

---

# **Horizon-EDA dokumentace programu**

***Vydání 1.0 CZ***

**Lukas K.**

**17.11.2019**



---

## Obsah:

---

<b>1</b>	<b>Přehled funkcí</b>	<b>3</b>
<b>2</b>	<b>Interaktivní manipulátor</b>	<b>13</b>
<b>3</b>	<b>Editor desky plošných spojů DPS (PCB)</b>	<b>15</b>
<b>4</b>	<b>Editor schémat</b>	<b>17</b>
<b>5</b>	<b>Sestavení (kompilace) programu na operačním systému Windows</b>	<b>19</b>
<b>6</b>	<b>Sestavení (kompilace) programu na operačním systému Linux</b>	<b>21</b>
<b>7</b>	<b>Pravidla</b>	<b>23</b>
<b>8</b>	<b>Vytvoření pouzdra součástky</b>	<b>25</b>
<b>9</b>	<b>Kopírování rozmístění a umístění</b>	<b>27</b>
<b>10</b>	<b>Parametry programu</b>	<b>29</b>
<b>11</b>	<b>Použití příkazového řádku (CLI Command Line Interface)</b>	<b>33</b>
<b>12</b>	<b>Začínáme</b>	<b>35</b>
<b>13</b>	<b>Pájecí místa / plošky (Padstacks)</b>	<b>39</b>
<b>14</b>	<b>Popis funkce</b>	<b>41</b>
<b>15</b>	<b>Fond knihovny součástek, pouzder, pájecích míst a schematických značek (Pool)</b>	<b>43</b>
<b>16</b>	<b>Správce fondu (Pool manager)</b>	<b>47</b>
<b>17</b>	<b>Zpětné úpravy (Backanotation)</b>	<b>51</b>

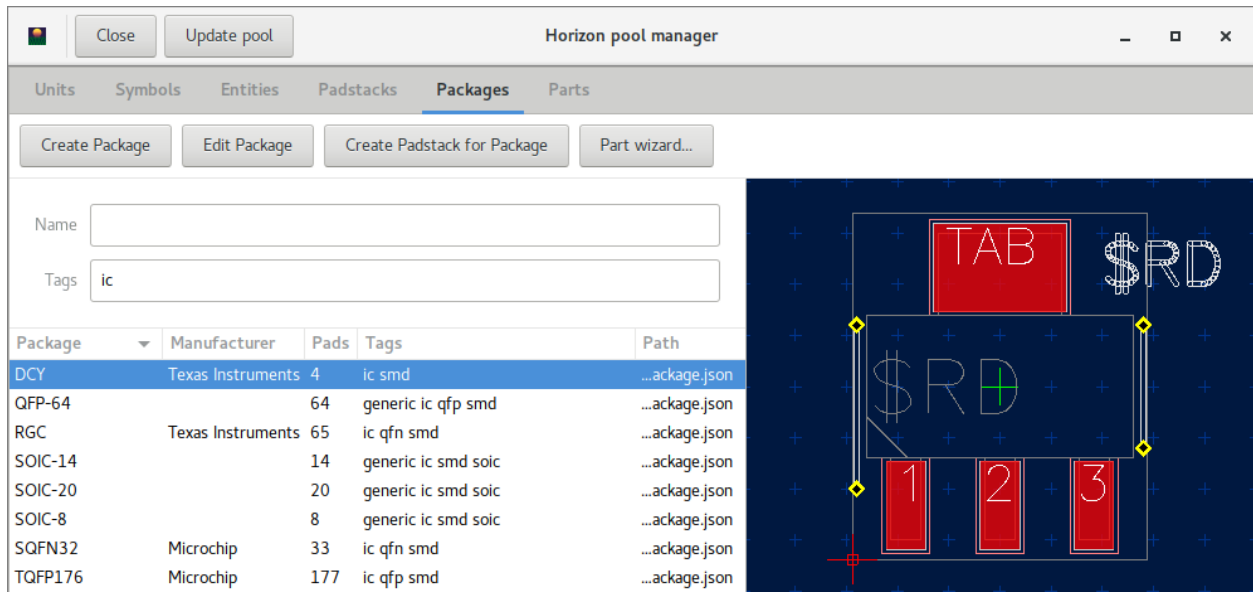






### 1.1 Interaktivní a jednoduchý management součástí

Snadná správa součástí, pouzder součástek a symbolů schemat pomocí správce fondu knihovny součástí (Pool manager):



Přiřazení vývodů k pájecím ploškám v editoru součástí (Part editor):

Save

Part Editor

×

MPN

LTC2400CS8

Inherit

Value

Inherit

Manufacturer

Linear Technology

Inherit

Tags

adc ic

Inherited:

Inherit

Entity

LTC2400 / Linear Technology

Package

SOIC-8 /

Base part

none

Parametric data

Copy from base

{}

Gate	Pin	Mapped
Main	~CS	✓
Main	Fo	✓
Main	GND	✓
Main	SCK	✓
Main	SDO	✓
Main	Vcc	✓
Main	Vin	✓
Main	Vref	✓

0 pins not mapped

Map

Unmap

Pad	Gate	Pin
1	Main	Vcc
2	Main	Vref
3	Main	Vin
4	Main	GND
5	Main	~CS
6	Main	SDO
7	Main	SCK
8	Main	Fo

0 pads not mapped

## 1.2 Snadné vytváření součástí

Jednoduše přidejte vývody tak, jak jsou uvedeny v katalogovém listu součástky:



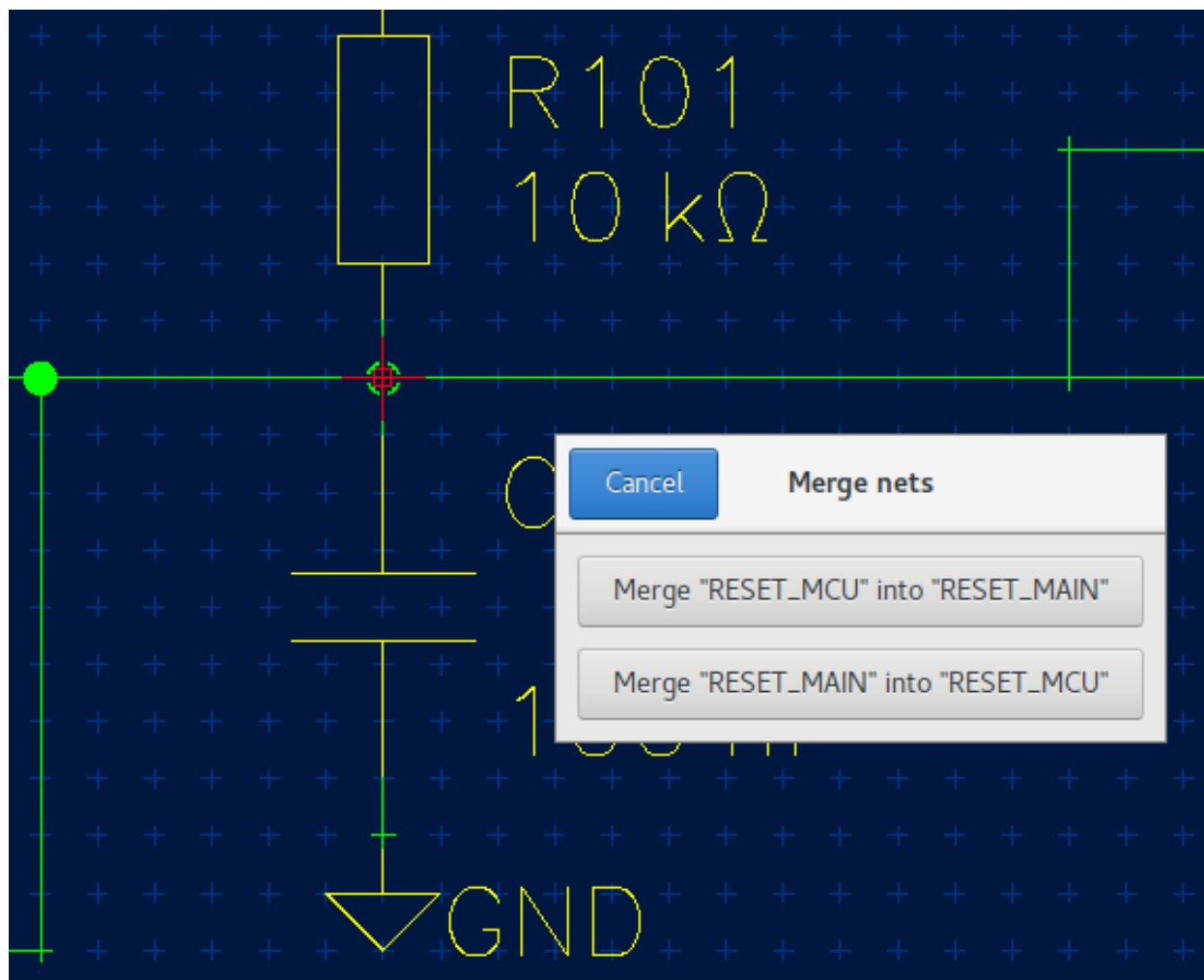
Next
Part Wizard
x

1	RESET	Input	Alt. Names	Gate	Main
2	D	Input	Alt. Names	Gate	Main
3	CLK	Input	Alt. Names	Gate	Main
4	~CLK	Input	Alt. Names	Gate	Main
5	Vee	Power Input	Alt. Names	Gate	Main
6	~Q	Output	Alt. Names	Gate	Main
7	Q	Output	Alt. Names	Gate	Main
8	Vcc	Power Input	Alt. Names	Gate	Main

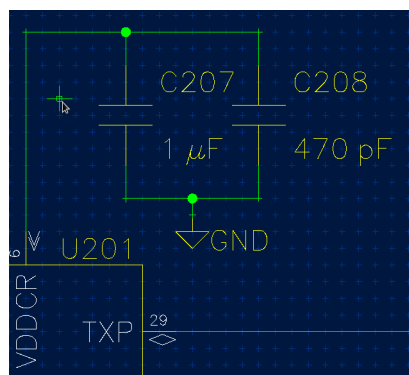
Horizon-EDA se dodává také se šablonami skriptů pro import průmyslového standardu ve formátech jako je IBIS, který vám ušetří zdlouhavou práci při opisování toho, co je v katalogovém listu.

### 1.3 Schematický editor, který ví co děláte

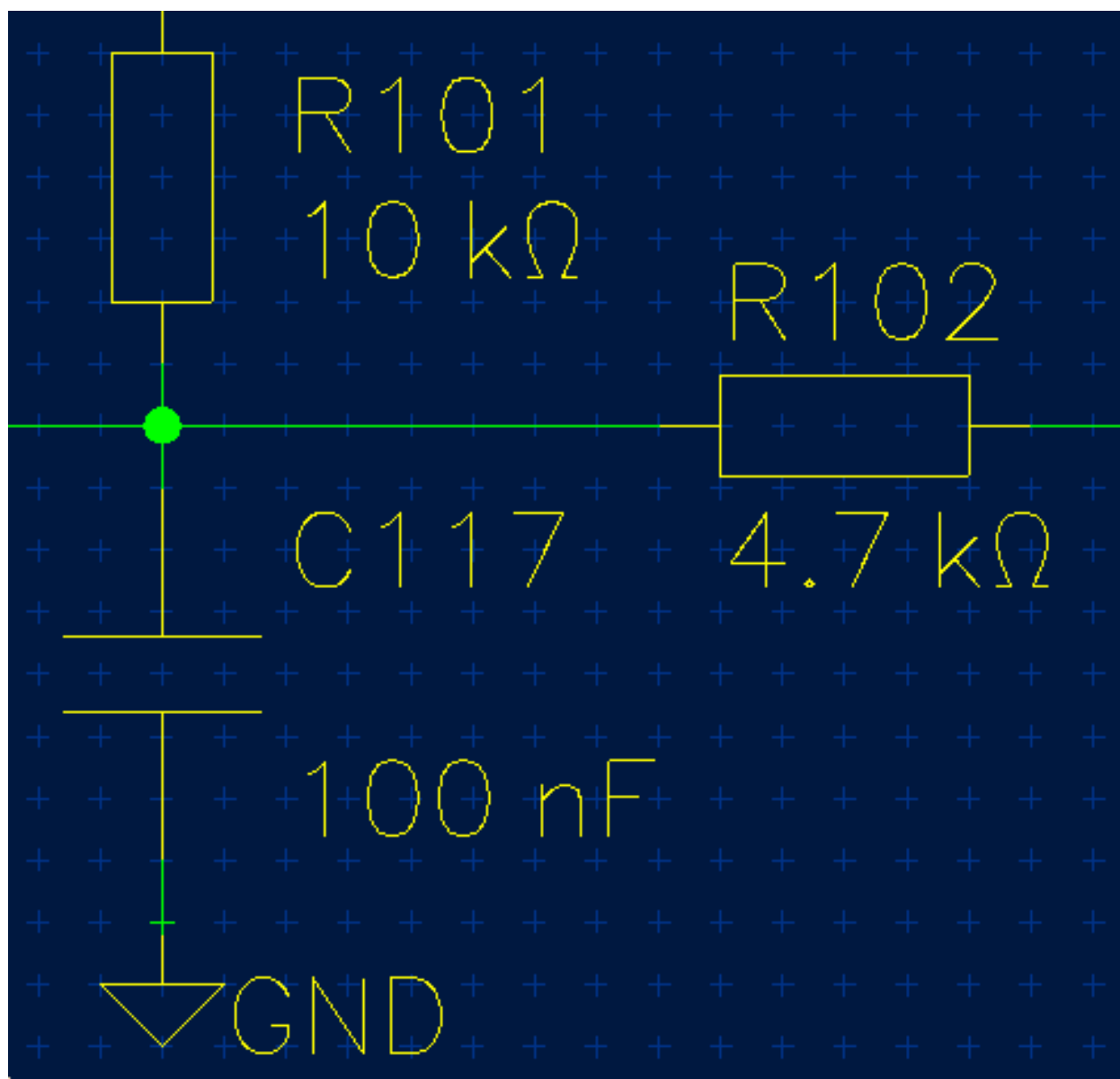
Kreslení schemat není jen o linkách a štítcích. Schéma v editoru Horizon EDA ví o jednotlivých spojích a zeptá se vás na jejich sloučení:



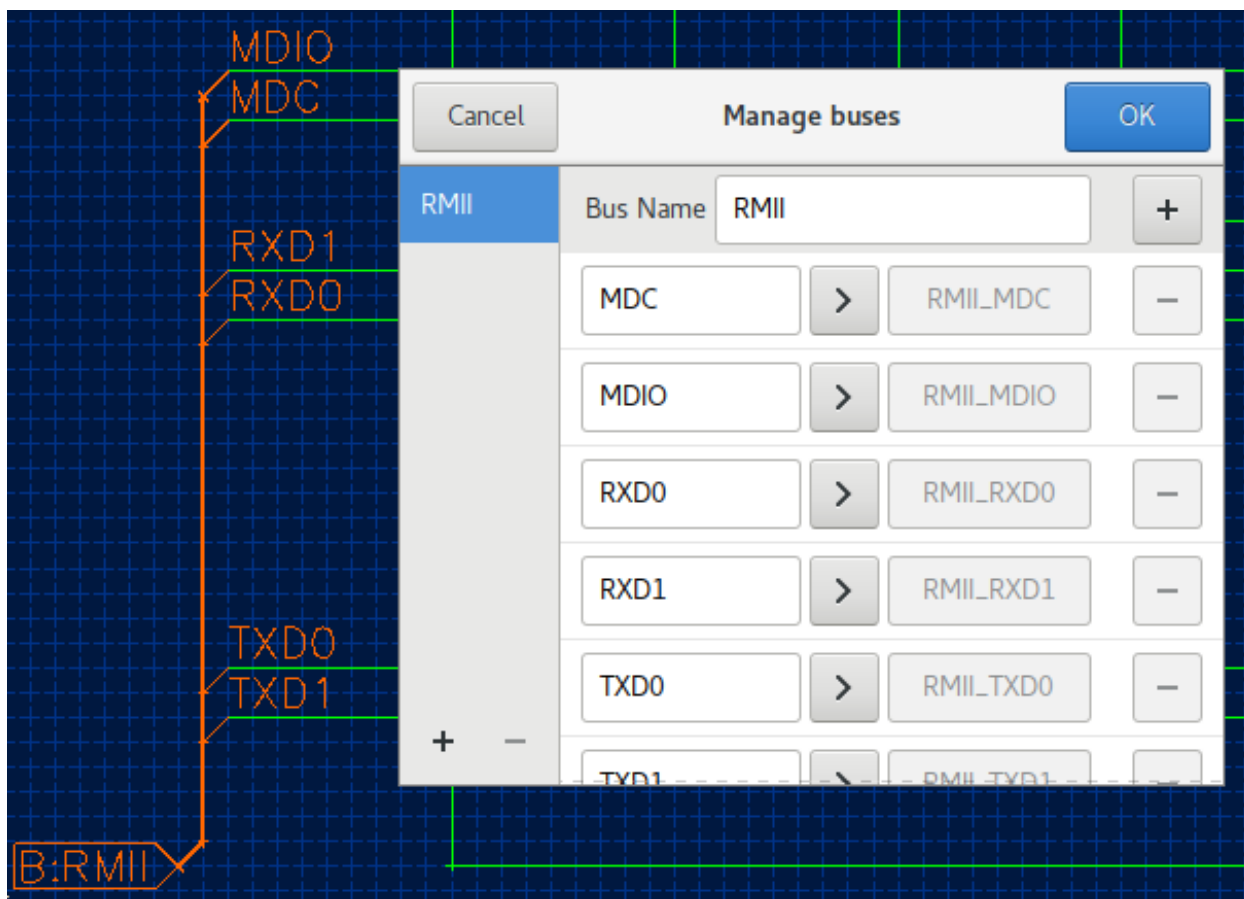
Umístí uzly tam, kde by měly být:



Také automaticky přeorientuje texty, takže se tím vyhnete těžko čitelnému referenčnímu označení:

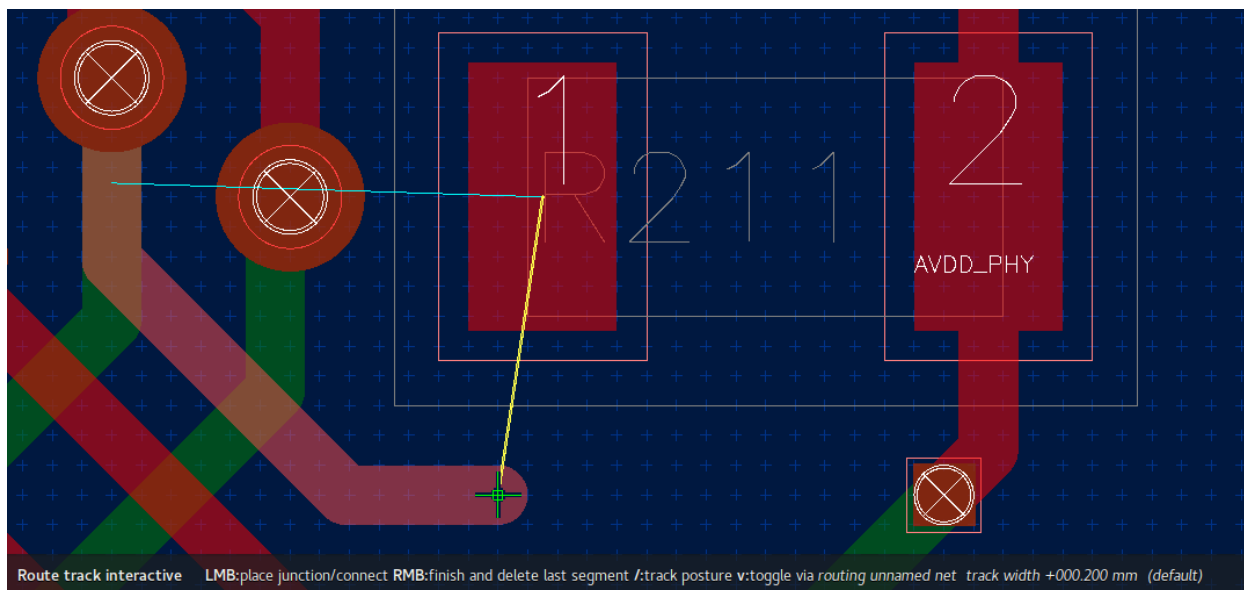


Ani sběrnice nejsou Horizonu cizí:



## 1.4 Interaktivní trasér spojů s přímou kontrolou pravidel (DRC)

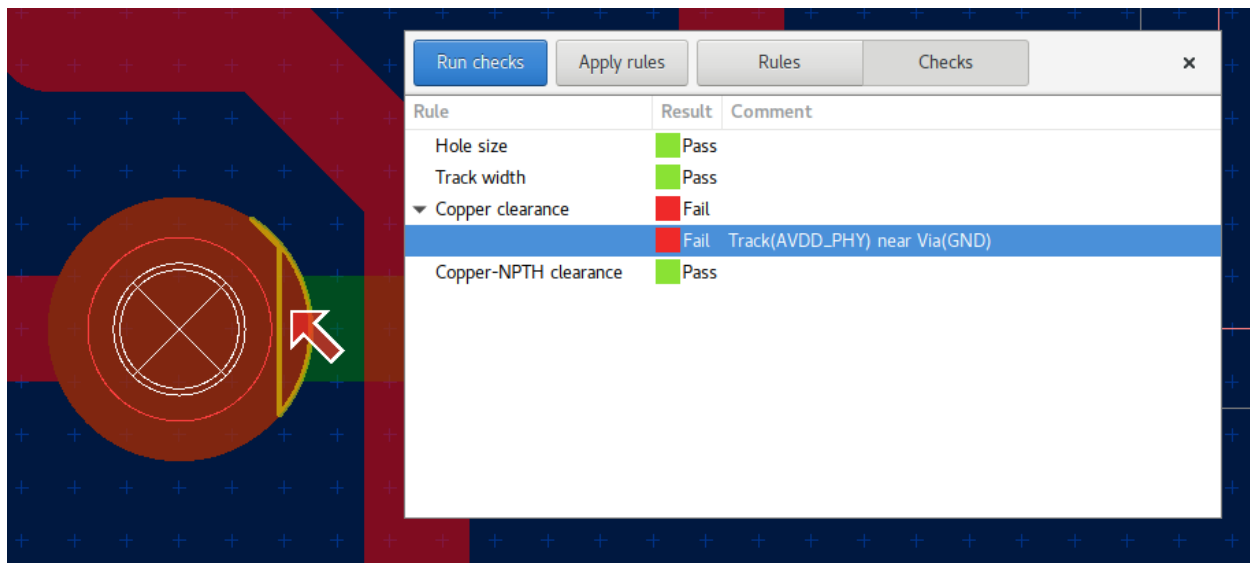
Pomocí interaktivního traséru (routeru) původně vyvinutého pro KiCad, se vytváření spojů stane hračkou. Samozřejmě respektuje vaše pravidla návrhu.



## 1.5 Silná pravidla

S výkonnými a flexibilními pravidly může horizont kontrolovat a upravovat váš návrh splňující nastavená kritéria:

Pokud něco nesplňuje vaše požadavky, horizont přesně řekne, co to je a na kterém místě:



## 1.6 Průmyslové standardy výroby

Až bude váš návrh připraven k výrobě, jednoduše exportujte ve formátu průmyslového standardu Gerber a NC-vrtání ve formátu RS-274X:

Generate

Fabrication outputs

Gerber Layers

☒ Outline
 Filename

☒ Top Paste
 Filename

☒ Top Silkscreen
 Filename

☒ Top Mask
 Filename

☒ Top Copper
 Filename

☒ Bottom Copper
 Filename

☒ Bottom Mask
 Filename

☒ Bottom Silkscreen
 Filename

☒ Bottom Paste
 Filename

NC-Drill

Mode

Drill filename

Gerber Settings

Outline width

Output Settings

Prefix

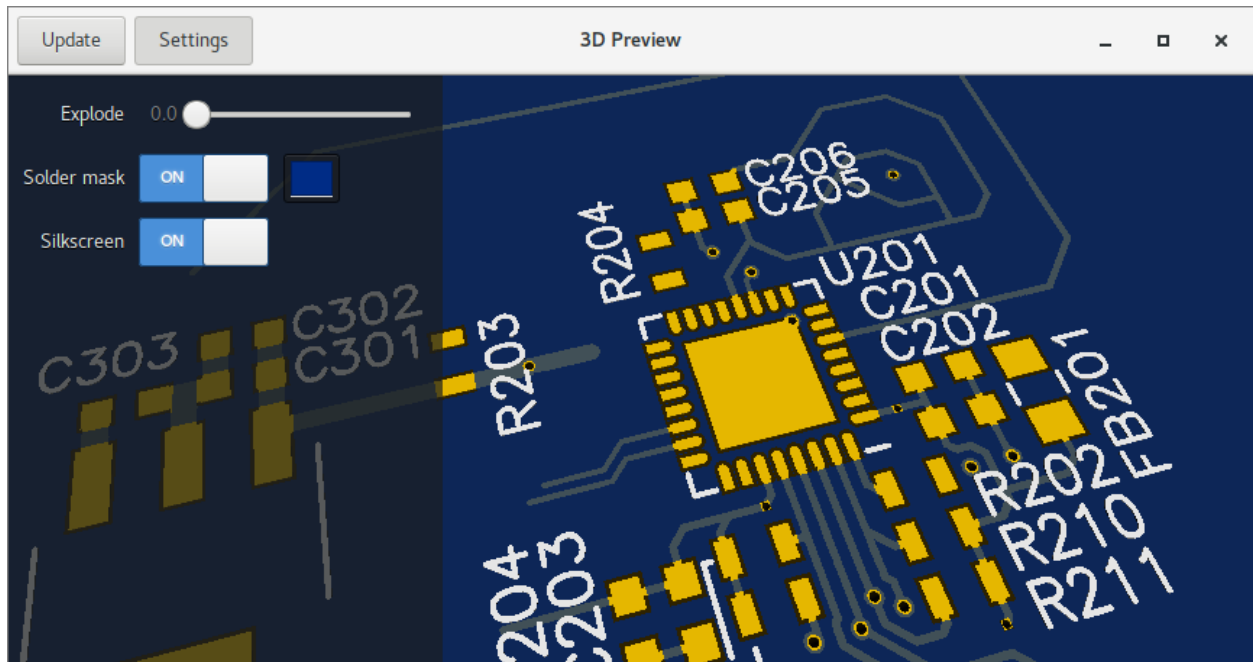
Directory

Gerber and NC-drill files will get prefixed and placed in the directory specified above.

Wrote layer Bottom Paste to gerber file /tmp/gerber/apollo-main.gbp  
 Wrote layer Bottom Silkscreen to gerber file /tmp/gerber/apollo-main.gbo  
 Wrote layer Bottom Mask to gerber file /tmp/gerber/apollo-main.gbs  
 Wrote layer Bottom Copper to gerber file /tmp/gerber/apollo-main.gbl  
 Wrote layer Top Copper to gerber file /tmp/gerber/apollo-main.gtl  
 Wrote layer Top Mask to gerber file /tmp/gerber/apollo-main.qts

## 1.7 3D pohled

Podívejte se na svou desku, jako byste ji drželi ve svých rukou. Za použití funkce rozpadu (Explode), můžete vrstvy zobrazit oddělené.



## 1.8 Je toho mnohem víc

- Akcelerované zobrazování (OpenGL)
- Neomezená možnost příkazů zpět / opakování
- Kopírování / vkládání, a to i mezi instancemi
- Vyplněné plochy spojů
- Spoje diferenciálního páru
- Libovolné tvary pájecích míst





---

## Interaktivní manipulátor

---

S cílem poskytnout jednotné uživatelské prostředí a umožnit opakované použití kódu, editory pro symboly, schémata, pájecí plošky, pouzdra součástek a desky plošných spojů jsou založeny na jednotném interaktivním manipulátoru (IMP - Interactive Manipulator).

### 2.1 Výběr objektů

Zpočátku je aktivní režim letmého výběru „hover select“. Jednoduše vybere nejmenší objekt pod kurzorem. Klepnutím nebo přetažením vyberte objekty natrvalo. Stisknutím klávesy „Esc“ se vrátíte do režimu výběru.

### 2.2 Nástroje (Tools)

Chcete-li objekty na obrazovce upravit, použijte k tomu určené nástroje. Spouštění nástrojů je možné vícero způsoby:

- Zadáním sekvence kláves daného nástroje. Dostupné klíčové sekvence se zobrazí kliknutím na tlačítko nápovědy v pravém horním rohu nebo zadáním klávesy „?“.
- Pomocí vyskakovacího okna nástrojů, které se zobrazí po stisknutí mezerníku. Stačí začít psát nástroj, který hledáte nebo procházet seznam.
- Kliknutím pravého tlačítka myši na objekt. Zobrazí se kontextové menu všechny nástroje související s tím, co je vybráno.
- Jedno z tlačítek v horní liště.

Po spuštění nástroj obdrží plné řízení dle vstupu z klávesnice a myši. Podívejte se dolů na lištu, která se objevila ve spodní části obrazovky, a uvidíte co právě provádí tento konkrétní nástroj po stisknutí kláves.

### 2.3 Mřížka (Grid)

Podržíte-li klávesu „Alt“, sníží se rozteč mřížky desetkrát.



---

## Editor desky plošných spojů DPS (PCB)

---

Chcete-li spustit editor desky plošných spojů (dále jen desky), klikněte na „Board“ ve správci projektu. Stejně jako všechny ostatní grafické editory je i editor desky v Horizon-EDA založen na *Interaktivním manipulátoru*

Editor desky si udržuje interní kopii seznamu spojů (netlist). Chcete-li aktualizovat seznam spojů, stiskněte „Save“ (Uložit) v editoru schémat pro zápis seznamu spojů na disk, pak klikněte v editoru desky na „reload netlist“ pro aktualizaci seznamu spojů nebo znovu otevřete editor desky.

### 3.1 Vyplněné oblasti (Planes)

Pro přidání vyplněných oblastí nejprve nakreslete do hladiny spojů (copper) mnohoúhelník (polygon) požadovaného tvaru. Poté pomocí nástroje „Add plane“ přiřadíte oblast. Pro zobrazení vyplněných oblastí je nutné aktualizovat tyto oblasti pomocí příkazu „Update all Planes“. Oblasti s nižší prioritou se vyplní jako první.

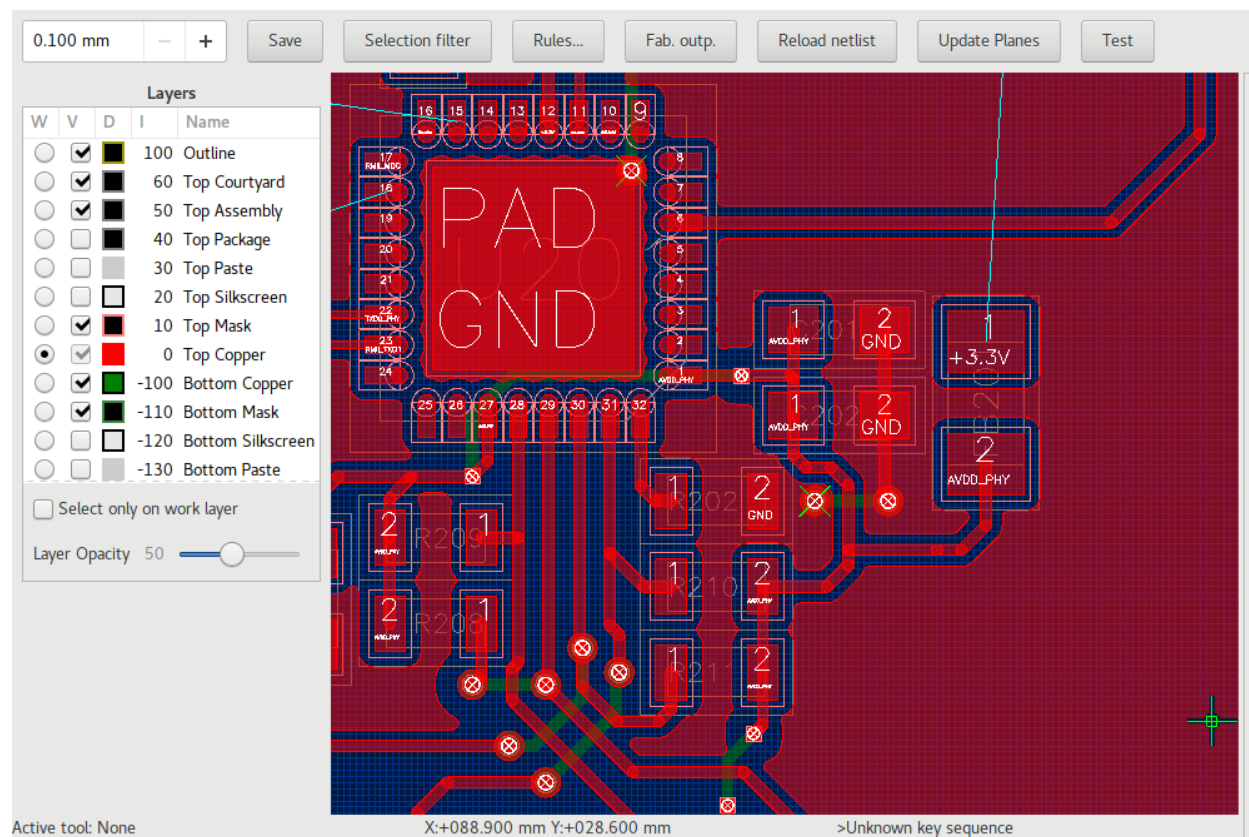
### 3.2 Prokovené otvory (Vias)

Před vložením prokoveného otvoru vytvořte výchozí pravidlo pro tyto otvory v okně pravidel „Rules“ a přiřadíte jim požadovaný tvar (padstack).

### 3.3 Diferenciální páry

Jedná se o speciální útvar trasy signálového páru se stanovenými parametry vzájemných elektrických vlastností. Chcete-li vytvořit diferenciální pár, podívejte se na *Diferenciální páry*. Před trasováním diferenciálního páru, vytvořte pravidlo určující šířku stopy a mezeru tohoto páru. Chcete-li vést trasu, použijte nástroj „Route diff“.

## 3.4 Snímek obrazovky



Chcete-li spustit editor schémat, klikněte v manažeru projektu na „Top Schematic“. Podobně jako jiné grafické editory, schématický editor v Horizon-EDA je založen na *Interaktivním manipulátoru*. Chcete-li umístit součástky, použijte příkaz „Place part“ nebo prohlížeč součástí v hlavním projektu. Ten je mnohem pohodlnější, protože může být trvale otevřený. Začněte kreslit propojení (vodiče) schematu stisknutím Klávesy „n“.


## 4.1 Spoje a segmenty spojů (Nets)

Na rozdíl od některých jiných schématických editorů Horizon-EDA si eviduje jednotlivá propojení a nejde jen o nakreslenou čáru (vodič), která se nakonec při generování seznamu spojů (netlistu) transformuje na vodič (spoj). Propojení může být reprezentováno jedním nebo více segmenty vodičů. Segment spoje je sada propojení vodičů, uzlů, vývodů atd. všechny propojené vodiči (čárami spoje). Editor sleduje, které vodiče patří do kterého segmentu a oznamuje, když se operace chystá sloučit dva segmenty.

Když se podíváte na spoj v editoru vlastností po výběru spoje, spoj v editoru vlastností je „celý“ spoj, nejen segment spoje. Proto přejmenování spoje nemění propojení. Pro připojení vývodu na segment jiného spoje, použijte nástroj pro přesunutí segmentu do jiného nebo nový spoj „Move net segment to other/new net“.

Štítek spoje (Net label) pouze zobrazuje název spoje, ke kterému je připojen, ale nemění ho. Upozorňuje vás na nekonzistenci v systému schématu, které by mohlo mít za následek nežádoucí propojení, editor schématu umísť varování na porušené položky.

## 4.2 Symboly napájení (Power Nets)

Nejjednodušší způsob vytvoření symbolu napájení ve schematu je použít příkaz pro správu napájení „Manage Power Nets“. Příkaz je k dispozici v menu hamburger . Poté použijte příkaz „place power symbol“, pro umístění symbolu napájení pro tento spoj. Symboly napájení „napájí“ jejich vodiče a připojené segmenty. Můžete si vybrat ze tří stylů napájecích symbolů ve výše uvedeném příkazu. Symboly antény a tečky lze umístit buď nahoru nebo dolů. Symbol GND může směřovat pouze dolů.

## 4.3 Sběrnice (Buses)

Pro seskupení souvisejících vodičů použijte sběrnice (Buses). Po vytvoření sběrnice přidejte vodiče. Můžete buď přiřadit existující vodiče nebo nově pojmenovat automaticky vytvořený vodič kliknutím na tlačítko se šipkou vedle ní.

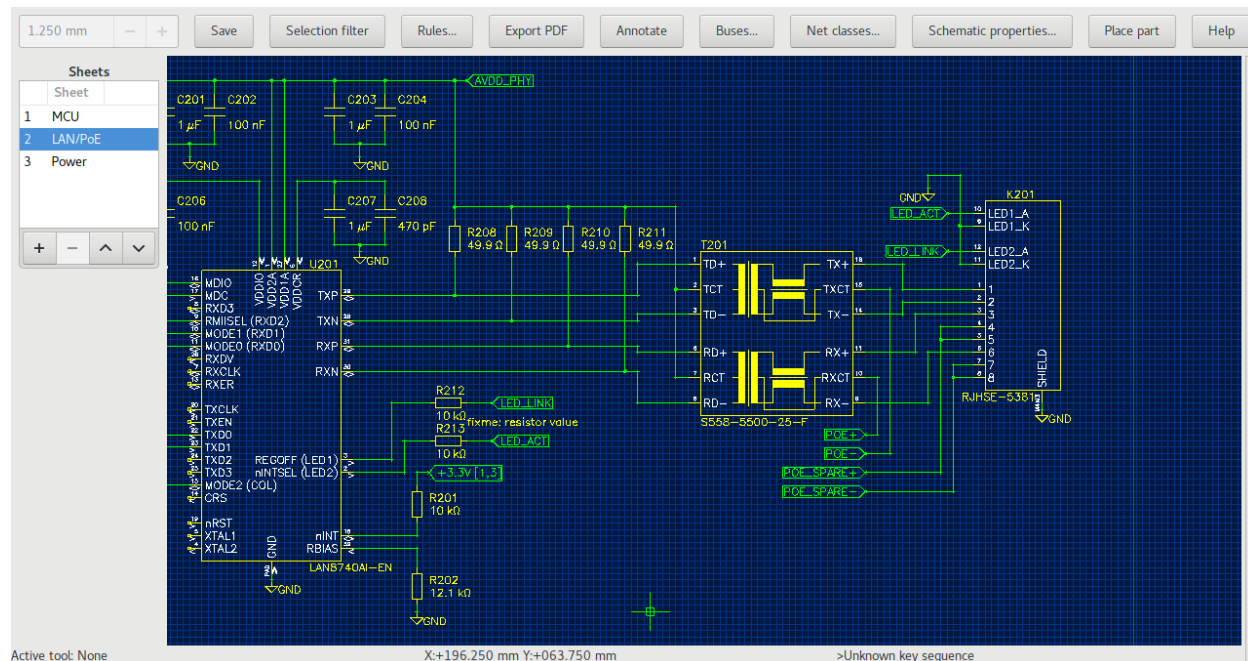
## 4.4 Diferenciální páry (Diff. pairs)

Chcete-li vytvořit diferenciální pár, vyberte dva vodiče, které chcete spárovat a spusťte příkaz „Set diff. pair“. Můžete také vybrat jeden vodič a budete požádáni o druhý. Chcete-li oddělit vodiče, použijte příkaz „Clear diff. pair“. Doporučuje se přiřadit oběma vodičům stejnou hladinu (Netclass) např. „100diff“, abyste jim mohli snadno nastavit stejná pravidla.

## 4.5 Přímo na desku (To board)

Chcete-li usnadnit umístění pouzder součástek na desku plošných spojů DPS (PCB), jednoduše vyberte požadované symboly a stiskněte „To board“. Tím se přepnete na editor desky dále spusťte nástroj „place package“ s pouzdry pro vybrané symboly. Možná bude nutné nejdříve znovu načíst seznam spojů (netlist) v Editoru desky, aby editor desky načtl nové komponenty.

## 4.6 Snímek obrazovky



---

## Sestavení (kompilace) programu na operačním systému Windows

---

### 5.1 Nainstalujte program MSYS2

Stáhněte si a spusťte instalátor programu msys2 z <http://msys2.github.io/> Mám testováno pouze s 64bitovou verzí, 32bit by měla fungovat také. Ujistěte se, že cesta, kterou jste vybrali pro instalaci neobsahuje žádné mezery (v názvech složek).

### 5.2 Spusťte konzolu MSYS

Spusťte položku nabídky Start „MSYS2 mingw 64 bit“, která by se měla zobrazit v okně konzoly. Všechny níže uvedené kroky se týkají toho, co by jste měli napsat do toho okna.

### 5.3 Nainstalujte aktualizace

Napište

```
pacman -Syu
```

**pokud vám řekne, že chce restartovat před restartováním, zavřete okno konzoly a po restartu znovu spusťte program** `pacman -Syu`.

### 5.4 Nainstalujte související součásti

Napište / vložte

```
pacman -S mingw-w64-x86_64-gtkmm3 git base-devel \
mingw-w64-x86_64-yaml-cpp mingw-w64-x86_64-boost \
mingw-w64-x86_64-sqlite3 mingw-w64-x86_64-toolchain \
mingw-w64-x86_64-zeromq mingw-w64-x86_64-glm zip \
mingw-w64-x86_64-libgit2 mingw-w64-x86_64-occe \
mingw-w64-x86_64-podofo --needed
```

Až budete vyzváni, stačí stisknout klávesu Enter. Posad'te se a počkejte, až instalátor dokončí instalaci téměř kompletního linuxového vývojového prostředí.

Než budete pokračovat, můžete přejít do jiné složky. Je to jednoduché zadejte “ cd“, mezeru a přetáhněte složku, do které chcete přejít do okna konzoly.

## 5.5 Klonovat zdrojový kód programu Horizon-EDA

```
git clone http://github.com/carrotIndustries/horizon
cd horizon
```

## 5.6 Sestavení (kompilace) programu ze zdrojového kódu

```
make -j 4
```

Číslo 4 na konci můžete upravit podle počtu procesorů ve vašem systému pro rychlejší kompilaci. Očekávejte 100% zatížení procesoru (CPU) po dobu několika minut dle výkonu počítače. Z důvodu zapnuté volby vkládání ladících symbolů mají výsledné spustitelné soubory značnou velikost.

## 5.7 Spuštění programu

Po kompilaci nebudete moci dvakrát kliknout na výsledné spustitelné soubory protože požadované knihovny DLL nejsou ve složce známé systému Windows. Budete muset spustit z příkazového řádku prostředí Mingw například pomocí příkazu `./horizon-eda`. Aby fungovalo stahování fondu, musíte zkopírovat soubor `/mingw64/ssl/certs/ca-bundle.crt` do složky obsahující `horizon-eda.exe`.

## 5.8 Vytvoření archivu

Chcete-li vytvořit archiv ZIP, jak je dostupný ke stažení, spusťte příkaz `./make_bindist.sh`.



---

## Sestavení (kompilace) programu na operačním systému Linux

---

Sestavení programu Horizon-EDA v Linuxu je podobně jednoduché jako ve Windows

### 6.1 Nainstalujte závislosti

Ujistěte se, že máte nainstalované tyto související knihovny:

- Gtkmm3 >= 3.20
- cairomm-pdf
- librsvg
- util-linux
- yaml-cpp
- sqlite
- boost
- zeromq
- glm
- libgit2
- curl
- opencascade / opencascade community edition
- zeromq with C++ bindings: <https://github.com/zeromq/cppzmq>
- podofo
- libzip

pro různé distribuce jsou zde předpřipraveny příkazy pro případnou instalaci knihoven.

Pro verzi Ubuntu >= 17.04:

```
sudo apt install libyaml-cpp-dev libsqlite3-dev util-linux librsvg2-dev \  
libcairomm-1.0-dev libepoxy-dev libgtkmm-3.0-dev uuid-dev libboost-dev \  
libzmq5 libzmq3-dev libglm-dev libgit2-dev libcurl4-gnutls-dev liboce-oaf-dev \  
libpodofo-dev
```

V systému Arch Linux:

```
sudo pacman -S yaml-cpp zeromq gtkmm3 cairomm librsvg sqlite3 libgit2 curl \  
opencascade boost glm podofo libzip
```

Na Fedoře 25/26/27:

```
sudo dnf install git make gcc gcc-c++ pkg-config cppzmq-devel OCE-devel\  
gtkmm30-devel libgit2-devel libuuid-devel yaml-cpp-devel sqlite-devel librsvg2-  
devel\  
cairomm-devel glm-devel boost-devel libcurl-devel podofo-devel
```

Na openSUSE Tumbleweed:

```
sudo zypper in git make gcc gcc-c++ pkg-config cppzmq-devel oce-devel\  
gtkmm3-devel libgit2-devel libuuid-devel yaml-cpp-devel sqlite3-devel librsvg-  
devel\  
cairomm-devel glm-devel boost-devel libcurl-devel libpodofo-devel binutils-gold
```

Na FreeBSD 12:

```
sudo pkg install git gmake pkgconf e2fsprogs-libuuid sqlite3 yaml-cpp \  
gtkmm30 cppzmq libgit2 boost-libs glm opencascade podofo libzip
```

## 6.2 Klonovat zdrojový kód programu Horizon-EDA

```
git clone http://github.com/carrotIndustries/horizon  
cd horizon
```

## 6.3 Sestavení (kompilace) programu ze zdrojového kódu

```
make -j 4
```

Číslo 4 na konci můžete upravit podle počtu procesorů ve vašem systému pro rychlejší kompilaci. Očekávejte 100% zatížení procesoru (CPU) po dobu několika minut dle výkonu počítače. Z důvodu zapnuté volby vkládání ladících symbolů mají výsledné spustitelné soubory značnou velikost.

## 6.4 Spuštění programu

Výsledné binární soubory jsou samostatné a nevyžadují žádné externí datové soubory jako ikony nebo podobně. horizon-eda je hlavní spustitelný program. Spouštějte jej ze složky kde byl sestaven pomocí příkazu:

```
./horizon-eda
```

## KAPITOLA 7

---

### Pravidla

---

Horizon-EDA používá pravidla pro specifikaci vazeb pro DRC i pro vstup různých nástrojů, jako je interaktivní trasér. Pravidla jsou vyhodnocena shora dolů a použije se první pravidlo odpovídající všem kritériím. Je tedy na vás, abyste se ujistili, že pravidla jsou uspořádána od konkrétnějších na méně konkrétní. Vždy je dobré mít obecné pravidlo úplně dole.



---

## Vytvoření pouzdra součástky

---

Rozložení desky plošných spojů může být tak dobré jak dobře jsou vytvořeny pájecí obrazce (footprints) pro jednotlivá pouzdra součástek, které program používá, proto je důležité vytvořit vysoce kvalitní podklady obrazců těchto obrazců (footprints).

V Horizonu-EDA se pouzdro součástky se skládá z těchto částí:

- Pájecí místa (pads), na které se součást připájí
  - Spoje - měděné vrstvy tras jednotlivých propojení v jednotlivých (horní / dolní / vnitřní) vrstvách
  - Otvory (pro díly TH)
  - Vynechání nepájivé masky
  - Masky pájecí pasty
- Obrys pouzdra součástky (Package outline)
- Montážní obrys a referenční označení (Assembly outline)
- Textový popis součástek (Silkscreen)
- Obrys zástavby součástky (Courtyard outline)

### 8.1 Pájecí obrazce (pads)

Každý pájecí obrazec je definován několika pájecími místy (padstack) uspořádanými do různých tvarů a parametry aplikovanými na tento tvar. Podrobnosti o pájecích místech viz. *Pájecí místa / plošky (Padstacks)*. Tyto pájecí obrazce jsou pravděpodobně nejdůležitější vlastností pouzdra součástky, je vhodné s nimi začít. Pájecí místa můžete umístit ručně pomocí příkazu „Place pad“ nebo je nechejte umístit automaticky podle běžně používaných vzorů pomocí příkazu „Footprint gen.“, který je k dispozici na horním panelu. Po umístění pájecích míst mají stále svou výchozí velikost, která patrně není ta, co potřebujete. Chcete-li velikost změnit, vyberte parametry, které chcete upravit a pomocí nástroje „Edit pad“ přidejte parametry do pájecího místa. V závislosti na vybraném tvaru pájecího místa jsou přiřazeny určité parametry. Nejčastěji se používají šířka a výška pájecího místa. Použijte tlačítko zaškrtnutí vedle parametru pro parametry, které se mají použít na všechny vybrané podložky.

## 8.2 Obrys pouzdra součástky

Obrys pouzdra se používá k vizualizaci obrysu součásti. Dá se říct, že by tedy měla sledovat jmenovité rozměry součásti. Můžete použít nástroj „import DXF“ pro import výkresu DXF získaného ze STEP modelu nebo jinak. Protože účel obrysu součásti je čistě vizuální, můžete použít buď čáry, nebo mnohoúhelníky. Vývody přidejte, pouze pokud výrazně přispívají ke vzhledu součásti.

## 8.3 Montážní obrys

Montážní obrys končí na výkresu sestavy (zatím není implementováno) a jeho účelem je pomoc při montáži a kontrole PCA. Montážní obrysová vrstva tedy obsahuje pouze tyto položky: Mnohoúhelník označující obrys součásti, volitelně vývody, pokud se výrazně přispívají ke vzhledu součásti a referenční označení součásti. Na rozdíl od obrysu pouzdra je montáž pouze hrubá náhrada tvaru součásti. Musí zahrnovat nějaký druh vizuálního označení umístění prvního vývodu součásti. Použijte příkazy „Draw polygon rectangle“ a jeho možnosti dekorace pro kreslení takového obrysu. Pro referenční označení, vložte text obsahující „\$ RD“ v takové velikosti, aby se vešel do obrysu sestavy, i když je předpona o 4 číslice delší.

## 8.4 Textový popis součástek

Účelem textového popisu je objasnit umístění dílů a orientace při ruční montáži a vizuální kontrole, také by měla pomoci pokud je součást citlivá na orientaci, použijte nějakou značku prvního vývodu. Nepoužívejte tečku pro označení prvního vývodu, místo toho sklopte nebo prodlužte grafickou značku. Doporučená tloušťka čáry je 0,15 mm. Také vložte text „\$ RD“, s tloušťkou čáry 0,15 mm do hladiny popisu.

## 8.5 Obrys zástavby součástky

Obrys zástavby označuje prostor potřebný pro plochu kolem součástky, která nesmí být obsazena jinými součástkami, aby byl ponechán dostatečný prostor pro montáž. Vzhledem k tomu, že velikost obrysu zástavby je třeba upravit v závislosti na uživatelských výrobních požadavcích, musí být nastaven pomocí parametru programu. Při zvětšení obrysu zástavby 0 mm je obrys zástavby (obdélníkový) kolem pájecích míst a obrysu pouzdra součástky. Chcete-li vytvořit obdélníkový obrys zástavby, který lze parametrizovat, proveďte toto:

Pomocí příkazu „Generate courtyard“ vygenerujete počáteční obrys zástavby na 0 mm rozšíření. Pokud to nemá za následek požadovaný mnohoúhelník, použijte příkaz „Draw polygon rectangle“ pro nakreslení počátečního obrysu a nastavení jeho skupiny parametrů na „courtyard“ pomocí editoru vlastností na pravé straně okna.

Otevřete okno „Parameters“ a klikněte na „Insert courtyard program“. Je-li vše jde správně, měl by se přidat program obrysu zástavby a jeho parametr „Courtyard expansion“ nastavený na 0,25 mm.

---

### Kopírování rozmístění a umístění

---

#### 9.1 Motivace

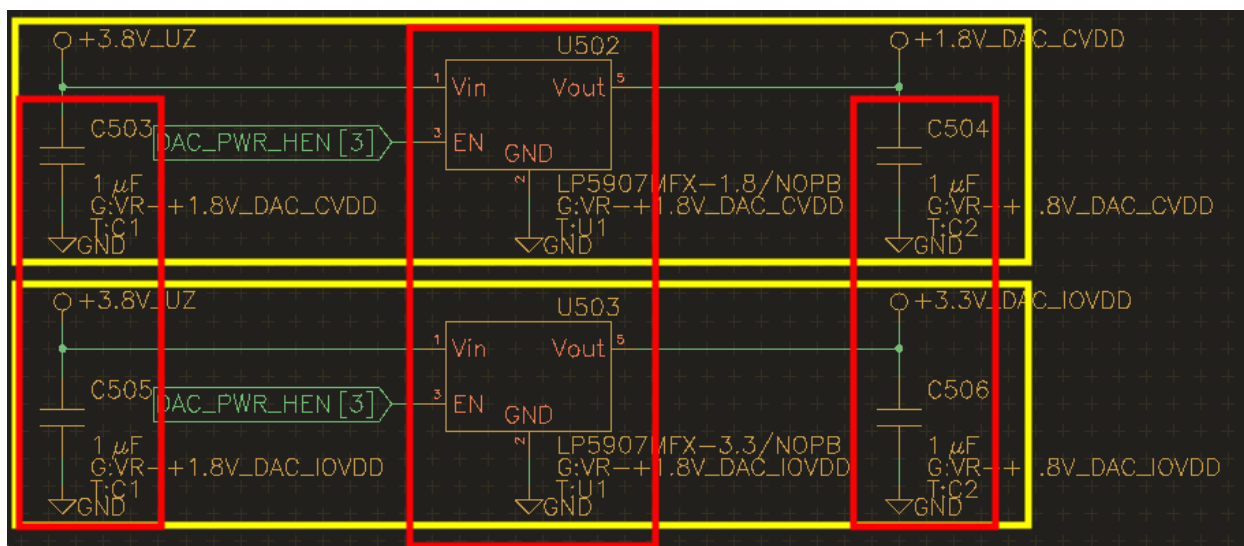
Návrh desky plošného spoje často obsahuje podobné, ale ne shodné celky jako jsou např. regulátory napětí. Nebylo by hezké, kdyby stačilo udělat rozmístění jednou a potom zkopírovat do ostatních kopií? Technologie Horzion-EDA vám to umožní v jednoduchém dvoustupňovém procesu.

#### 9.2 Skupiny a značky

Aby tato funkce fungovala, musíte nejprve sdělit Horizonu-EDA, jak spolu komponenty souvisí. Toho je dosaženo přiřazením skupin a značek ke komponentám. Každá sekce, tj. Všem komponentám spojeným s jedním regulátorem napětí, přidělte jednu skupinu. Vyberte všechny symboly jedné sekce a pomocí příkazu „Set new group“ přiřadte všechny do nové skupiny. Pro zviditelnění skupin a značek na schematu použijte „Toggle group & tag visibility“. Skupina komponent a značka se poté zobrazí pod referenčním označením.

Abychom odlišili v Horizonu-EDA související komponenty v každé skupině, dostanou přiřazené shodné značky. Protože nově umístěná součást již bude mít shodnou značku a skupinu a značky zůstanou zachovány při kopírování / vkládání jiné kopie stejného okruhu budou pravděpodobně mít vhodné značky již nastaveny. Chcete-li změnit značku na součásti, použijte příkaz „Set tag“.

Až budete hotovi, schéma by mělo vypadat zhruba takto (obdélníky přidány pro názornost). Všechny komponenty uvnitř žlutého obdélníku patří do stejné skupiny, vše uvnitř červeného pole patří ke stejné značce.



Chcete-li se ujistit, že jste udělali přiřazení správně, můžete použít akci „Highlight group/tag“.

## 9.3 Deska

Umístěte a trasujte nějakou skupinu jako obvykle.

### 9.3.1 Kopírovat umístění

Pro každou skupinu umístěte pouzdro součástky tak, aby ostatní pouzdra součástek byly odkázány na požadované místo a umístěte okolo všechny ostatní pouzdra součástek. Poté vyberte všechny součástky skupiny, které chcete znovu použít a spusťte nástroj „Copy placement“. Klikněte na referenční pouzdro (kterékoli pájecí místo nebo středový bod) v již umístěné skupině a všechny vybrané součástky budou umístěny shodným způsobem.

### 9.3.2 Kopírování skladeb

Vyberte všechny spoje (ostatní objekty budou ignorovány), které chcete kopírovat v natrasované skupině, spusťte příkaz „Copy tracks“ a klikněte na libovolné pouzdro (libovolné pájecí místo nebo středový bod) v cílové skupině.



---

## Parametry programu

---

Jak již bylo popsáno jinde v dokumentu, Horizon-EDA podporuje parametrizovatelné tvary pájecích míst a (v omezené míře) pouzder součástek. Chcete-li použít dané parametry na existující geometrii, každé pájecí místo a podobné objekty je možné změnit pomocí krátkého makra.

Tyto makra jsou psány ve vlastním zásobníkovém jazyce. Uživatelé kalkulaček HP se měli cítit jako doma. Protože v programu není možné provádět žádné smyčky, budou tyto programy ukončeny v daném čase. Zásobník obsahuje 64bitová celá čísla se znaménkem. Koncepčně roste shora dolů.

### 10.1 Syntaxe

Na nejvyšší úrovni je program tvořen značkami. Značky jsou oddělené libovolným množstvím mezer.

Typy značek:

- Celá čísla: číslo, volitelně opatřené znaménkem
- Rozměr: číslo s volitelnou zlomkovou částí, s příponou „mm“ plovoucí řádová značka před značkou mm se vynásobí  $1 \times 10^6$ , od vnitřní jednotka měření programu Horizon-EDA je 1nm
- Matematické operátory, jako například: + - \* /
- Řetězce znaků
- Parametry začínají znakem [ a končí ] jakákoliv značka mezi těmito dvěma znaky bude přidána jako parametr předchozího příkazu

### 10.2 Obecné příkazy

#### 10.2.1 Nulový operand

`get-Parameter [ <parameter> ]` načte paramter a vloží jej na zásobník

### 10.2.2 Jeden operand

Před operací vypadá zásobník takto:

```
. .  
. .  
+---+  
| a |  
+---+
```

Operátory: | Hodnoty v zásobníku  
dup | a a  
chs | -a

### 10.2.3 Dva operandy

Před operací vypadá zásobník takto:

```
. .  
. .  
+---+  
| a |  
+---+  
| b |  
+---+
```

Operátory: | Hodnoty v zásobníku  
+ | a+b  
- | a-b  
\* | a\*b  
/ | a/b  
dupc | a b a b (Duplikování souřadnic)

### 10.2.4 Tři operandy

Před operací vypadá zásobník takto:

```
. .  
. .  
+---+  
| a |  
+---+  
| b |  
+---+  
| c |  
+---+
```

Operátory: | Hodnoty v zásobníku  
+xy | a+c b+c  
-xy | a-c b-c

## 10.3 Příkazy pro pájecí místa

Aby program mohl manipulovat s objektem (tvaru atd.), musí být přiřazena třída parametrů. `## set-shape set-shape [<třída parametrů> <form>]` Nastaví tvar na zadaný formulář nebo jej přesune na určenou pozici Platného formuláře:

- obdélník, zobrazí výšku, šířku
- kruh, zobrazí průměr
- obround, zobrazí výšku, šířku
- pozice, zobrazí polohu y, x

### 10.3.1 Zadání otvoru

`set-hole [<třída parametrů> <shape>]` Nastaví díru na specifikovaný tvar z přednastavených možností:

- kulatý, zobrazí průměr
- slot, zobrazí délku, průměr

## 10.4 Polygonové příkazy (pájecí místa a pouzdra součástek)

### 10.4.1 Zadání polygonu

`set-polygon [<třída parametrů> <shape> <x0> <y0>]` Nastaví polygon před připraveného tvaru se středem na (x0, y0) z vybraných možností:

- obdélník, zobrazí výšku, šířku
- kruh, zobrazí průměr

### 10.4.2 Zadání vrcholů polygonu

`“ set-polygon-vertices [<třída parametrů> <n_vertices>] “` Načte `n_vertices` souřadnic vrcholů ze zásobníku a vytvoří z nich mnohoúhelník.

### 10.4.3 Vytvoření polygonu

`expand-polygon [<třída parametrů> <x0> <y0> <x1> <y1> ... <xn> <yn>]` Vytvoří polygon určený hodnotami souřadnic v parametrech načtených ze zásobníku.

## 10.5 Příklad programu (pro SMD obdélníkové pájecí místo)

```
get-parameter [ pad_width ]
get-parameter [ pad_height ]
dupc dupc
set-shape [ pad rectangle ]
get-parameter [ solder_mask_expansion ]
```

(continues on next page)

(pokračujte na předchozí stránce)

```
2 *  
+xy  
set-shape [ mask rectangle ]  
  
get-parameter [ paste_mask_contraction ]  
2 *  
-xy  
set-shape [ paste rectangle ]
```

---

## Použití příkazového řádku (CLI Command Line Interface)

---

Správce projektu a správce fondu do značné míry eliminovali potřebu spouštět interaktivní manipulátor a další nástroje přímo z příkazového řádku, ale je to stále užitečné pro vývoj.

Všechny níže uvedené příkazy vyžadují proměnnou prostředí `HORIZON_POOL`, nastavenou na složku fondu (obsahující soubor `pool.json` a `pool.db`)

### 11.1 Použití horizon-imp

Režim symbolů:

```
horizont-imp -y <soubor symbolu>
```

Schematický režim:

```
horizon-imp -c <soubor schematu> <soubor bloku>
```

Režim padstack:

```
horizon-imp -a <soubor pájecího obrazce>
```

Režim balíčku:

```
horizon-imp -k <soubor pouzdra součástky>
```

Režim desky:

```
horizon-imp -b <soubor desky> <soubor bloku> <přes složku>
```

## 11.2 Použití horizon-pool

Většina příkazů `-edit` a `-create` spustí příslušný `$EDITOR`, který vytvoří odpovídající soubor ve formátu YAML.

```
horizon-pool create-unit <soubor jednotky>
horizon-pool edit-unit <soubor jednotky>
horizont-pool create-symbol <soubor symbolu> <soubor jednotky>

horizon-pool create-entity <soubor entity> [<soubor jednotky> ...]
horizon-pool edit-entity <soubor entity>

horizon-pool create-package <soubor pouzdra>
horizont-pool create-padstack <soubor pájecího obrazce>

horizont-pool update #Aktualizuje SQLite databázi fondu knihovny.
```

Po vytvoření souborů nezapomeňte spustit `horizon-pool update`

## 11.3 Použití horizon-prj

Pomocí těchto příkazů můžete vytvářet prázdné bloky, schémata atd.

```
horizon-prj create-block <souboru bloku>

horizon-prj create-schematic <soubor schematu> <soubor bloku>

horizon-prj create-board <soubor schematu> <soubor bloku>
```

Takže jste se rozhodli vyzkoušet Horizon-EDA? Skvěle! Zde je návod.

## 12.1 Pořízení programu Horizon-EDA

### 12.1.1 Operační systém Windows

Stáhnete nejnovější komprimovaný přeložený binární soubor z adresy [AppVeyor CI](#) a někam ho rozbalte. Pamatujte, že se jedná o 64bitové binární soubory. V případě, že překlad dosud probíhá nebo ho někdo přerušil, si můžete stáhnout předešlé verze z [the build history](#) (kliknutím na odkaz stáhnete soubor ve formátu zip)

### 12.1.2 Linux

Viz kapitola *Sestavení (kompilace) programu na operačním systému Linux* v tomto návodě, jak zkompileovat program Horizon-EDA na linuxu.

## 12.2 Získejte fond knihovny

### 12.2.1 program Git

Pokud znáte program git, stačí někam naklonovat soubory z veřejného archivu [horizon-pool](#). Je vhodné použít program git, aby se udržely vaše místní kopie aktuální a mohli případně publikovat vaše nové díly do veřejného archivu.

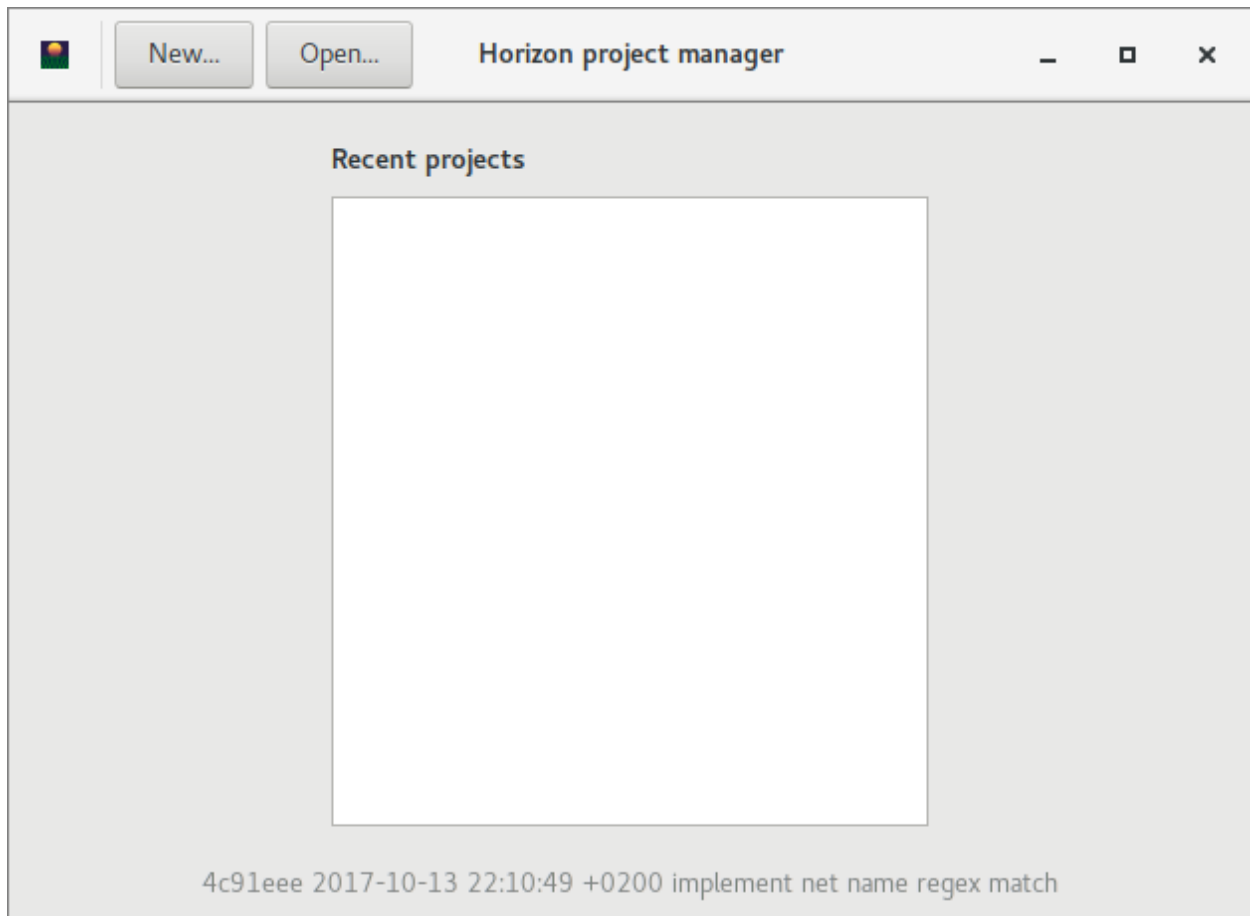
### 12.2.2 Správce fondu knihovny

Nevíte, jak na to? Žádný problém! Poklepejte na `horizon-eda.exe` nebo spusťte `./horizon-eda` z vašeho příkazového řádku a klikněte na „Download...“ pro stažení fondu. Výchozí fond `carrotIndustries/horizon-pool` je ten který potřebujete. Správce fondu vám pomůže s udržováním vašeho fondu v aktuálním stavu,

viz karta „Remote“. Také vám pomůže s vytvořením žádosti o doplnění nových součástí do veřejného archivu, a tak můžete snadno přispět do veřejného archivu bez jakýchkoli znalostí programu git.

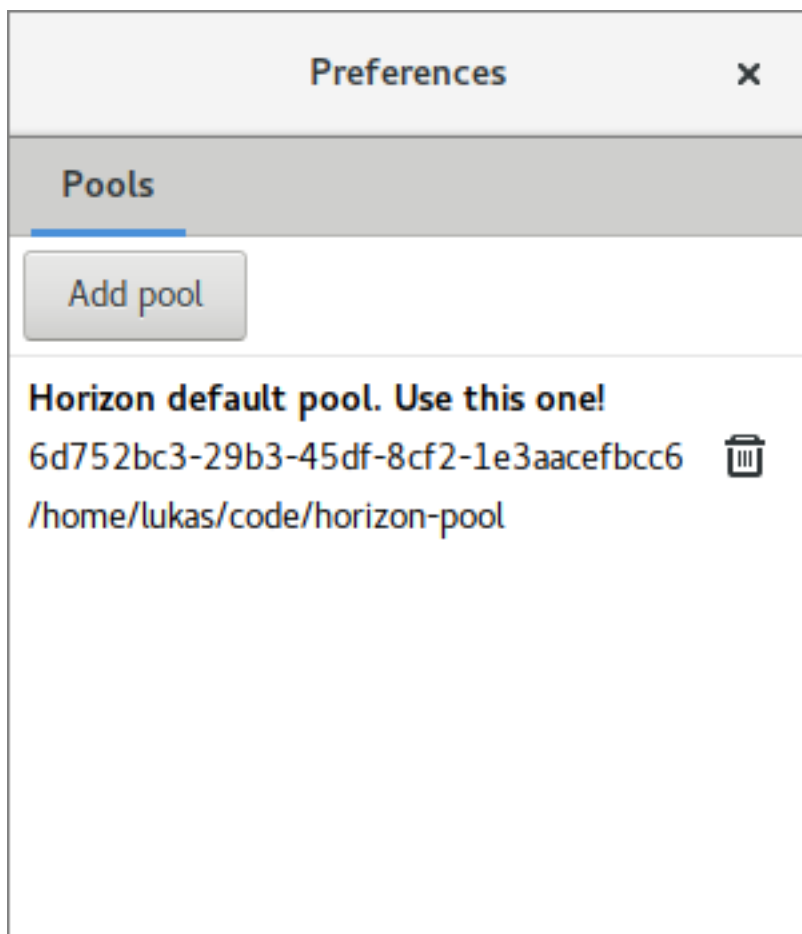
### 12.3 Vytvořte nový projekt

Klikněte na `horizon-eda.exe` nebo spusťte program `./horizon-eda` z vašeho příkazového řádku. Poté by se mělo zobrazit okno podobné tomuto (obrázek potřebuje aktualizovat):



Klikněte na ikonu aplikace v levém horním rohu a otevřete dialog nastavení. Přidejte fond, který jste právě stáhli, tím, že vyberete složku obsahující soubor `pool.json`. Když to dokončíte dialogové okno předvoleb by mělo vypadat takto:





Nyní vytvořte nový projekt klepnutím na „New...“.

## 12.4 Schéma

Po vytvoření nového projektu otevřete editor schémat a umístěte nějaké součástky a propojte je.

## 12.5 Deska

Po vložení a propojení součástek v editoru schémat klikněte na uložit do souboru schématu a otevřete editor desky v projektovém manažeru. Umístěte pouzdra součástek na plochu desky zadáním „pp“. Pro trasování propojení jednotlivých spojů, stiskněte klávesu „x“. Pro přenesení nové komponenty ze schématu na desku, klikněte na tlačítko ‚save‘ v editoru schémat a následně ‚reload netlist‘ v editoru desky plošných spojů.

## 12.6 Příklad projektu

Místo zahájení vlastního projektu si také můžete projekt stáhnout např. [návrhové soubory projektu X-Band transmitter](#). Pro otevření, vyberte složku obsahující soubor `ddstx.hprj` v projektovém manažeru. Ujistěte se, že jsou rozbaleny všechny soubory obsažené v tomto archivu.



---

## Pájecí místa / plošky (Padstacks)

---

V Horizonu-EDA pájecí obrazec (pad) součástky odkazuje na definici pájecího místa (padstack) příslušného vývodu, která určuje tvar pájecího místa (plošky). V programu existují dva druhy pájecích míst: Globální pájecí místa jsou k dispozici pro použití na všechna pouzdra součástek a měly by pokrývat většinu případů použití. V případě pouzdra součástky vyžadující speciální speciální pájecí obrazec, je možné jej vytvořit pomocí tlačítka „Create padstack for package“ na záložce pouzdra „Packages“ ve správci fondu.

Pájecí místo se skládá z těchto částí:

- Obrazec měděné vrstvy
- Obrazec pájecí masky
- Obrazec pájecí pasty
- Otvor (volitelné)

Upřednostňovaným způsobem definování geometrie pájecích míst je použití takových tvarů, které jsou vhodné pro export do gerber souboru. Pro se souhlasení více než jednoho parametru pouzdra lze použít pájecí místa, která mění svoji velikost. Lze využít také globální parametry specifické pro aplikaci, jako je rozšíření pájecí masky a zúžení masky. Chcete-li získat představu o tom, jak to doopravdy funguje podívejte se na globální pájecí místa ve fondu knihovny.

Chcete-li použít parametry na geometrii pájecího místa, je to popsáno v kapitole [Parametry programu](#), které se postarají o změnu parametrů tvarů pájecích míst.



### 14.1 Interaktivní manipulátor

Primárním rozhraním Horizontu je tzv. „Interaktivní manipulátor“. Je to jednotný editor symbolů, schémat, pájecích míst (padstacks), pouzder součástek (packages) a desky plošných spojů (PCB).

#### 14.1.1 Pracovní plocha (Canvas)

Na pracovní ploše se vykreslují objekty, jako jsou symboly schématu, pouzdra součástek nebo jednotlivé spoje. Výstupem vykreslení jsou segmenty čar a trojúhelníky, které jsou následně vykresleny pomocí grafického procesoru (GPU). Chcete-li vykreslit na pracovní ploše jiné než grafické (OpenGL) objekty, pracovní plocha poskytuje nástroje pro získání více informací o tom, co je vykresleno. Zatím program umožňuje výstup ve formátu Gerber, 3D náhled na desku a kontrolu dle pravidel návrhu (DRC - Design rule checking).

#### 14.1.2 Jádro (Core)

Protože některé dokumenty, jako jsou symboly a schémata, obsahují ty samé typy objektu (např. texty), schémata a seznam spojů (netlist) musí být modifikované synchronizovaně, musí dojít k zapouzdření. Jádro je spojovací článek mezi dokumentem, pracovní plochou a nástroji.

#### 14.1.3 Nástroje (Tools)

Pro každou akci, kterou může uživatel udělat, existuje nástroj. Vyvolaný nástroj začne přijímat vstupy z klávesnice a myši a podle funkce upravuje dokument pomocí jádra. V případě potřeby může nástroj vyvolat další dialogy pro vyžádání dodatečných informací od uživatele.

#### 14.1.4 Editor vlastností (Property editor)

Jednoduché úpravy parametrů objektů, jako je šířka čáry, neumožňují přímo jejich nástroje, proto jádro poskytuje rozhraní úprav vlastností. ovládací prvky editoru vlastností jsou automaticky generovány z popisu objektů.

---

## Fond knihovny součástek, pouzder, pájecích míst a schematických značek (Pool)

---

Co to vlastně je fond knihovny součástek (Pool)? Pro funkci programu je potřeba organizovat mnoho souborů pouzder součástek, symbolů a podobných v jakýchsi knihovnách. (Moje) zkušenost ukázala, že jsou často chaotické a správa verzí je obtížná, protože mnoho nezávislých částí definic je vloženo do jednoho souboru. Zejména toto ztěžuje spolupráci.

V Horizonu-EDA neexistují knihovny. Místo toho všechny neprojektové součásti (symboly atd.) jsou umístěny do jednoho fondu (repozitáře). Stejně jako přístup do „centrální knihovny“, která je běžnější spíše ve firemních programech pro tvorbu desek plošných spojů. I když si můžete vytvořit svůj vlastní fond, důrazně se doporučuje používat tento fond <https://github.com/carrotIndustries/horizon-pool/>. Chcete-li do něj přidat nové díly, jednoduše odešlete požadavek na přidání do veřejné knihovny pomocí příkazu „Merge“.

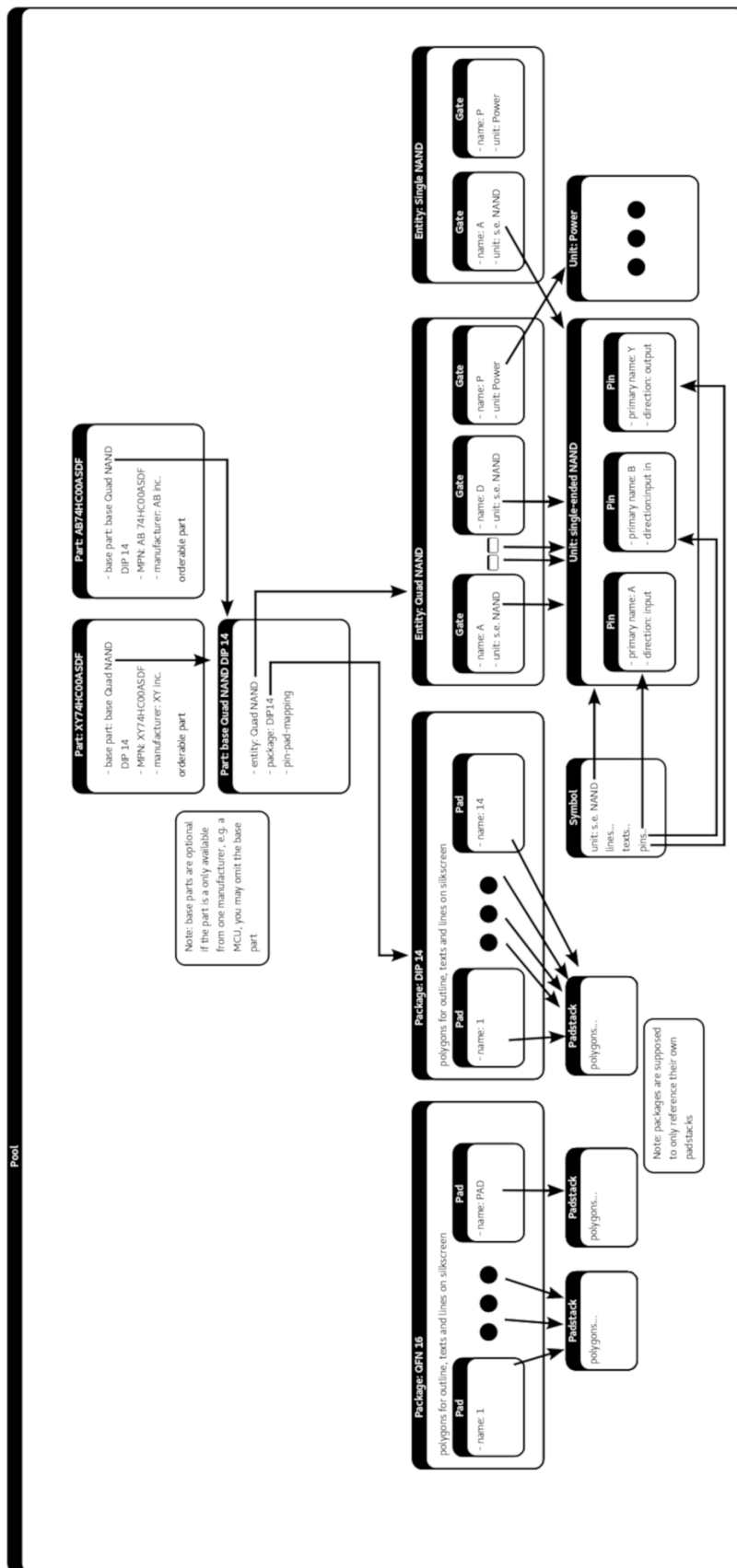
Nyní je fond organizován pomocí značek místo hierarchického systému, protože tyto často vedou k záměně nad aspekty, jako je tomu, zda skupiny dílů podle výrobce nebo jiné atributy.

Aby byl fond uspořádaný, přidávejte pouze díly, které můžete skutečně koupit s jejich odpovídajícími symboly, entity atd. Takže nepřidávejte součástku s názvem 7805, místo toho přidejte MC7805BDTRKG vyrobený společností ON Semiconductor.

Každá z níže uvedených položek je uložena v jednom souboru s příponou .json v příslušné hlavní složce, tj. /symbols, /parts, atd. Přesné umístění v této složce je nepodstatné, pokud je soubor json uložen ve správné hlavní složce. Je důležité, aby soubory měly příponu „.json“, aby byly nalezeny Správcem fondu. Chcete-li pohodlně vyhledávat díly je možné vložené vyhledávací údaje ze všech souborů json řadit pomocí databáze sqlite. To je umožněno tlačítkem ‚Update pool‘ ve Správci fondu.

### 15.1 Struktura fondu

Pro pochopení souvislostí jednotlivých souborů ve fondu je nutné pochopit jejich strukturu, která je na níže uvedeném obrázku, nebo v případě nečitelnosti je možno ji vidět [zde](#)





### 15.1.1 Součástky (Parts)

Na vrcholu struktury fondu je součástka (Part). Chcete-li se vyhnout zdvojení, součástka může zdědit svoji definici z jiné součástky. To je určeno pro použití ve skupině součástí, které se liší pouze v některých vlastnostech jako odpor nebo výstupní napětí pro pevné regulátory napětí. Každá součástka může být doplněna parametrickými daty, aby bylo vyhledávání snadnější. Tato funkce je zatím ve vývoji, protože není k dispozici žádné uživatelské rozhraní pro součástky založené na parametrických datech. Součástky také obsahují mapování Entity vývodů k pájecím obrazcům pouzder.

### 15.1.2 Pouzdra (Packages)

Pouzdro definuje průmět (otisk) půdorysného tvaru součásti na desku plošného spoje. Pokud výrobce součástky definuje doporučený tvar průmětu, použijte tento. Používejte pouze obecné pouzdra, pokud odpovídající neexistují. Podrobnosti o pouzdrech viz. *Vytvoření pouzdra součástky*.

### 15.1.3 Entity (Entities)

Entita je reprezentace součásti v seznamu spojů (netlist). Díly, které jsou logicky stejné jako např. různé typy konektorů USB proto mohou všechny sdílet stejnou entitu, např. „USB konektor se stíněním a ID“. Vlastní součástky se sestávají z jedné nebo více Jednotek (Units).

### 15.1.4 Jednotky (Units)

Jednotka ve skutečnosti definuje logické vývody součástky. Pouze pro součástky, které sestávají z jednoho celku „gate“ jako např. regulátory napětí, jejich entity jednoduše odkazují na jednu jednotku (Unit). Pro části sestávající z více celků „gate“ jako duální operační zesilovač nebo velký mikroprocesor, každý celek „gate“ odkazuje na jednu jednotku. Mít jednotky oddělené od entit umožňuje sdílení více entit stejnou jednotkou. Předpokládá se tedy, že entita jednoho logického celku „gate“ může odkazovat na stejnou jednotku např. čtyřikrát. Kromě jména má vývod i směr (pro ERC) a volitelně alternativní názvy vývodů (pinů) pro popis vývodů, které mají více funkcí, jak je běžné např. u Mikroprocesorů.

### 15.1.5 Symboly (Symbols)

Symbol se používá ve schématu k reprezentaci jednotky. Na rozdíl od jiných EDA aplikací, symbol právě zobrazuje vývody z jeho jednotky a nedefinuje je.



---

## Správce fondu (Pool manager)

---

Správce fondu a Průvodce součástmi pomáhají se správou komponent jako jsou symboly, entity a součástky ve fondu. Pravděpodobně budete používat Správce fondu pro vytváření nových dílů. Chcete-li otevřít správce fondu, spusťte `horizon-eda` („`.exe`“) a vyhledejte soubor `pool.json` fondu, který chcete upravit. Podle toho, jakou komponentu chcete vytvořit, je k dispozici několik pracovních postupů:

### 16.1 Zdědění nové součásti z existující součásti

Když součástka, kterou se chystáte vytvořit, již existuje v jiné variantě (jiná hodnota, nebo jiný teplotní rozsah), ale jinak shodná, tak by nová součástka měla být zděděna ze stávající součástky. Chcete-li to provést, vyberte požadovanou základní součástku na záložce „Parts“ a klikněte na „Create Part from Part“. Po zadání umístění souboru nové součástky se zobrazí Editor součástek. Zrušte zaškrtnutí možnosti „zdědit“ (inherit) pro atributy, které chcete změnit a uložte novou součástku.

### 16.2 Vytvoření nové součásti ze stávající entity

Tento pracovní postup je vhodný, pokud již entita pro novou součást existuje. Odporů nebo LED v nestandardních pouzdrech jsou typickým příkladem. Pro vytvoření nového pouzdra viz [Vytvoření pouzdra součástky](#). Na záložce „Parts“ klikněte na „Create Part“ pro vytvoření nové součásti. Potom zadejte entitu, pouzdro a umístění souboru součástky, kterou můžete pozměnit a mapovat pájecí místa na vývody v Editoru součásti.

### 16.3 Vytvoření zcela nové součásti

Mnoho součástek, jako jsou mikroprocesory (MCU), FPGA, ADC a další zázraky dnešního světa vyžadují vytvoření nových jednotek a entit. To by bylo manuálně velmi zdlouhavé, v tomto vám může pomoci průvodce součástmi (Part Wizard). Poté, co vyberete pouzdro součástky (pro vytváření pouzder součástek viz [Vytvoření pouzdra součástky](#)) na záložce „Packages“ klikněte na „Part Wizard...“. Budete uvítáni seznamem všech pájecích míst pouzdra.

Vyplňte názvy vývodů podle katalogového listu součástky. Vložte pouze primární jméno vývodu (např. PB5) na MCU do sloupce zcela vlevo a vložte všechny ostatní názvy (jako UART0\_TX, TA0) oddělené mezerou do sloupce „Alt. names“. Pokud je vaše součástka *opravdu* velká (jako FPGA nebo velký MCU), může potřebovat, aby se ve schématu zobrazil více než jeden symbol. Vyberte všechny vývody, které chcete mít se stejným symbolem a typem v alternativním názvu vývodu. V případě, že je více podložek elektricky identických (např. několik vývodů GND), můžete je seskupit jejich výběrem a kliknutím na tlačítko „Propojit podložky“ na dolním panelu nástrojů. Tímto způsobem bude pro vybrané položky vytvořen pouze jeden vývod.

U opravdu velkých součástek s více než 100 vývody může být ruční vkládání příliš zdlouhavé. Chcete-li se tomu vyhnout, je možné použít průvodce součástmi (Part wizard) a importovat názvy vývodů ze souboru s příponou json. Tento soubor může být generován např. nějakým skriptem v jazyce Python nebo podobném. Struktura souboru json by měla vypadat takto:

```
{
  "1": {"pin": "PB0", "alt": ["TXD", "SDA"], "gate": "Main"},
  "2": {"pin": "PB1", "alt": ["RXD", "SCL"], "gate": "Main"}
}
```

Klíč určuje název vývodu součástky. Záznamy s pin-gate přiřazují názvy k číslu vývodu.

Vhodnými zdroji údajů se jmény vývodů a pájecích míst jsou:

- IBIS modely
- BSDL soubory
- katalogové listy PDF dané součástky

Není tak složité extrahovat údaje o vývodech a pájecích místech z jednoho konkrétního katalogového listu pomocí funkcí kopírovat/vložit text do programu LibreOffice Calc, vyčistit a přeuspořádat ho a následně exportovat jako soubor CSV. Tento soubor poté převést na výše uvedený soubor s příponou json.

Jakmile vyplníte názvy vývodů, klikněte vlevo nahoře na „Další“ pro postup na další obrazovku. Vyplňte položky podle vaší součástky. Pokud si nejste jisti, co tam má být, podívejte se na stávající součástky ve fondu. Pokud je vaše součástka k dispozici vícekrát v téměř identických variantách, které se liší pouze v aspektech jako teplotní rozsah nebo možnost balení (Páska / Role, Trubka, atd.) vytvoří součástku, kterou se chystáte použít. Pro vytvoření dalších variant postupujte podle pokynů v horní části. Postarejte se o správné zadání umístění jednotek / symbolů / entit a částí tak, aby končily v podsložkách jejich příslušné složky ve fondu.

Pro každý celek (gate) klikněte na „Edit Symbol“ pro spuštění interaktivního manipulátoru vytvoříte symbol pro tuto jednotku (unit). Použijte příkaz „Map pin“ pro umístění vývodů do symbolu a „Draw line rectangle“/„Edit line rectangle“ pro nakreslení těla symbolu. Nezapomeňte dát symbolu smysluplné jméno a umístěte texty „\$REFDES“ a „\$VALUE“.

Když nakreslíte všechny symboly a vyplníte všechny údaje, kliknutím na „Finish“ konečně vložíte součást do fondu.

## 16.4 Kam ukládat komponenty

Při vytváření nových symbolů, součástek a podobně správce fondu / průvodce tvorbou součástek vás dříve nebo později požádá o název souboru nebo složky (v případě pouzder) k uložení nové součástky. Technicky, vámi specifikovaná cesta pouze musí splňovat dva požadavky:

- **Musí být ve správné složce nejvyšší úrovně, tj. pro každou schematickou značku** musí být někde ve složce /symbols atd. Pájecí obrazec specifický pro a pouzdro musí být umístěno do složky /padstack.
- Název souboru musí mít příponu . json

Pro získání představy jak to všechno prakticky vypadá se podívejte do veřejného fondu na adrese <<https://github.com/carrotIndustries/horizon-pool/>>‘\_\_

## 16.5 Databáze fondu

Fond uchovává údaje (názvy souborů, UUID, jména atd.) V SQLite databázi usnadňující vyhledávání. Normálně si správce fondu aktualizuje databázi pokaždé, když se něco změní. Nicméně, pokud externě manipulujete / odstraňujete soubory, musíte kliknout na „Aktualizovat“ (Update Pool) databázi, která zahrne provedené změny.

## 16.6 Integrace GitHub

Aby integrace GitHub fungovala, musí být fond stažen pomocí tlačítka „Download...“ na úvodní stránce správce fondu. Správce fondu naklonuje veřejný fond do složky `.remote` v systému vašeho místního fondu. Pokud vše půjde dobře, neměli byste nikdy zasahovat do této složky. K dispozici jsou dvě operace pro uchování vaší místní kopie aktuální a zahrnutí vašich součástí do veřejného fondu

### 16.6.1 Aktualizace fondu

Tento příkaz aktualizuje vaše kopie globálního fondu ve složce `.remote` od posledního potvrzení a zeptá se vás, jaké změny chcete použít ve vašem místním fondu.

### 16.6.2 Vytvoření žádosti o zahrnutí součástí do veřejného fondu

Nejprve přidejte součástky / entity / atd. do seznamu „items to be merged“, poté vyplňte název a tělo žádosti o zahrnutí. Správce fondu bude automaticky přidávat položky, které jsou nutné, aby nebyly porušeny závislosti. Takže když vytvoříte zcela novou částku s novou jednotkou, entitou a pouzdrům přidejte ji do seznamu pro přidání součástí. Nezapomeňte přidat nové symboly. Poté, co se ujistíte, že to je to, co chcete, klikněte na „Create pull request“. Budete vyzváni k zadání přihlašovacích údajů pro server GitHub.



---

## Zpětné úpravy (Backannotation)

---

Někdy můžete potřebovat propojit vývody součástek na základě jejich umístění na desce, jako například konektory nebo vývody FPGA. Nebylo by krásné, pokud by jste mohli definovat tato spojení přímo v editoru desky bez přechodu tam a zpět mezi editory desky a schémat pro každý spoj? S Horizonem-EDA je to možné!

### 17.1 Jak na to

Pomocí nástroje „Draw connection line“ (spustit přes mezerník) propojte jednotlivá pájecí místa (plošky), které požadujete propojit. Potom použijte funkci „Backannotate connection lines“ pro zpětnou změnu v editoru schémat. Nově vytvořená připojení se zobrazí jako prázdné štítky nového přerušného spoje. Po uložení schématu a aktualizaci seznamu spojů (Reload netlist) v editoru desky, budou spoje automaticky nahrazeny naznačenými spoji, které je nutno následně ručně propojit stopou mědi. Pozn. Pro správnou činnost této funkce je nutné mít spuštěn editor desky a editor schémat současně.

### 17.2 Omezení

Protože tato funkce není přidána příliš dlouho, některé možnosti nejsou zatím podporovány:

- Nelze propojit dva existující spoje
- Řetězové spoje (propojení dvou spojů s jedním vývodem) nepřinesou očekávaný výsledek

<sup>1</sup>

---

<sup>1</sup> Přeložil / Translated by : Peta-T 16.11. 2019