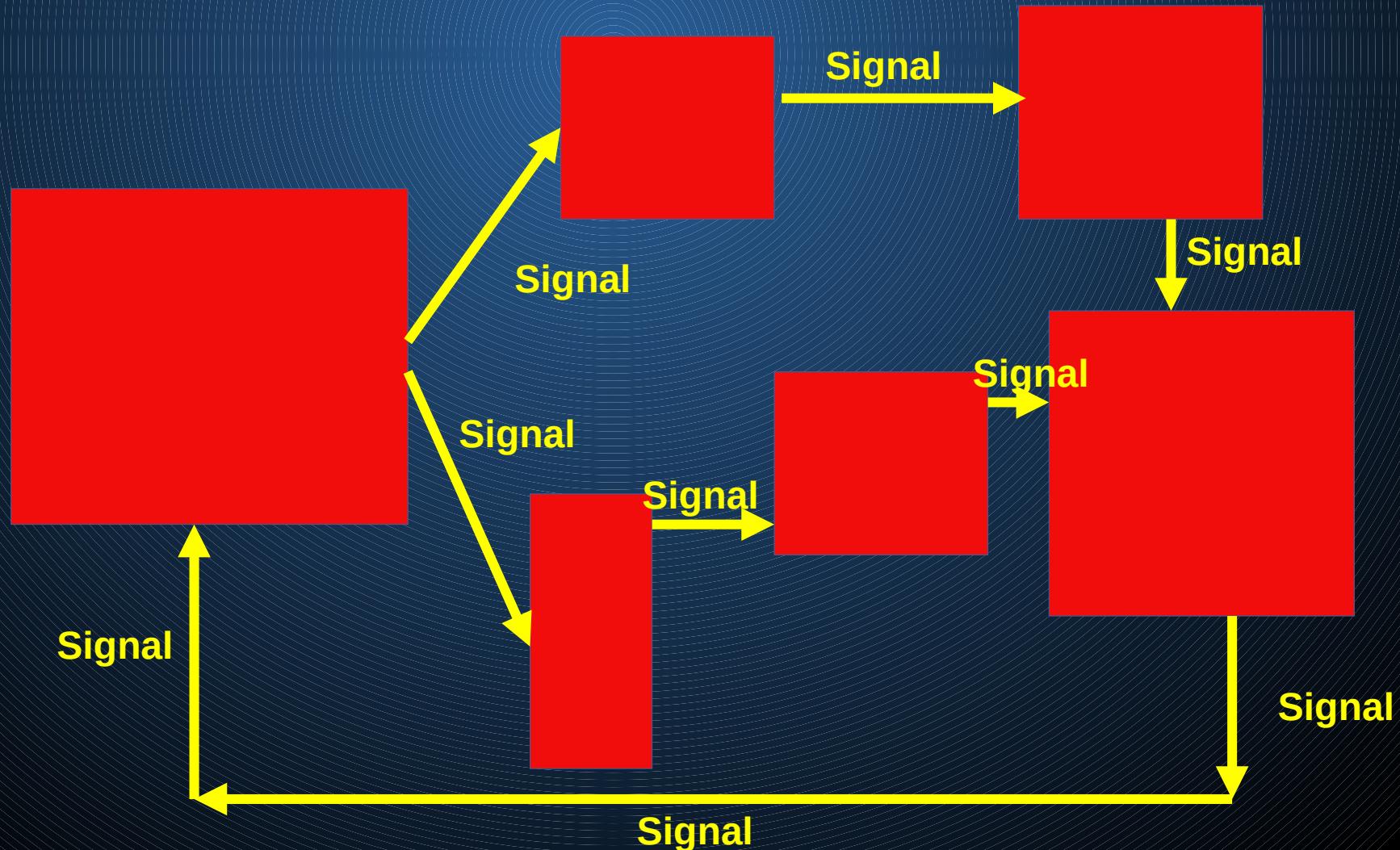
# What is a Black Box?



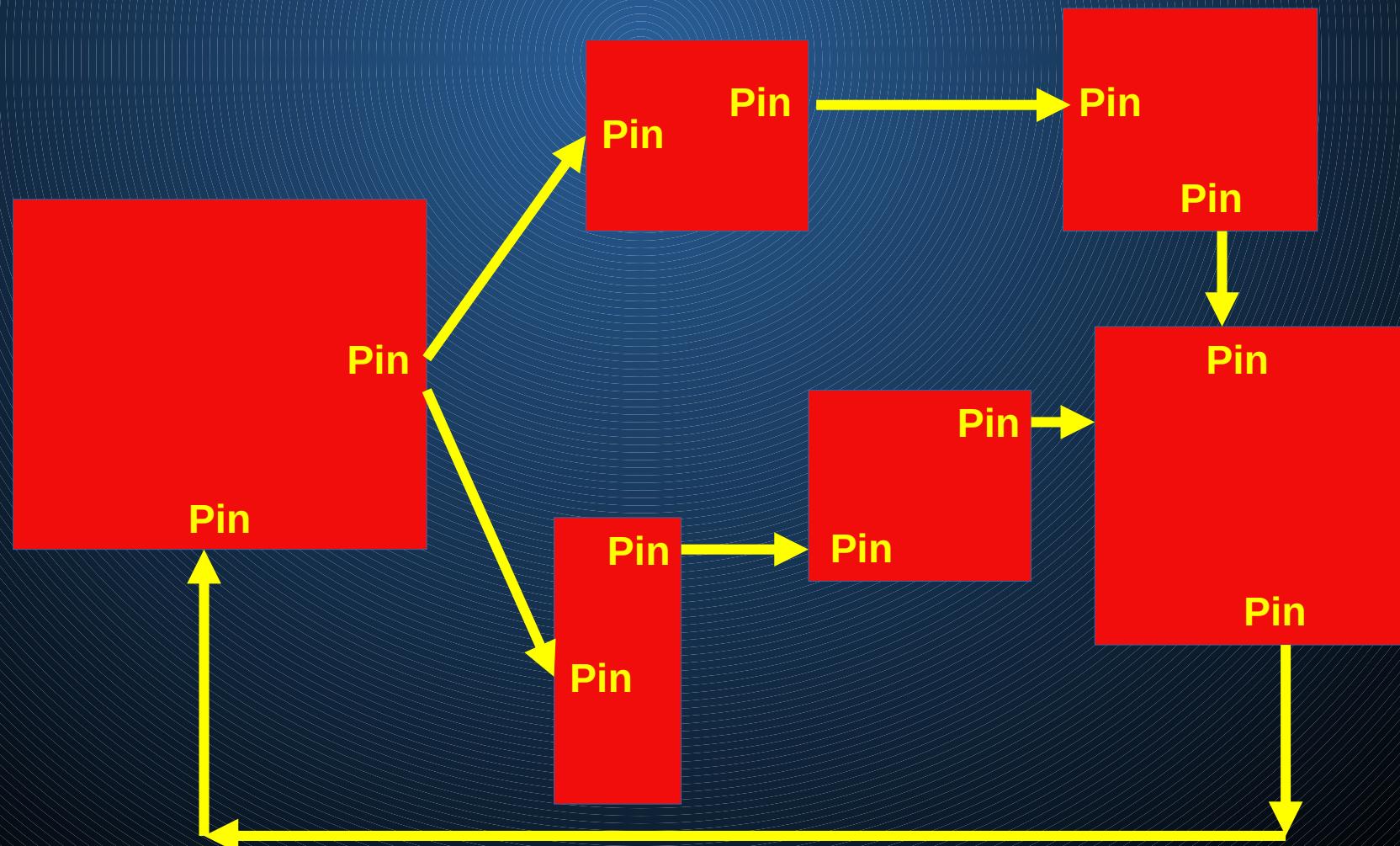
# Black Box is called a “Component”



# Connecting Wires are called “Signals”

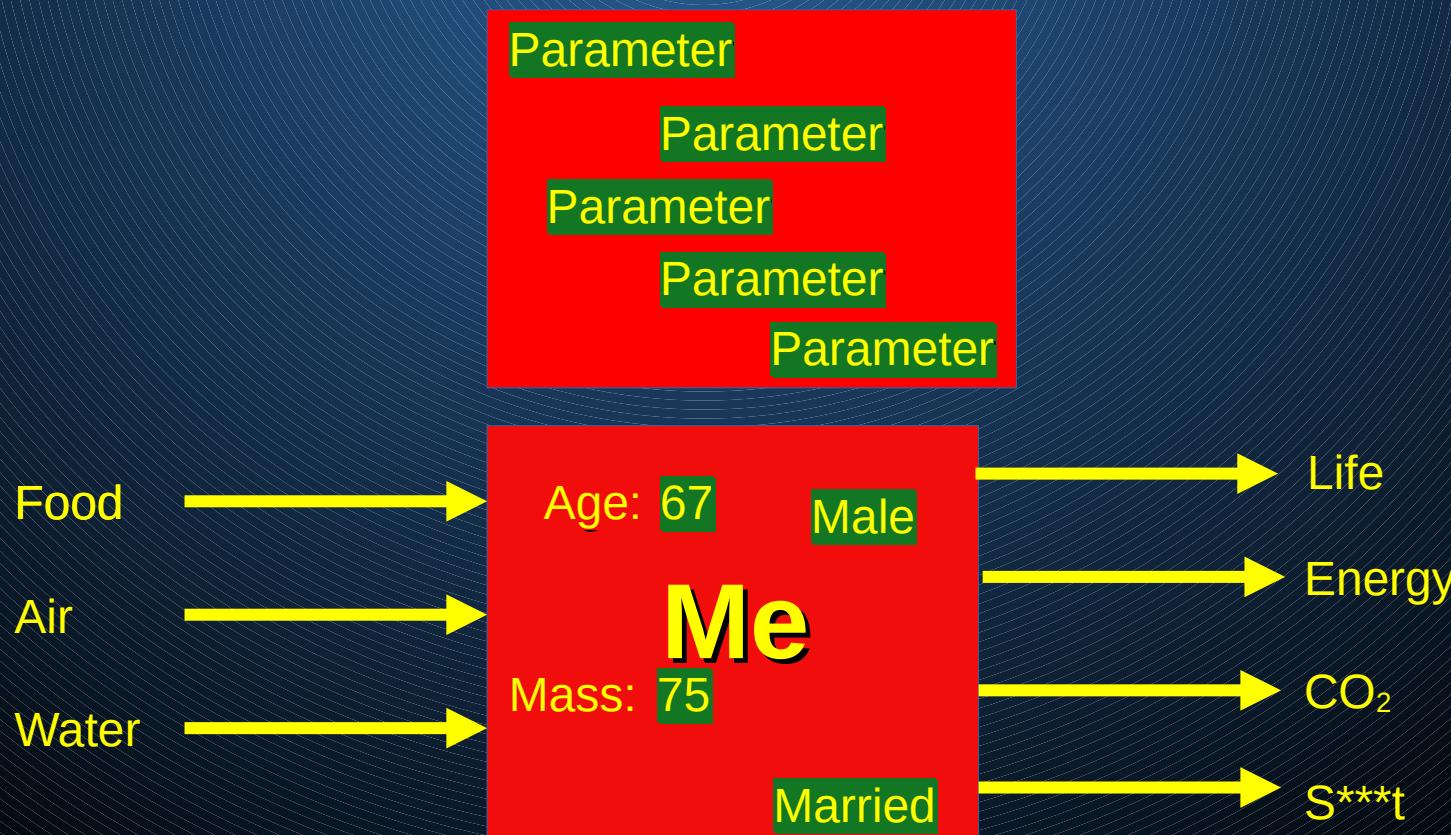


# Inputs & Outputs called Pins



# “Parameters” are Values which need Storing

Constant Values inside a Component which need storing are called “Parameters”



# “Functions”

“Functions” are the engine(s) inside a Component, which do all the work

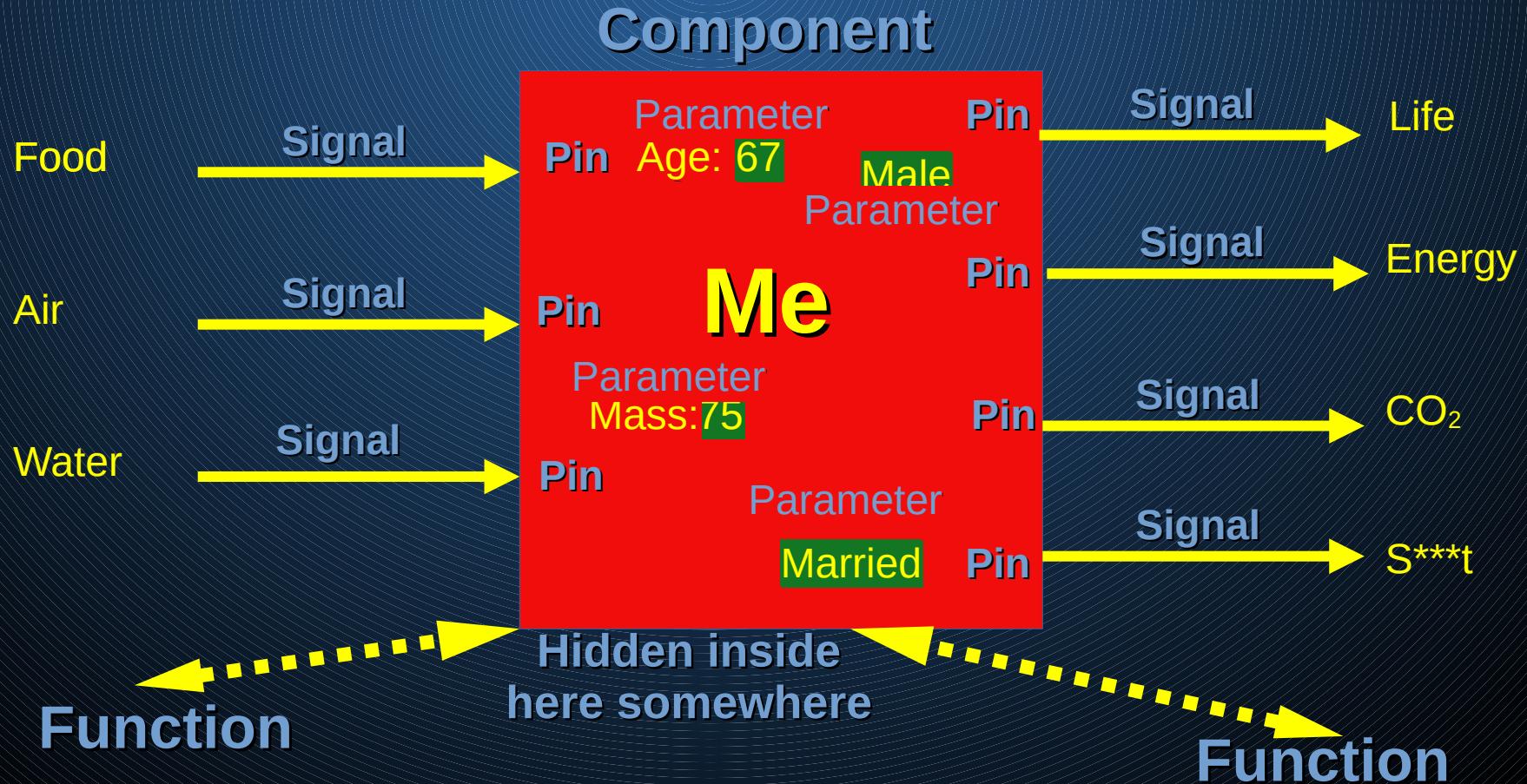
“Functions” are created inside Components by the original author of the Component

“Functions” read the input pins, do something with them, and set the output pins accordingly

“Functions” often have the same name as the component (with an appended number)

There is often only one “Function” inside a Component

# Naming Revision:



# **Pin & Signal Names look Complicated**

**but “Any Name will Do”**

**Tom**

**Dick**

**Harry**

**TomAndHarryTogether**

**Toms.Mother.in.law**

**Toms.Mother.in.law.cat**

**Toms.Mother.in.law.cat.whisker**

# LinuxCNC pin names reflect hierarchy

examp

  examp.1

    examp.1.1

  examp.2

    examp.2.1

      examp.2.1.1

        examp.2.1.1.input

        examp.2.1.1.output

        examp.2.1.1.output.noninverted

        examp.2.1.1.output.inverted

      examp.2.1.2

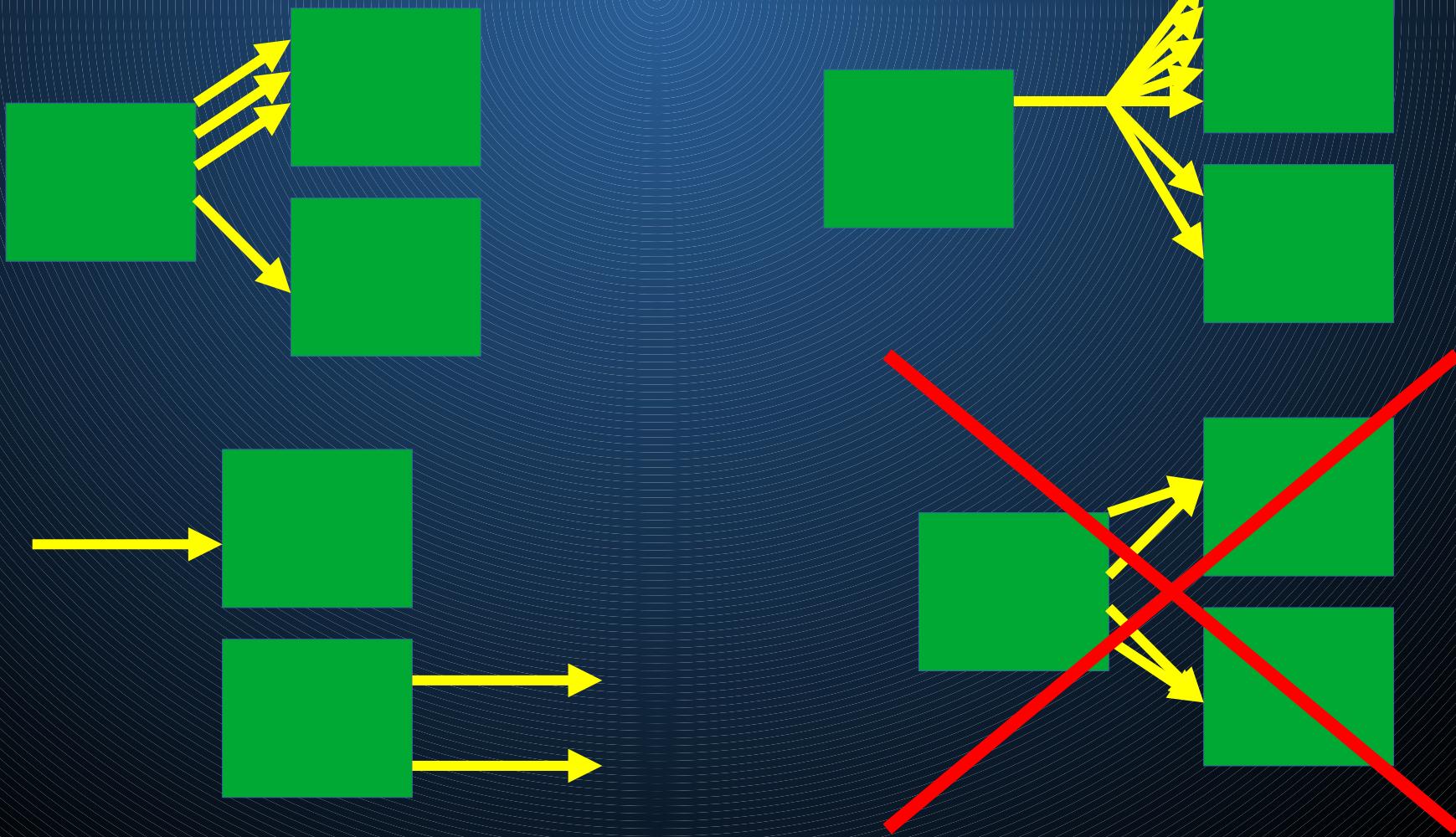
      examp.2.1.3

    examp.2.2

  examp.3

**They are just names**

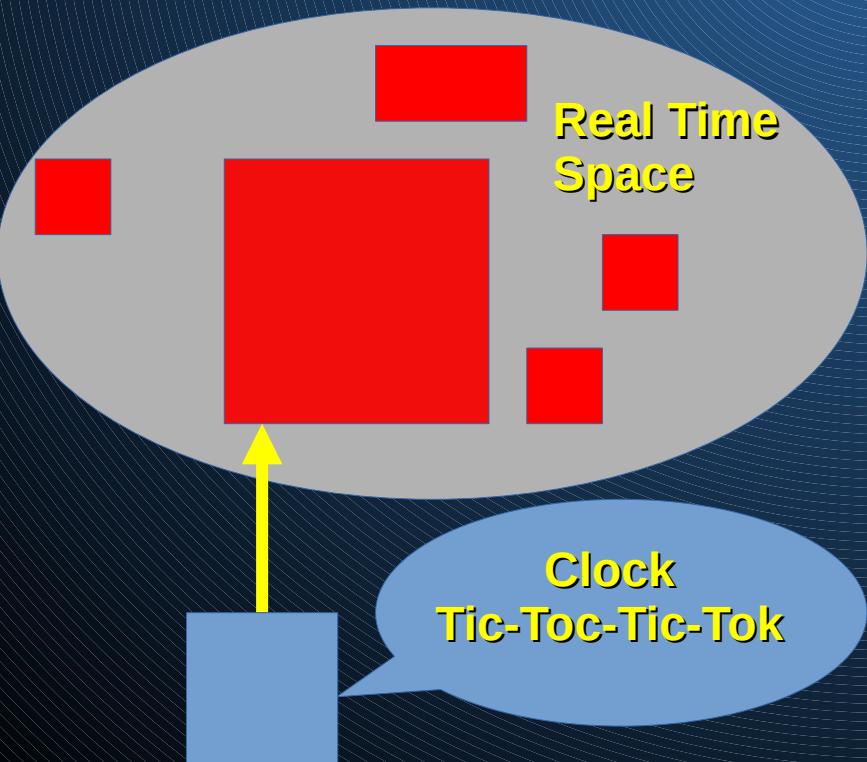
# Any (non-stupid) Connection Between Pins with Signals is Possible



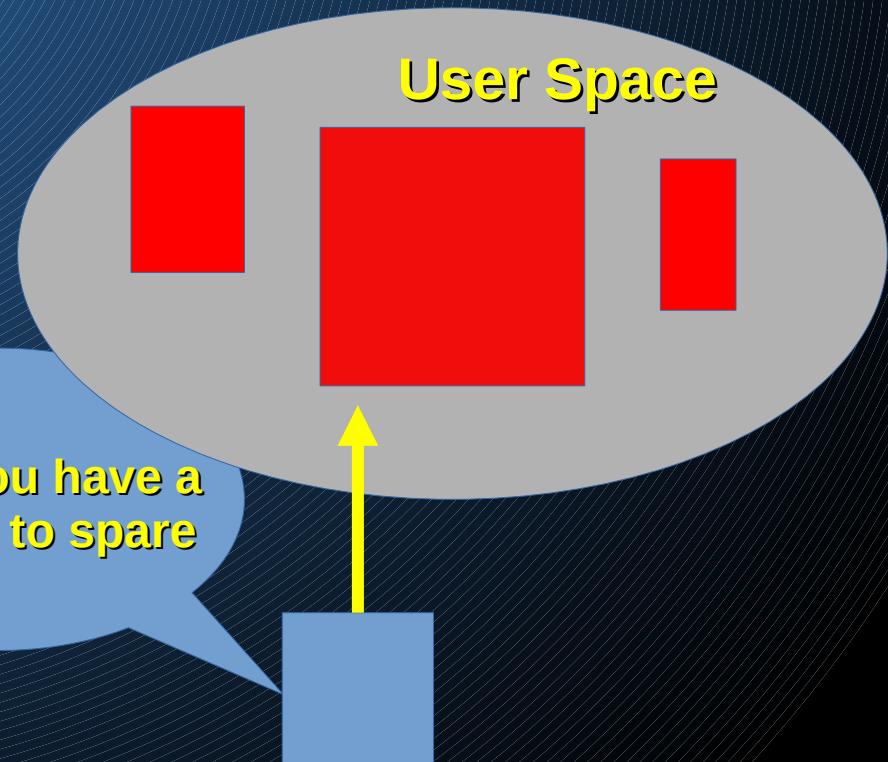
# Components get updated:

Very Regularly

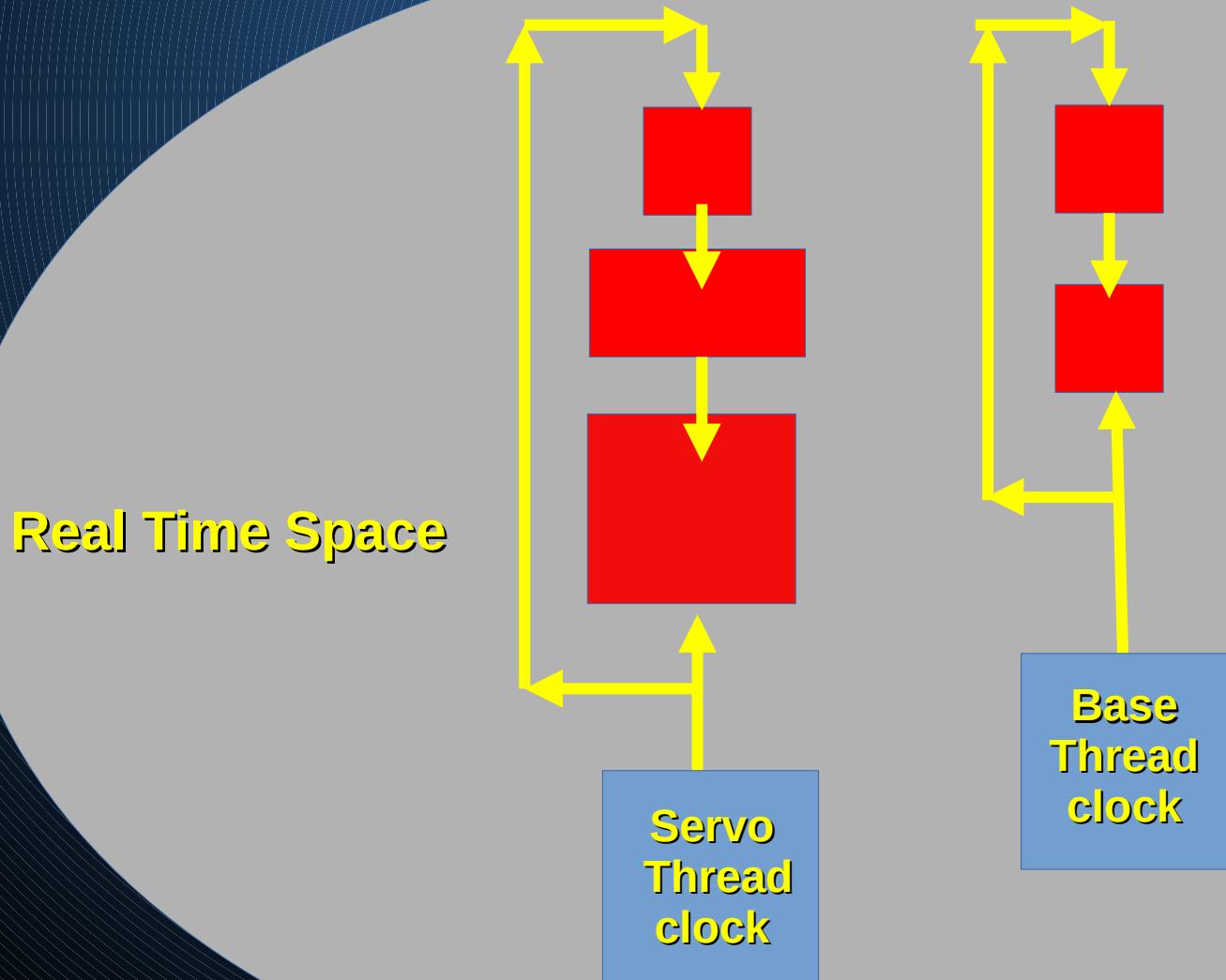
(Real Time Space)



Some Time  
Any Time  
(User Space)



# The List of functions connected to the same clock is called a “thread”:



# Revision:

**Black Box ---> Component**

**Connecting Wire ---> Signal**

**Connection Point ---> Pin**

**Value stored inside component --> Parameter**

**Engine inside a component --> Function**

**Update timing ---> Realtime / User space**

**List of functions conn. to same clock --> Thread**

# **HAL Configuration File**

**Ordinary  
Text file**

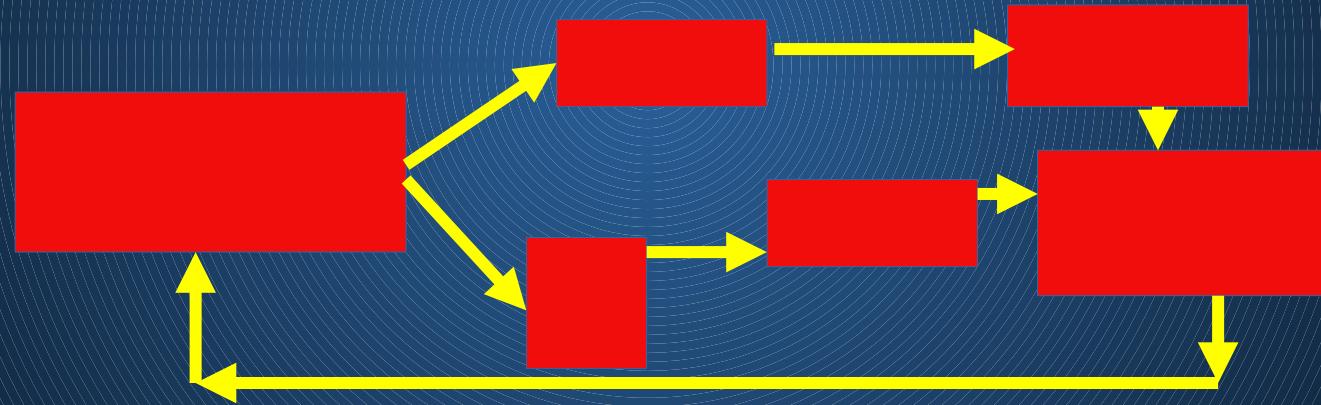
**Understanding???????**

**Editing?????????**

**Creating?????????**

**.hal  
filename  
extension**

# Load Components



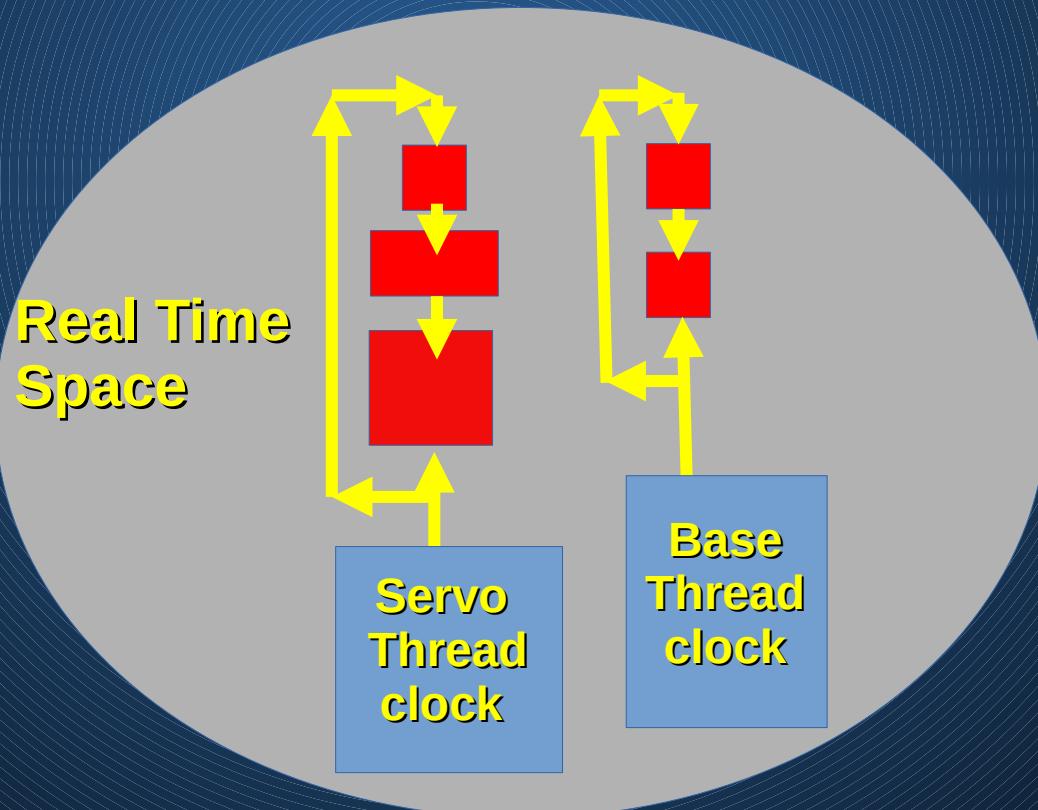
Real Time Space

**loadrt <component name> <options>**

User Space

**loadusr [-W] <component name> <options>**

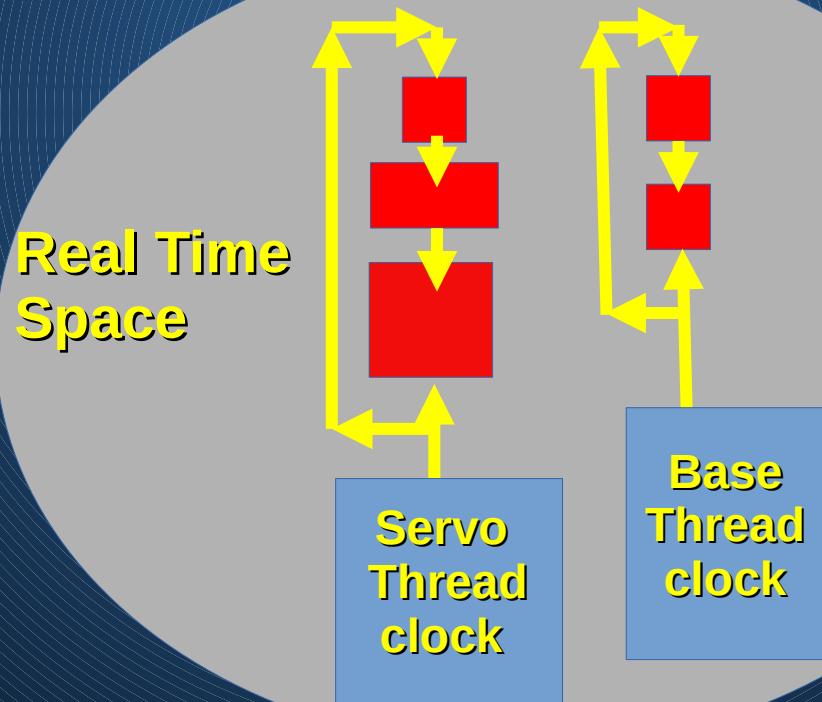
# Create the thread(s) with their clock(s)



```
loadrt motmod [base_period_nsec=<period>] [servo_period_nsec=<period>]
```

eg. `loadrt motmod base_period_nsec=1000 servo_period_nsec=1000000`

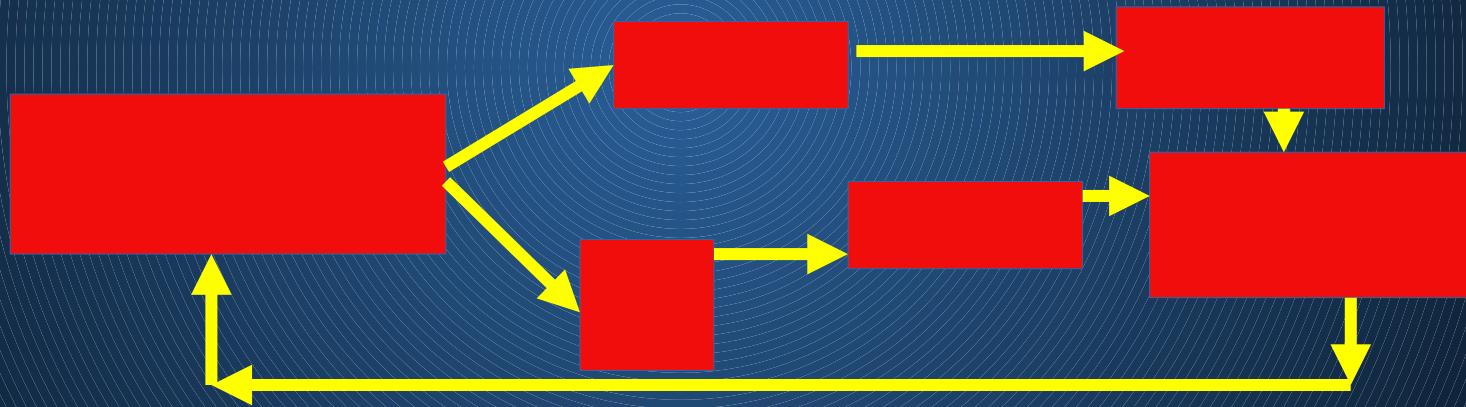
# Add the Functions inside the components to their respective Threads



**addf <function> <thread>**

eg. addf encoder.update-counters base-thread  
addf encoder.capture-position servo-thread

# Join up Pins with Signals



**net <signal name> <pin name> [<pin name>] etc.....**

**you may add => or <= to show 'in' to 'out' direction  
but HAL does not look at them**

**net commands can join any number of signals / pins  
anywhere in HAL file**

# Set the pins & parameters

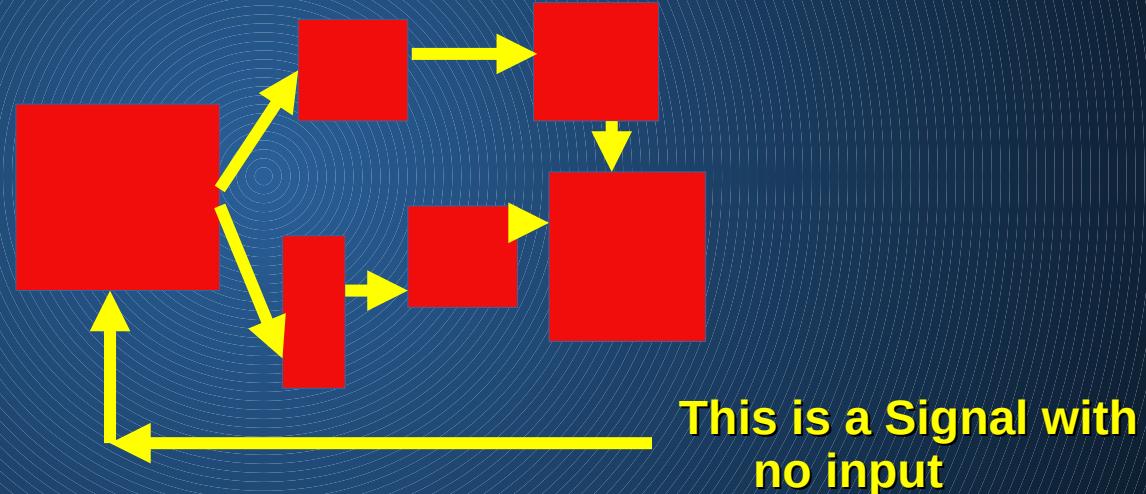


**setp <pin | parameter> <value>**

eg. **setp parport.0.pin-08-out TRUE**

**setp siggen.0.amplitude 5**

# Set a Signal



`sets <signal-name> <value>`

eg. `sets mysignal 1`

# Import Values from .ini File

The image shows two side-by-side Emacs windows. The left window displays a configuration file with several `loadrt` commands. A red circle highlights the file name `TestMill.hal` in the title bar. The right window displays an `.ini` file with sections for `[RS274NGC]`, `[EMCMOT]`, and `[HAL]`. A red circle highlights the file name `TestMill.ini` in the title bar. Both windows have a toolbar at the top with icons for file operations like Save, Undo, and Cut/Copy/Paste.

```
emacs@cnc
File Edit Options Buffers Tools Help
Save Undo Cut Copy Paste Find Replace
#####
# Load the Run-Time Components
#####

loadrt [KINS]KINEMATICS

loadrt [EMCMOT]EMCMOT base_period_nsec=[EMCMOT]BASE_PERIOD servo_
period_nsec=[EMCMOT]SERVO_PERIOD num_joints=[KINS]JOINTS

loadrt sim_parport

loadrt stepgen step_type=0,0,0

loadrt pwmgen output_type=1

#####
- :**- TestMill.hal Top L12 (Fundamenta
text document

[RS274NGC]
PARAMETER ETLE = linuxcnc.var

[EMCMOT]
EMCMOT = motmod
COMM_TIMEOUT = 1.0
BASE_PERIOD = 100000
SERVO_PERIOD = 1000000

[HAL]
HALFILE = TestMill.hal
HALFILE = custom.hal
POSTGUI_HALFILE = postgui_call_list.hal

TestMill.ini 44% L48 (Conf[Unix])
```

# Understanding .hal file - 1

```
emacs@cnc

File Edit Options Buffers Tools Help
Save Undo Cut Copy Find

#####
# Load the Run-Time Components
#####

loadrt [KINS]KINEMATICS

loadrt [EMCMOT]EMCMOT base_period_nsec=[EMCMOT]BASE_PERIOD servo_period_nsec=[EMC
MOT]SERVO_PERIOD num_joints=[KINS]JOINTS

loadrt sim_parport

loadrt stepgen step_type=0,0,0

#####
# Add functions to the base-thread
#####

addf sim-parport.0.read          base-thread
addf stepgen.make-pulses         base-thread
addf sim-parport.0.write          base-thread
addf sim-parport.0.reset          base-thread

#####
# Add functions to the servo-thread
#####

addf stepgen.capture-position    servo-thread
addf motion-command-handler      servo-thread
addf motion-controller           servo-thread
addf stepgen.update-freq          servo-thread

#####
# Spindle Stuff
#####
```

# Understanding .hal file - 2

```
emacs@cnc

File Edit Options Buffers Tools Help
Save Undo Cut Copy Find

#####
# Spindle stuff
#####

net spindle-cmd-rpm      <= spindle.0.speed-out
net spindle-cmd-rpm-abs   <= spindle.0.speed-out-abs
net spindle-cmd-rps       <= spindle.0.speed-out-rps
net spindle-cmd-rps-abs   <= spindle.0.speed-out-rps-abs
net spindle-at-speed      => spindle.0.at-speed
net spindle-cw             <= spindle.0.forward

#####
# sim-parport stuff
#####

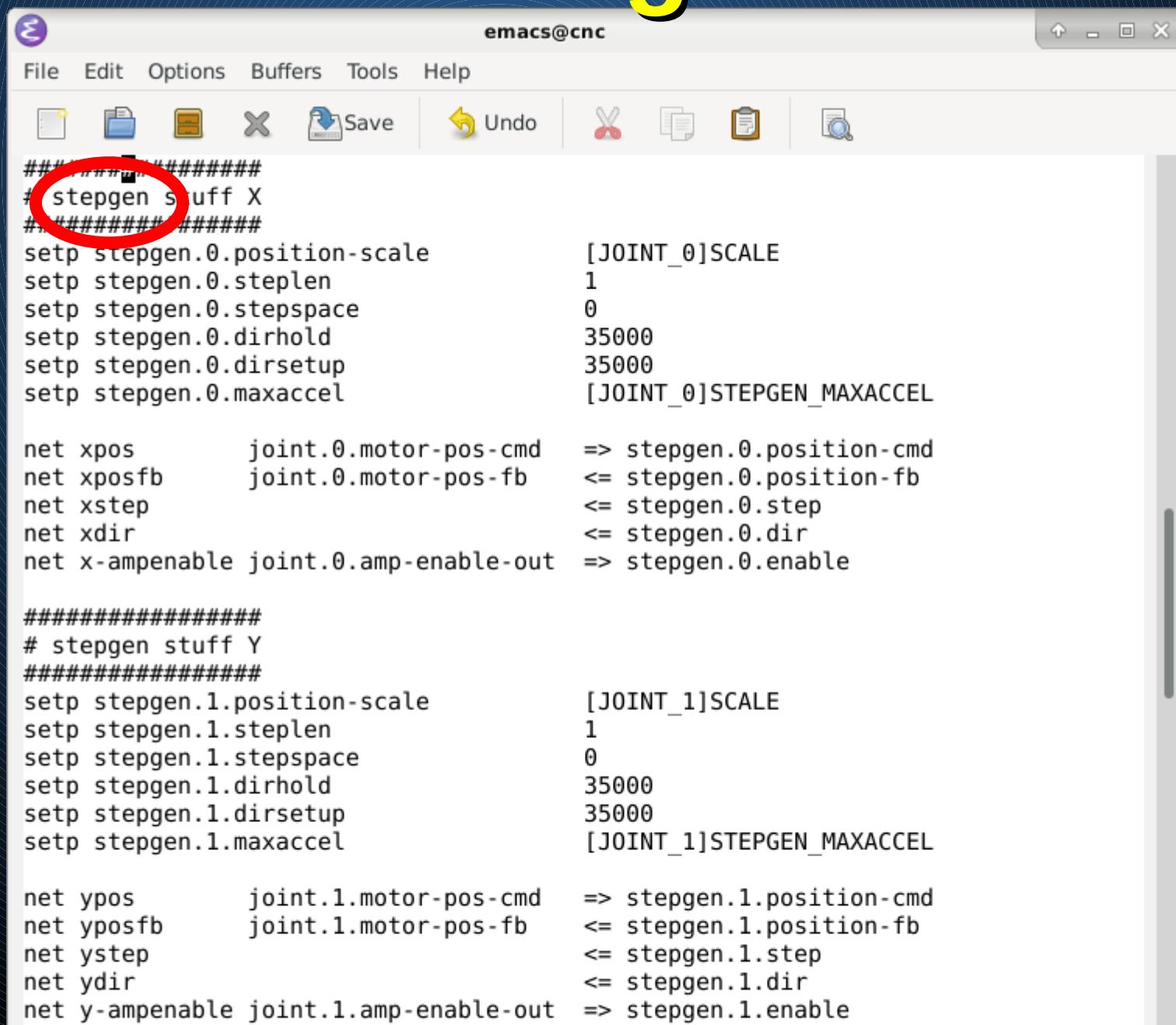
setp sim-parport.0.reset-time 5000

net estop-out      => sim-parport.0.pin-01-out
net xstep          => sim-parport.0.pin-02-out
net xdir           => sim-parport.0.pin-03-out
net ystep          => sim-parport.0.pin-04-out
net ydir           => sim-parport.0.pin-05-out
net zstep          => sim-parport.0.pin-06-out
net zdir           => sim-parport.0.pin-07-out
net astep          => sim-parport.0.pin-08-out
net spindle-cw     => sim-parport.0.pin-14-out

setp sim-parport.0.pin-02-out-reset 1
setp sim-parport.0.pin-04-out-reset 1
setp sim-parport.0.pin-06-out-reset 1
setp sim-parport.0.pin-08-out-reset 1

#####
```

# Understanding .hal file - 3



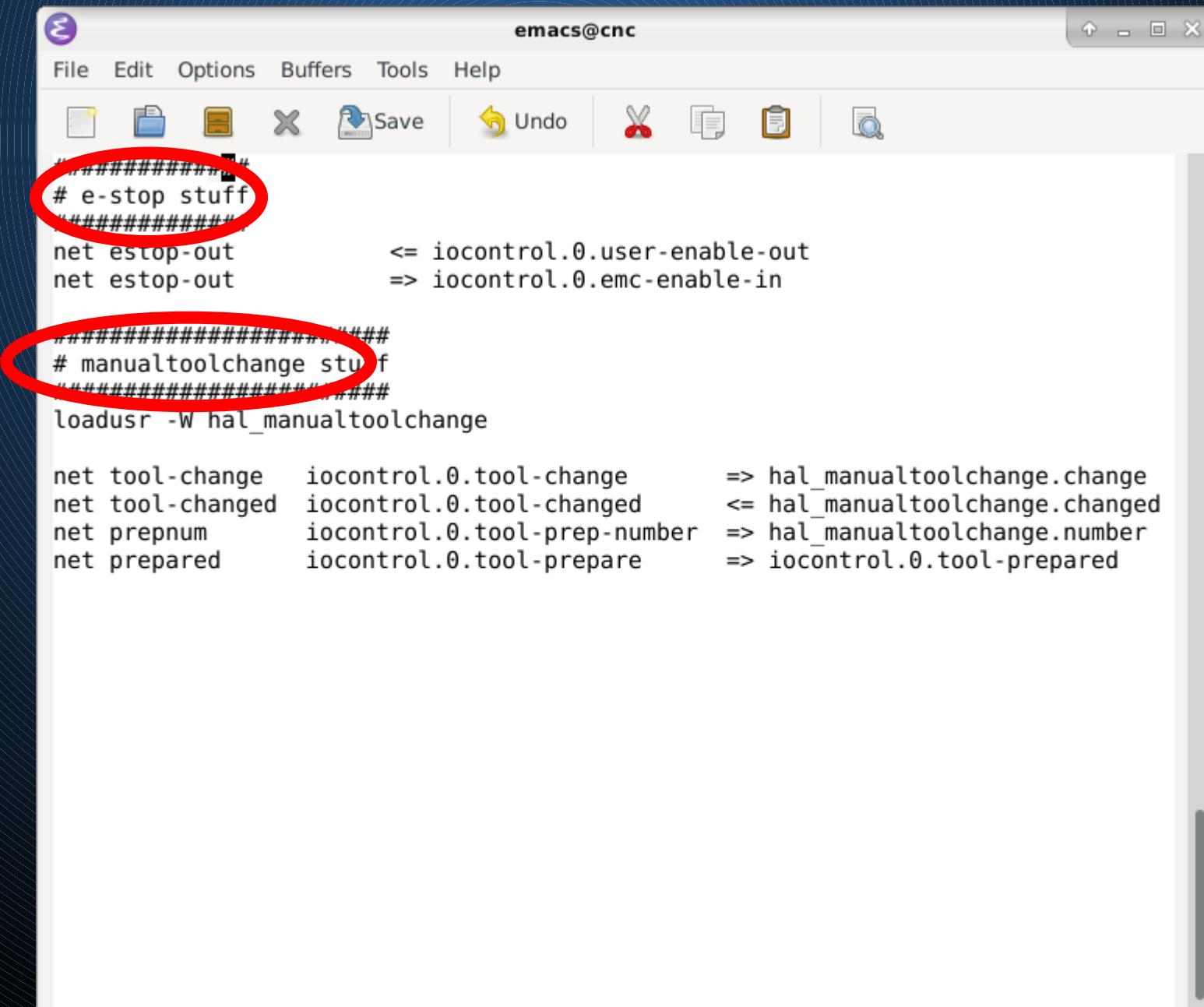
```
emacs@cnc
File Edit Options Buffers Tools Help
Save Undo
#####
# stepgen stuff X
#####
setp stepgen.0.position-scale      [JOINT_0]SCALE
setp stepgen.0.steplen             1
setp stepgen.0.stepspace            0
setp stepgen.0.dirhold              35000
setp stepgen.0.dirsetup             35000
setp stepgen.0.maxaccel             [JOINT_0]STEPGEN_MAXACCEL

net xpos          joint.0.motor-pos-cmd  => stepgen.0.position-cmd
net xposfb        joint.0.motor-pos-fb   <= stepgen.0.position-fb
net xstep          joint.0.step           <= stepgen.0.step
net xdir           joint.0.dir            <= stepgen.0.dir
net x-ampenable    joint.0.amp-enable-out => stepgen.0.enable

#####
# stepgen stuff Y
#####
setp stepgen.1.position-scale      [JOINT_1]SCALE
setp stepgen.1.steplen             1
setp stepgen.1.stepspace            0
setp stepgen.1.dirhold              35000
setp stepgen.1.dirsetup             35000
setp stepgen.1.maxaccel             [JOINT_1]STEPGEN_MAXACCEL

net ypos          joint.1.motor-pos-cmd  => stepgen.1.position-cmd
net yposfb        joint.1.motor-pos-fb   <= stepgen.1.position-fb
net ystep          joint.1.step           <= stepgen.1.step
net ydir           joint.1.dir            <= stepgen.1.dir
net y-ampenable    joint.1.amp-enable-out => stepgen.1.enable
```

# Understanding .hal file - 4



```
emacs@cnc

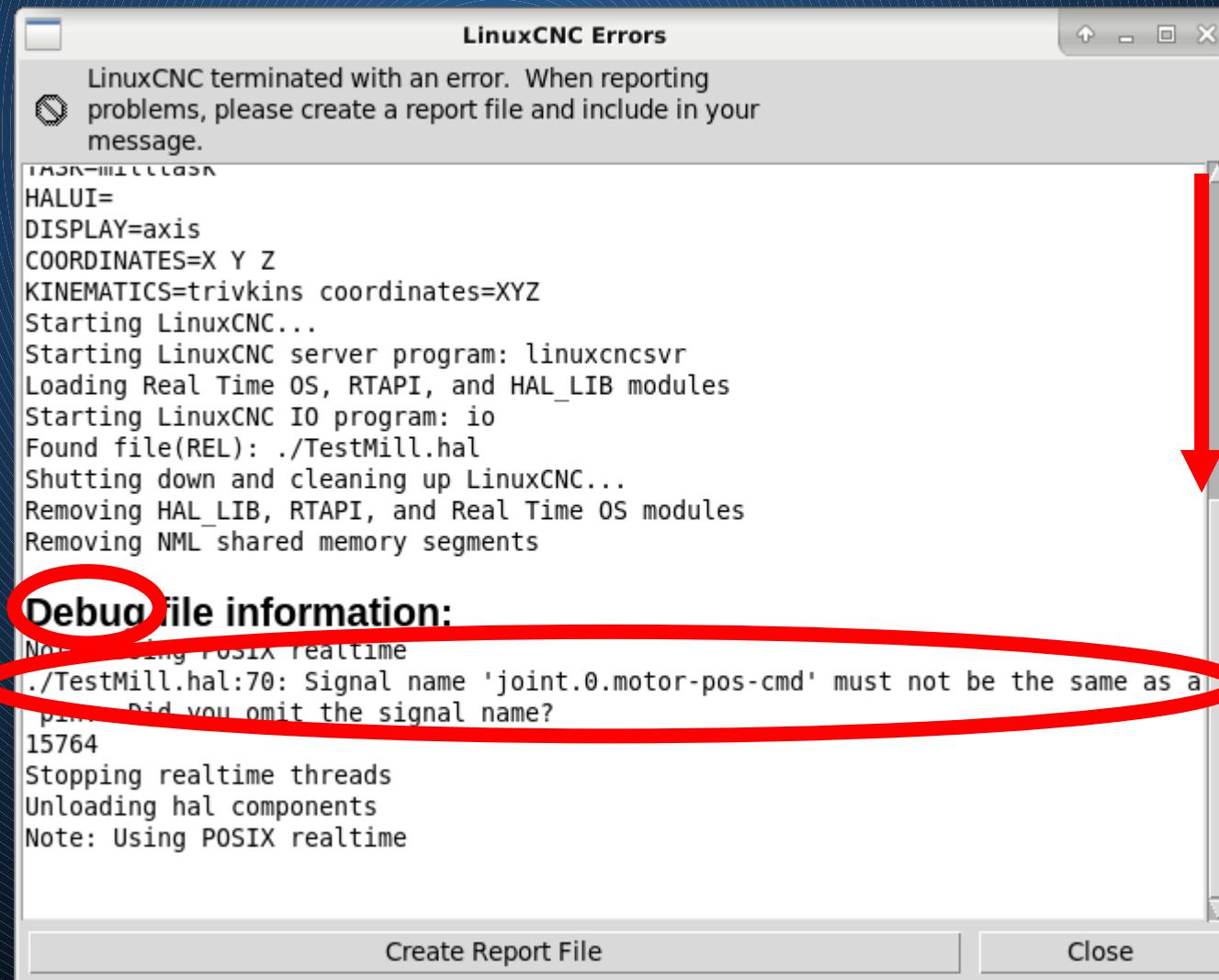
File Edit Options Buffers Tools Help
Save Undo

#####
# e-stop stuff
#####
net estop-out      <= iocontrol.0.user-enable-out
net estop-out      => iocontrol.0.emc-enable-in

#####
# manualtoolchange stuff
#####
loadusr -W hal_manualtoolchange

net tool-change    iocontrol.0.tool-change      => hal_manualtoolchange.change
net tool-changed   iocontrol.0.tool-changed    <= hal_manualtoolchange.changed
net prepnum        iocontrol.0.tool-prep-number => hal_manualtoolchange.number
net prepared       iocontrol.0.tool-prepare     => iocontrol.0.tool-prepared
```

# Errors in .hal file



# **Creating .hal File:**

## **Draw a Diagram!**

# stepgen

0 dir-hold 3500  
dir-setup 3500  
frequency  
maxaccel ini  
maxvel  
position-scale ini  
rawcounts

1

2

capture-position

update-freq

make-pulses

steplen 1  
stepspace 0

position-cmd  
position-fb  
counts

step  
dir

enable

xpos

counts

# parport

xstep → parport.0.pin-02-out  
xdir → parport.0.pin-03-out  
x-ampenable

# motmod

joint.0.motor-pos-cmd  
joint.0.motor-pos-fb  
joint.0.amp-enable-out

servo-thread

base-thread

# Writing a Component

**Next Time!**









