- Create directory tree to store projects/components in
- Launch KiCad
- Create a New Project
  - Browse to directory created above, and type in project name.
  - When asked to create a new empty directory, say yes.

- Launch Schematic Library Editor
  - http://kicadhowto.wikidot.com/mcf1comp1

  - Click on OpAmp like icon "Create a new component"
  - Call the component TestPoint
  - Change designation to TP
  - Click OK
  - Zoom in on origin - you should TestPoint and TP? stacked on top of each other
  - Menu Place->Pin
  - Click at origin (or whereever)
  - Give Pin name or 1, Pin Number of 1, Change electrical type to passive
  - Click OK
  - Move mouse around and click at final destination
  - Hover over "TestPoint" and press M
  - Move "TestPoint" above line
  - Move TP? below
  - Click on book icon (right of center) "Save current component to new library"
  - Create a folder (say "dh-lib"
  - Browse into folder
  - Save library, probably renaming to dh-lib.lib (using my initials)
  - Close Library editor

- Launch Footprint editor
  - File->New Footprint
  - Move Label and reference out of the way
  - Under Drill change Size X to 0.8128mm (0.32in)
  - On left change SizeX to 1.6mm
  - File->Save Footprint in new library
  - Choose kicad directory
  - Set active library
  - resave - Save footprint in active library
  - Launch PCBnew
  - Preferences->Footprint library Wizard
    - files on my computer
    - Choose test-lib.pretty directory
  - Quit kicad (seems to be needed before it will recognize the new path we just added)

Launch eeshema

- Menu Preferences->Component Libraries
- In "User defined search path" click Add button
- Browse to dh-lib directory created previously
- Click OK
- Click Yes to "Use a relative path?"
- Click OK
- Menu Preferences->Component Libraries
- Click Add in the "Component Library Files" section
- Browse to the dh-lib directory and choose dh-lib.lib
- Click Add
- Browse to dh-lib directory
- Click on dh-lib.lib
- You need to scroll to the bottom of the list and you should see
  dh-lib added with no path.
- Click OK

Now that the component library has been saved - lets create a schematic

- Add pin headers (CONN_01X02, CONN_01X03)
- mention visibility of field on components
- Add pullup resistor, switch (R, SW_PUSH)
- Wire things up - leave resistor not connected to VCC
- Run ERC
- Annotate
- Run ERC
- Place->Power Port and add VCC for pullup resistor
- Run ERC
- Add PWR_FLAGs
- Run ERC
- Export NET

Launch CvPCB to assign footprints to all components

- Pin_Header:Pin_Header_Straight_1x02
- Pin_Header:Pin_Header_Straight_1x03
- Discreet:R4

- Now we'll go and create a footprint for the pushbutton
- Launch Footprint editor
- File->New Footprint and name it SW_PUSH-2pin-6.5mm
- Move label and value out of the way
- Place a pad at the origin and a second one 6.5mm away
- Edit both pads and change Drill Size X to 0.8128mm (0.32in)
- Change Size X to 1.6mm
- Add a silkscreen box
- Tweak position of label and ref
- Save footprint in active library

Switch back to CvPCB and assign SW_PUSH-2pin-6.5mm to our push button
Switch back to eeSchema and save netlist

Run PCBnew

- Read netlist
- Right click on background, Global Spread and Place
- Position components, talk about single sided
- Go into Design Rules and set track width and clearance
  - Width 0.032 Clearance 0.019 Via Diam 0.064 Via Drill 0.032
- Set layer to F.Cu
- Draw a few tracks
- Leave something unconnected
- Run DRC
- List unconnected tracks
- Add board outline

- show editing footprint (perhaps changing hole diameter)

Launch Gerber Viewer
Launch gerbv as comparison

Show Nanino8 schematic and board layout