

(Ab)using Continuous Integration for Hardware Design

Continuous Integration?

- Traditional flow
- Long cycle
- Testing is boring!
- Every developer needs full build toolchain for every project

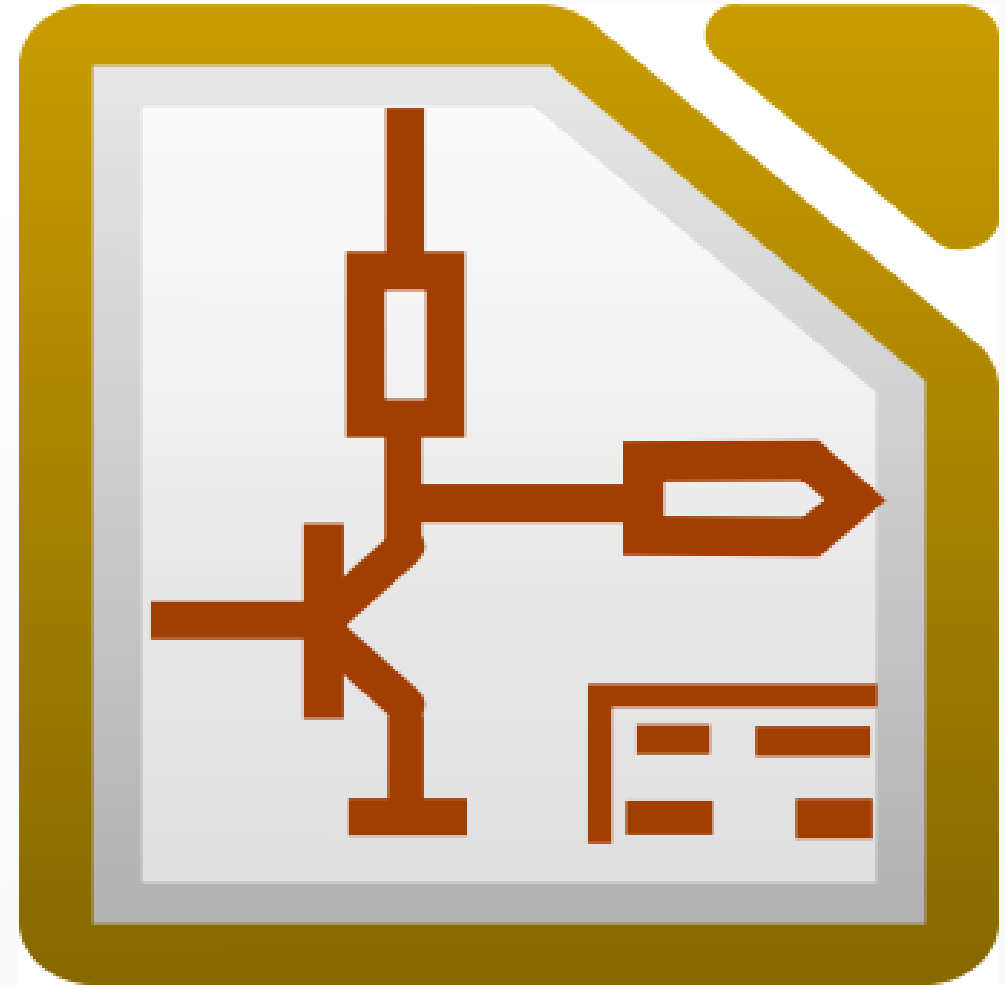
1. Checkout from source control
2. Modify code
3. Compile/build
4. Test
5. Check in

Continuous Integration?

- CI workflow
 - Server does the boring testing (well some of it...)
 - Always built with the correct toolchain
 - Can't skip any tests
 - Sends nagging emails
1. Checkout from source control
 2. Modify code
 3. (Quick sanity check compile)
 4. Commit changes
 5. CI Build/test triggered by commit

KiCad www.kicad-pcb.org

- Started in 1992 as a “learn to code” project...
- Now maintained by a team sponsored through CERN by RPi, Arduino...
- Worth £1000s per seat



KiCad

- Superb quality footprint libraries
- Includes Python3 scripting interface
- Industry standard manufacturing file output
- Runs natively on Linux
- Successfully built & shipped 1000s of commercial boards 2-layer, 4-layer RF & mixed signal

Compile a PCB design?

- GERBER files – binary mask for each layer
- Drill files – X, Y and size for each hole to be drilled
- Schematic – diagram of connections for fault finding
- Fab notes – PCB stackup and notes for manufacturer
- Bill of Materials (BOM) – list of parts to fit
- Pick & Place files – X, Y and rotation for each part
- Netlist, CAD files etc – for automated PCB test
- Fairly simple board can be 20 files...
- Automating the process guarantees reproducible version control tag

Automating

- Python scripts
- GERBER files & Drill files easy – use Python API
- Schematic outputs (PDF, BOM) – not so easy use GUI automation
- Fabnotes, P&P, CAD – also GUI automation

Automation

- Use Xvfb so it works in a “headless” environment (e.g. docker)
- Use xdotool to send “Ctrl-P”, “Tab” type key commands to navigate GUI
- Use recordmydesktop for debugging
- Based on Scott Bezek’s work on KiCad 4:
<https://scottbezek.blogspot.com/2016/04/automated-kicad-openscad-rendering.html>

Testing

- Automatically check for missing Pick & Place lines
- Missing BOM information
- TODO: DRC (no way to manage exceptions yet)
- TODO: ERC (as above)

Loads more detail...

- Check out the details on Github
 - <https://github.com/hairymnstr/ndkicadlibrary>
- Example Project
 - <https://github.com/hairymnstr/oggbox>