

# Machinekit Projects Goal and Issues

# Table of Contents

1. What should Machinekit be .....	1
1.1. Connection between userland and hardware .....	1
1.2. Reconfigurable .....	1
1.3. Hardware agnostic .....	1
1.4. Real time connections to Non-realtime .....	1
2. List .....	1
3. Diagram .....	2

This depicts the open issues and shows which ones are related.

# 1. What should Machinekit be

## 1.1. Connection between userland and hardware

Machinekit is a configurable driver stack, capable of realtime (motion control) tasks. Machinekit connects userland code with hardware drivers by means of configuring components. Machinekit is usable on different platforms like x86, x64 and ARM. Legacy LinuxCNC has been split out into `machinekit-hal` and `machinekit-cnc`.

## 1.2. Reconfigurable

The machinekit-hal stack is a reconfigurable (realtime) Hardware Abstraction Layer where control functionality can be made by configuring standard components together. Wiring them up with signals. Reconfiguring a system by instantiating or removing components, adding or deleting signals can be done on-the-fly.

## 1.3. Hardware agnostic

Various drivers in machinekit-hal provide access to the hardware. For example the MESANET PCI FPGA cards, a beaglebone black with PRU, or other platforms like the De(1)0-nano (FPGA) or the MYIR Zynq (FPGA) boards.

You can thusly configure your control system on various boards. Having the same control (behaviour) on different hardware.

## 1.4. Real time connections to Non-realtime

The realtime HAL part of machinekit can connect to the non-realtime domain in various ways:

- Thru the API by means of reading/setting scalars.
- Thru the API by means of ringbuffers.
- remote components (machinetalk) enable remote UI's.

# 2. List

this section shows a list of issues that need some additional explaining, linking to important comments in that discussion.

General:

- [API's missing \(99\)](#)
- [Design error: command serials may collide \(114\)](#)

- Interpreter API: rearchitecting required (106)
- HAL remote component protocol: firm up security aspects #204
- triggered RT execution of HAL function chains missing #219
- we need a regression test framework which actually works. runtests has to go. #220
- deprecate runtests for good. #221
- HAL thread creation API: enable triggered and controlled-environment execution (eg simulated-time) #285
- HAL API Changes #237
- unit test for delayline component #713

CNC stack:

- GCode preview speed/data size/quality too slow #134
- more flexible EmcPose representation #151
- joints-axes branch not yet merged #438

not looked at:

- firmware path handling : needs work #24
- document HAL\_STACKSIZE config param and merge into framesize checks #30
- runtest for pci logic in userland threads #47
- hal.kernel\_version missing in non RTAPI versions #102
- Configuration & distributed setups: INI files wont get us there #104
- Switch ARM configurations to velocity-mode #131
- watchdog functionality #210

discuss of these are solved/obsolete

- out of memory/large files results in stopped LinuxCNC #193

### 3. Diagram

