# Description

The Generic High Energy Physics Parts Library (GHEPlib) is a CAD data library with support for multiple CAD tools.

# Objectives

* Multi-CAD tool support
* Data reuse where reasonable
* Quality control: revision controlled, standard compliant, production tested
* Uniform look, size, and style across CAD tools
* User friendly: good-looking, paper print readable, publication ready, documented
* Low coupling with optimization for different use cases (design density, formatting preferences, etc.)
* Wide manufacturing technology support

# Dependencies

* Supported CAD tools
  + Siemens PADS VX2.3 or higher
  + Siemens DxDesigner (assumed import ability from PADS)
  + Altium Designer 23 or newer (tested)
  + KiCAD
* Symbol library
  + None
* Management tools
  + Python 3

# Used On

* Caribou v2.0 project
* Internal (confidential) projects at contributor sites

# Primary Developers

* Brookhaven National Laboratory
* Carleton University Particle Physics Instrumentation Group

# IP Control

* See License File
* Data can be imported from manufacturers into this library only if that data is accompanied by a waiver of ownership (stated as “free to use for any purpose including sale and open source”)
* Imported data without modification is kept in separate import libraries. If data is modified or customized, it should first be copied to the custom libraries. The indicated author should also change.

# Conventions

## Pin Mapping for CAD Data Reuse / “Generic Parts”

There are multiple ways to handle generic symbols and footprints:

* part + generic symbol + pin remapped footprints
  + E.g. PART\_NUM, BJT, SOT23\_123
  + In practice this is very challenging to manage
* part + pin remapped symbol + footprint
  + E.g. PART\_NUM, BJT\_123, SOT23
* part + generic symbol + generic footprint + pin map file to associate pin numbers, pin names, or pin IDs between the symbol and footprint
  + Different CAD tools use different values to link schematic and layout.
* Part + one symbol + one footprint, 1 to 1 map
  + PART\_NUM, BJT\_123, SOT23\_123

Symbols and footprint pin mapping is dictated by the CAD tool (pin names or pin IDs are sometime used) and symbols and footprints must comply with this.

However, within this library, linkage by a positive integer is generally preferred. The reasons for this are

* This transcends 2D design limits (column+row format)
* This also applies to virtual / logical pins
* This is compact to store

The conventions for this are:

* Pin numbering provided by manufacturer should generally be preferred
* Alphanumeric pin IDs (such as for BGAs) should be converted by sorting alphanumerically and ascending
* Pins missing a number on manufacturer datasheets (E.g. power pads) should be assigned am unused number

**Selected strategy:**

For parts with 3 pins or less (transistors, potentiometers, etc.):

* generic symbols are available (pin remapping is done at the symbol level).
* 2 pins gives 2 unique symbols
* 3 pins gives 6 unique symbols and one unique footprint

123 mapping conventions:

* BJT CBE E.g. C→ 1 B→ 2 E→ 3
* MOSFET DGS
* Potentiometer top wiper bottom
* TVS diodes vcc signal gnd

For parts with more than 3 pins:

* It is considered a coincidence that the pinout is identical between two given parts. This even applies to fairly standard parts such as quad opamps, comparators, etc. since the pinouts usually still vary.
* If a high (>3) pin count part is found to have an identical pinout to an existing symbol and/or footprint, this CAD data can be directly reused without renaming any files and only changing properties as needed. This linkage should be indicated in the database.

A “drop in” replacement occurs when:

Manufacturer: May vary  
Manufacturer PN: May vary  
Symbol (includes pin numbering / connectivity): No change  
Footprint: No change  
Part specifications: May vary but typically must be close

A “near drop in” replacement occurs when:

Manufacturer: May vary  
Manufacturer PN: May vary  
Symbol (includes pin numbering / connectivity): No change  
Footprint: May vary  
Part specifications: May vary but typically must be close

## **CAD Data Reuse**

* Note that, due to JEDEC, IEC, ANSI, ISO and other mechanical standards, footprint pin numbering is unlikely to change and footprint reuse is much more likely than symbol reuse.
  + It should not be assumed that manufacturers are standards compliant: pin numbering on footprints should always be verified before using a footprint
* A symbol which is a superset of another symbol is considered a completely different symbol
  + This is clearer in the schematic and prevents design errors
  + This is to target DRC cleanliness
  + E.g. DFN and TSSOP parts have the same pin numbering but the DFN has an additional PAD connection
* A symbol with multiple sub sections (units, heterogenous parts) is considered a single symbol
  + This is partly driven by CAD tool limitations: many CAD tools (Altium, KiCAD) actually have the symbols grouped within the database.
  + While some CAD tools (Siemens) have piece-meal symbols that could be reused, the symbols must still be linked together with properties or other data entry and it is considered best if this is included in the symbol.
* Separate footprints are allowed to accommodate different assembly methods (wave, reflow, bonding, etc.)
* Separate footprints for different design densities are allowed.
  + If possible, it is recommended to not make a new footprint and modify footprints on a per instance basis to meet density needs. E.g. Silk screen can be removed.
  + Separate footprints (schematic based footprint modification) are recommended when the CAD tool does not allow modifying footprint instances or when parts are very frequently used (passives) and manual modifications would be tedious.

## Schematic Symbol Drafting

* Nominally hard metric except for pin and schematic routing grid
* Primary scale: 2.5mm = ½ of preferred, print ready, scale
  + Note that all dimensions are scaled from what are considered comfortable in terms of reading on a paper print, hand drawing, and hand writing Text without any assistive technologies (magnifying glass, digital zoom, etc.).
  + Text heights for reference:
    - 5mm ~=16pt
    - 4mm ~10pt
    - 3mm ~= 8pt
    - 2.54mm ~=6.7pt
    - 12pt and 10pt are standard publishing font sizes.
  + Preferred technical drafting paper would be 5mmx5mm course grid with a 1mm fine grid (mm paper, 2x the primary scale). This is at small handwriting scale (4mm ~=10pt), hand drawing scale (~1mm tolerance), this is a standard grid paper size, and a decade increment.
  + Note that, when drawing by hand, lines have the width of the pen or pencil tip. This is assumed to be 1mm (unscaled) and sets a minimum spacing on lines and geometry for them to be legible: roughly 1mm edge to edge = 2mm center to center = 1mm scaled center to center
    - In contrast, vector graphics on computers can scale and rerender lines automatically and allow them to have practically a 0 unit width.
    - This requires a clear distinction between edge to edge and center to center dimensions.

|  |  |  |  |
| --- | --- | --- | --- |
| Grid Name (relative to drafting or component origin) | Grid Dimensions | Use for | Note |
| primary imperial pin grid | 2.54mm x 2.54mm | * components (origin) * pins * nets * busses | By observation, a 0.1” grid is the de-facto standard for manufacturer provided data, data generated by Ultra Librarian, and the KiCAD libraries. This library is made to be compatible.  For PADS with metric units, grid must be entered manually.  For a fast library conversion to a hard metric grid, the pins lengths can be resnapped and length rounded to the nearest 1mm. |
| course metric snap grid | 2.5mm x 2.5mm | * boxes * large polygons * text |  |
| text snap grid | 1.00mm x 1.00mm | * text * properties | Convenient since font rules are specified in mm increment font heights. |
| imperial drawing snap grid | 0.508mm x 0.508mm | * pin end point * boxes * large polygons * shapes which intersect with pins * Pin name labels * Pin number labels | Must not be smaller than the fine drawing snap grid to ensure readability of pin lines. |
| fine drawing snap grid | 0.5mm x 0.5mm | * lines * polygons * arcs * circles * text * pin labels * special symbols * arrows | Smaller grids are not allowed – smaller features are not easily visible. |
| intersection snap grid | 0.1mm x 0.1mm | * arcs * line-line intersections | This is primarily needed for aesthetics. Without this fine resolution grid, gaps between lines become visible. |
| imperial intersection snap grid | 0.127mm x 0.127mm | * pin line to geometry line intersections |  |

* Origin should generally be at the lower left of the symbol
* Imperial metric conversion
  + Pins are sized on on the fine imperial grid and intersecting metric geometry is resnapped.
    - In some cases, this results in the entire symbol being on an imperial grid.
    - This is much faster to draw than native metric length pins.
  + Originally metric graphics (text, special symbols) unrelated to pins are kept on the metric grids.
* Text
  + Pin and net labels and numbers: 2mm tall
    - Goal is to nearly fill all available space between pins and nets.
  + Symbol adjacent text: 2mm tall
    - All uniform
    - Should generally be at lower left of symbol. Exceptions apply to small parts.
  + General writing text: 2mm tall
    - For printing, text should not be smaller than 4mm in height but this value matches the primary scale and users are expected to scale as needed for printing.
* Pin length
  + Minimum, no pin number label label: 0.5mm
  + Minimum, with pin number label: 2.5mm
    - Fits 3 handwritten digits giving a highest pin number of 999
  + Typical: 2.5mm or 2.54mm (for convenience)
  + Length integer multiples of 2.5mm or 2.54mm
* Lines
  + Minimum length: 1mm
  + Typical width for vector graphics: 1px / vector
  + Typical width for raster graphics or vector graphics with line weight: 0.5mm
  + Minimum parallel line spacing, center to center: 1mm = 1 fine grid point between lines
* Pin labels
  + Pin number labels are not mandatory but are recommended for ease of use and clarity. E.g. large ICs, multi-component devices, etc.
  + Pin name labels are not mandatory if the connectivity is clear.
* Triangles
  + Perfectly equilateral triangles are preferred but have a base/height ratio of 2/sqrt(3)=1.1547 so they do not fit well onto a grid
  + Closest preferred base/height ratios are
    - 4/3 = 1.333
    - 6/5 = 1.200
    - 8/7 = 1.142
  + 1/1 triangles are not recommended
* Circle and arcs
  + smallest radius, arc, filled circle, or hollow circle: 0.5mm
  + digital bubbles: 0.5mm radius
  + testpoints: 0.5mm radius
* Symbols should be as small as possible while maintaining clarity, readability, and general good appearance.
  + Usually, the size is not limited by the geometry but by the adjacent text (Ref. Des.) Almost all symbols have at least 2 lines of text and thus have a minimum height of 2\*2mm=4mm = 2 grid spaces.
  + If drawing by hand, a similar approach would be used to not waste paper area or drawing time.
  + Uniformity in scale across symbol types is not mandatory (e.g. same size triangle for all amplifiers regardless of feature set, e.g. all 2 pin components are the same size). It is recommended to size symbols as small as possible while fitting all the required information (subsymbols, text, pin labels, etc.) and while respecting all the other drafting rules (e.g. minimum and default text sizes).
  + Scaling subsymbols (pin label text, digital bubbles) with overall symbol size is not mandatory. Readability is typically maintained at the original scale.
* Symbol variants are allowed to accommodate different display formats or preferences (e.g. pin number labels or no pin number labels) but one symbol must always be designated a default.
* Symbol names should
  + be human readable (no reverse order)
  + not contain special symbols (URL safe, for compatibility with file systems and to be easy to type)
  + be written from least to most specific
  + not contain abbreviations

## Layout Footprints

* Nominally metric
  + Imperial is often what is given by the manufacturers or required by a standard. Use the following rules to convert:

|  |  |  |  |
| --- | --- | --- | --- |
| Conversion(s) | Typical Applications | Metric grid (rounding) | Imperial equivalent |
| Strict tolerance | Drill sizes  RF  Wafer scale dimensions | 1um | 39.37e-6 inches |
| Standard imperial to metric | Connectors | 25um | ~1mil=0.001 inches |
| Rounded imperial to metric | General Purpose | 100um | 3.937 mil |
| Large rounded imperial to metric | Copper Fills  Mechanical Outlines | 1mm | 39.37mil |

* + The conversion strategy appropriate for the application should be used. Tight tolerances require smaller rounding errors. Note that the courser rounding is usually an integer multiple of the finer options.
  + For drill size matching between imperial and metric sizes, drills should always use strict (1um) conversion
  + Standard passive components footprint names should include whether they are imperial or metric to avoid confusion in duplicate names
* Silkscreen
  + Polarity indicators included but may be covered after assembly
  + All footprints should include a bounding box. The primary reason for this is to show which pads are grouped when a part is not installed. This also serves as a type of layout keepout.
* Soldermask
  + Dedicated shapes
* Solder paste
  + Dedicated shapes
* Assembly layers
  + Dedicated shapes
* Drill layer
  + Preferred mechanical drill size: 1.1mm
  + Preferred microvia /laser drill size: 250um
* Metal layers (external and internal, general rules for wide technology support)
  + Min metal width, any geometry to any geometry, any net to any net, any plating thickness, any layer: 400um
  + Min metal to metal spacing: same as min metal width

# Device Property Conventions

* Multiplier Format
  + “DNI” = “DNP” = “NP” = 0
  + multiplier (integer) = 1xmultiplier
  + “” (empty) = 1
* Unique Resistance, Capacitance, Inductance, … properties
  + Siemens tools typically prefer a single VALUE property that is displayed. This is also more convenient for BOM export. However, this lacks clarity since the exact property being referred to may need to be assumed (e.g. diode forward vs. reverse voltages)
* Siemens
  + Convention for DEVICE = “Symbol Name” without any suffixed that are used to identify a part of a heterogenous symbol
    - Formerly, DEVICE = “Symbol Name”\_PN. As of 2023-03-08, the MANUFACTURER\_PART\_NUMBER property was added to enable the use of different symbols for the same part within the same design. This property name is also more explicit, clearer, and CAD tool independent
    - DEVICE must be unique since the tool uses this property for layout-schematic linkage
  + MANUFACTURER\_PART\_NUMBER= “PART\_NUMBER” before completion
  + Property names target the Netlist PADS/DxDesigner flow. The PADS documentation indicates that the cases for the properties vary between the Netlist and Intgrated design flows. E.g. [Part Number (PART\_NUMBER)](../../MentorGraphics/PADSVX.2.10/docs/htmldocs/attr/topics/General_PartNumberPartNumber_idee3e0fd4.html" \l "idee3e0fd4-f115-4b2d-a24a-fb65ff02efa8__General_PartNumberPartNumber_idee3e0fd4.xml%23idee3e0fd4-f115-4b2d-a24a-fb65ff02efa8)
  + HETERO property is fully completed if applicable. Symbol grouping is not explicitly stored in the database
  + PKG\_TYPE=”PKG\_TYPE” (unlinked – determined at PCB layout start)
    - PCB only parts do not have this property
  + Only properties necessary for display in the symbol and basic packaging are added to the symbol
    - This is faster to enter and allows direct addition of other parameters at the schematic level as preferred by designers, assuming that mapping to database fields are managed correctly
    - This is typically, DEVICE, PKG\_TYPE, PART\_NUMBER, REFDES
    - Note: For some reason, properties can only be moved if they don’t contain the default value

# Verification Flows

* PADS example project holds all newly developed symbols
  + Visual review
  + Tools→ PCB Interface… (Packaging)
  + Tools → Diagnostics
    - Shows if schematic library is out of date
  + PADS Databook → New hierarchical Verification Window
    - Shows if library items can’t be found
* Tested on the following CAD tools
  + Altium 23
  + PADS VX.2.10

# Debugging Tips

## Siemens

* Symbols can “break” and exhibit strange behavior when instantiated. This may occur if the text of the symbols is directly edited.
  + One observed symptom of this is that the tool may automatically (per an ambiguous / unknown rule set) change the case of characters in symbol names in schematics however, references to symbol filenames with different case characters still work
  + Another observed symptom of this was the inability to delete or add specific properties, especially ones that are not present in the properties list.
  + The solution to this is to copy and paste the symbol graphics to a new symbol – PADS will generate a new file and therefore clean errors
* All symbols and symbol instances in a HETERO part must have exactly the same properties, otherwise packaging fails
  + The visibility of the property can vary between symbols
* After Tools→ Update Libraries, pins may become corrupted and fail to package. To resolve this, the part must be full replaced: right click→ Replace Symbol→ Search for same symbol name→ Replace
* PADS layout libraries are **not** compatible with DxDesigner
* Tools → Property Definition Editor
  + Anytime modifications are made, the project has to be closed and reopened for the changes to be visible
  + This includes the symbol editor – it must be closed and reopened along with the project
  + Not all properties can be modified from this dialog
  + The properties available depend on
    - the selected design flow
    - the object type (border, composite, etc.)
    - the tool editing the properties (E.g. symbol editor vs. regular schematic editor)
  + A workaround to get properties available anywhere is to manually add them to the properties file.
  + An example of this is the @DATETIME property commonly used for borders. While it is an official property for a netlist flow project, for an unknown reason, it was not available in the symbol editor until it was manually added as a user defined property.
    - Note that is is unclear if a custom property was necessary or PROPERTY\_NAME=”@DATETIME” would also work
    - @DATETIME function worked correctly with @DATETIME=”” and @DATETIME=”@DATETIME”. @DATETIME=”@DATETIME” is more practical for seeing and placing the property.

# Table Descriptions

Primary keys in each table are indicated with “PK”

* CAD\_table
  + Relates parts to the **preferred** CAD files. A single part can have many CAD models or data files but only a few are preferred and/or tested.
  + MFG = Manufacturer
  + PN = Part Number. This is the primary manufacturer part number (not distributor, not internal)
  + All symbols, footprints, and simulation models are references to CAD files in the Data\_table
* Data\_table
  + path: This can be a path to a file or a section of a file.
    - If this refers to sections of a file, identifiers relevant to each CAD tool should be used.
    - Symbols recognized by the CAD tool can be used. This is useful for default libraries.
  + author: The most recent editor of the data. As soon as the data or a file is modified, the author is considered to have changed. This is for liability tracing
  + release\_version:
    - If a default library is used, the exact release version should be indicated.
    - If a file was directly downloaded from a manufacturer, the date of the download should be indicated.
  + sym\_group:
    - Some parts are best represented by multiple separate symbols. This indicates that they should be grouped together.
  + deployment\_history: A short description of the harshest environmental deployment that the design within the file has survived. Examples include “not deployed”, “functional”, “functional after shock and vibration testing…”, “functional after high temperature testing…”

# TODO

* Table generators
  + database to Altium database with dblink file
  + database to DxDatabook database
* Cleanup functions
  + CAD entry deduplicator with reference updates
* Import functions
  + Digikey BOM
  + Mouser BOM
  + General BOM