

Eisler 0.1 言語仕様

この文章はDraft版です。予告なく変更される場合があります。

Eislerとは

Eisler(アイスラー)は、テキストベースの回路記述言語です。Eislerで回路を記述し、それを処理系に通すとネットリストと部品表が出力される、そんな使い方を想定しています。

サンプル

コネクタに電源を接続するとLEDが光る、というシンプルな回路を例に、Eislerの基本を説明します。

```
ledBoard.eis
```

```
import passive;
import discrete;

defpart CONN2(1:1,2:2)
{
    param type;
}

defmodule main()
{
    part cn as CONN2("B2B-EH");
    part led as LED("OSDR3133A");
    part r as R("100K", "1/4W", "5%");

    cn.1 - 1.r.2 - A.led.K - cn.2;
}
```

import

importはC言語の#includeに似ています。`import passive;`によって、はpassive.eisの中で定義されている部品をledBoard.eisの中で使用できるようになります。

passive.eisの中には抵抗器、コンデンサといった受動素子が、discrete.eisの中には、バイポーラトランジスタやFET、ダイオードなどのディスクリート素子があらかじめ定義されています。

defpart

defpartによって新しい部品を定義することができます。

```
defpart 部品型番(ピン番号1:ピン名1, ピン番号2:ピン名2 ... )
{
    param パラメータ名1
    param パラメータ名2
    ...
}
```

部品型番には、例えば「74HC14」や「NE555」などの部品の型番を記述します。

ピン番号は数字でなければならない、また全てのピンが異なる値を持つ必要があります。ですが、値が不連続であっても構わず、またソートされている必要ありません。例えば `(3:VCC, 5:GND, 1:IN, 4:OUT)` としてもかまいません。

ピン名はVCCやGNDなど、そのピンの機能を表します。全てのピンが異なるピン名を持つ必要があります。

パラメータ名は部品の型番(type)や、抵抗値(registance)などを自由に設定できます。

defmodule

defmoduleによって、新しいモジュールを定義することができます。partは単一の部品ですが、moduleは複数(0を含む)の部品から構成されています。

```
defmodule モジュール名(外部配線名1, 外部配線名2...)
{
  part 部品名 as 部品型番(パラメータ1, パラメータ2...);
  ...

  結線を記述
}
```

結線は主に結線演算子"-"を用いて記述します。最も基本的な結線の仕方は以下の通りです。

```
部品名.ピン名 - ピン名.部品名;
```

部品のピンと外部配線を結線することもできます。

```
外部配線名 - ピン名.部品名;
部品名.ピン名 - 外部配線名;
```

結線は任意の数だけ、つなげて書くことができます。

```
部品名.ピン名 - ピン名.部品名.ピン名 - ピン名.部品名.ピン名 - ...;
```

例

```
// VCCとLEDのA(アノード)ピンを接続する
VCC - A.Led;

// 抵抗とLEDをVCCとGNDに接続する
VCC - 1.RLed.2 - A.Led.K - GND;

// VCCの電圧をRHiとRLoで分圧して、ICのSENピンに接続
VCC - 1.RHi.2 - SEN.IC.SEN - 1.RLo.2 - GND;
```

拡張子

Eislerの拡張子は*.eisです。

字句要素

この章ではEBNFによって文法を表記しています。その中でletterとdigitが共通して使用されます。

```
letter = "A" ... "Z" | "a" ... "z" .
digit = "0" ... "9" .
```

コメント

EislerのコメントはC言語のコメントと同じです。

コメントには行コメントと範囲コメントがあります。1. 行コメントは、`//` ではじまり行末で終わります。1. 範囲コメントは `/*` ではじまり `*/` で終わります。複数行にまたがってコメントアウトすることができます。

識別子

識別子は1文字以上の字もしくは数字から構成されます。

```
identifier = { letter | digit } .
```

例

```
a
LED
123
```

多くのプログラミング言語では123は整数リテラルとして扱われますが、Eislerでは識別子として扱われます。

キーワード

以下のキーワードは言語によって予約されているため、識別子として使用できません。

```
defpart defmodule part module param ref import
```

演算子

以下の文字および文字シーケンスは演算子です。

```
- | ( ) { } : ; ,
```

整数リテラル

整数リテラルは"z"ではじまり、その後につづく数字によって構成されます。

```
int_lit = "z" { digit } .
```

例

```
z0
z123
z003
```

浮動小数点リテラル

浮動小数点リテラルは"f"ではじまり、その後につづく数字と"."によって構成されます。

```
float_lit = "f" { digit } "." digit { digit } .
```

例

```
f0.0
f12.34
f.123
f123.450
```