

Autor Tema: KiPart, asistente generador de simbolos KiCAD (Leído 3679 veces)

 elektroquark.com/forokicad/index.php

Autor



KiPart, asistente generador de simbolos KiCAD

« en: 08 de Febrero de 2016, 16:39:41 »

La creación de simbolos en KiCAD puede llegar a ser tedioso, pero gracias KiPart, una herramienta hecha en Python desarrollada por un integrante de la comunidad de KiCAD ([xesscorp](#)) podemos facilitar mucho el proceso de creación de simbolos.

Mi PC donde tengo KiCAD tiene instalado Windows, por lo que no tiene instalado Python por defecto, asi que tenemos que instalarlo, una version 3.x, además debemos de tener instalado pip, [aquí](#) un video de como instalarlo de manera fácil en Windows , estos pasos deben ser adaptados si usan una distro de Linux.

Una vez teniendo Python y pip instalados podemos proceder a instalar KiPart, abrimos una terminal y tecleamos:

Código: [\[Seleccionar\]](#)

```
$pip install kipart
```

Presionamos enter y obtenemos lo siguiente:

```
>pip install kipart_
```

```
>pip install kipart
Requirement already satisfied (use --upgrade to upgrade): kipart in c:\python34\lib\site-packages
Requirement already satisfied (use --upgrade to upgrade): affine>=1.2.0 in c:\python34\lib\site-packages (from kipart)
Requirement already satisfied (use --upgrade to upgrade): future>=0.15.0 in c:\python34\lib\site-packages (from kipart)
```

Podemos comprobar que versión se instalo con el comando:

Código: [\[Seleccionar\]](#)

```
$kipart -v
```

Y podemos obtener la ayuda con el comando:

```
>kipart -v
KiPart 0.1.13
```

Código: [\[Seleccionar\]](#)

```
$kipart -h
```

```

>kipart -h
usage: kipart [-h] [-v] [-r [{generic,xilinx7,xilinx6s,xilinx6v,psoc5lp}]]
              [-s [{row,num,name}]] [-o [file.lib]] [-f] [-b] [-a] [-w]
              [-d [LEVEL]]
              file1.[csv|zip] file2.[csv|zip] ... [file1.[csv|zip]
              file2.[csv|zip] ... ...]

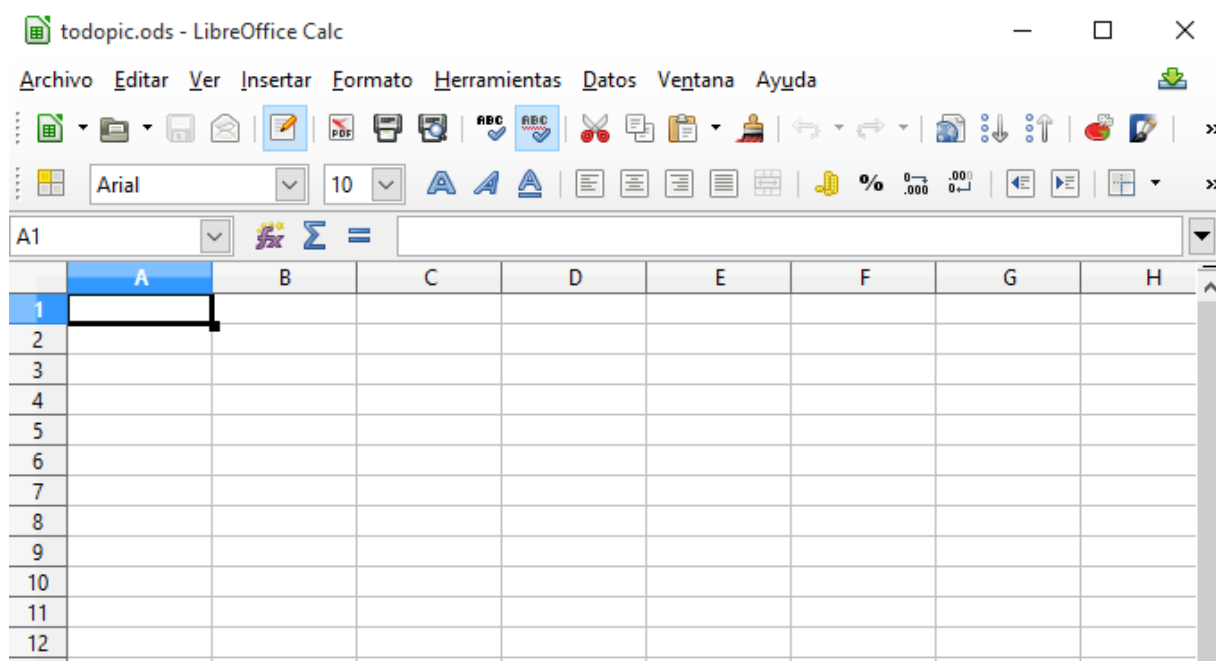
Generate single & multi-unit schematic symbols for KiCad from a CSV file.

positional arguments:
  file1.[csv|zip] file2.[csv|zip] ...
                    Files for parts in CSV format or as CSV files in .zip
                    archives.

optional arguments:
  -h, --help            show this help message and exit
  -v, --version          show program's version number and exit
  -r [{generic,xilinx7,xilinx6s,xilinx6v,psoc5lp}], --reader [{generic,xilinx7,xilinx6s,xilinx6v,psoc5lp}]
                        Name of function for reading the CSV file.
  -s [{row,num,name}], --sort [{row,num,name}]
                        Sort the part pins by their entry order in the CSV
                        file, their pin number, or their pin name.
  -o [file.lib], --output [file.lib]
                        Generated KiCad library for part.
  -f, --fuzzy_match     Use approximate string matching when looking-up the
                        pin type, style and orientation.
  -b, --bundle           Bundle multiple, identically-named power, ground and
                        no-connect pins each into a single schematic pin.
  -a, --append           Append to an existing part library.
  -w, --overwrite       Allow overwriting of an existing part library.
  -d [LEVEL], --debug [LEVEL]
                        Print debugging info. (Larger LEVEL means more info.)

```

Hasta aqui todo bien, ahora necesitamos crear un archivo .csv que tenga la información de nuestro simbolo, para esto cree un documento .ods (LibreOffice Calc), lo nombre todopic, el nombre puede ser el que deseemos:



La información con la que debemos llenar el formulario es la siguiente:

- * En A1 ponemos el nombre de nuestro componente, en este caso:
- * En A2 ponemos Pin, en esta columna va la numeración de los pines del componente.
- * En A3 ponemos Name, en esta columna van los nombre de los pines.
- * En A4 ponemos Type, en esta columna pondremos el tipo de pin.
- * En A5 ponemos Side, en esta columna pondremos el lado del simbolo donde queremos que se coloque el pin.

(El orden de las columnas no importa, las unicas columnas que son necesarias son Pin y Name, las opcionales son Type, Side, Unit y Style)

	A	B	C	D	E	F
1	18F4550					
2	Pin	Name	Type	Side		
3						
4						
5						
6						
7						
8						
9						

Y comenzamos a llenar nuestras columnas, primero la de Pin, los pines no tienen que ir en orden ascendente, ya que podemos indicarle a KiPart que ordene los pines por orden ascendente. Asi que lo llene de la siguiente forma:

	A	B	C	D
1	18F4550			
2	Pin	Name	Type	Side
3	1			
4	2			
5	3			
6	4			
7	5			
8	6			
9	7			
10	8			
11	9			
12	10			
13	11			
14	12			
15	13			
16	14			
17	15			
18	16			
19	17			
20	18			
21	19			
22	20			
23	21			
24	22			
25	33			
26	32			
27	31			
28	30			
29	29			
30	28			
31	27			
32	26			
33	25			
34	24			
35	23			
36	44			
37	43			
38	42			
39	41			
40	40			
41	39			
42	38			
43	37			
44	36			
45	35			
46	34			
47				

En la columna Name pondremos el nombre de nuestros pines, en este caso podemos copiar directamente el nombre de los pines desde el datasheet de nuestro componente y

pegarlos sin formato en el documento, algunos pines son activos en un estado lógico en bajo, para indicar dichos pines podemos incluir el simbolo "~" delante del nombre del pin, como recomendación la funcipon del pin que sea activa en bajo la ponemos al último en el nombre, ya que al crear el simbolo la raya que nos indica la lógica negada se pone apartir del lugar donde encuentra el simbolo ~ y si la ponemos al principio se nos van a marcar todas las funciones, cosa que en este caso no queremos.

Por ejemplo el pin , la función MCRL en activa en bajo, por lo tanto si la ponemos al principio con el simbolo ~ al crear la libreria se verá asi el pin:

Para evitarlo podemos poner ~MCRL al último y se vera asi:



En la columna Type ponemos el tipo del pin, para conocer que parametros podemos poner ahi nos vamos a la documentación de [KiPart](#):



The type column specifies the electrical type of the pin. The allowable values are:

input, inp, in, clk

output, outp, out

bidirectional, bidir, bi, inout, io

tristate, tri

passive, pass

unspecified, un, analog

power_in, pwr_in, power, pwr, ground, gnd

power_out, pwr_out, pwr_o

open_collector, open_coll, oc

open_emitter, open_emit, oe

no_connect, no_conn, nc

En este caso solo coloque pines de alimentación (pwr) y bidereccionales (bi). Por último la columna Side, otra vez podemos checar los parametros validos en la ayuda de KiPart:

The side column specifies the side of the symbol the pin will be placed on. The allowable values are:

left

right

top

bottom

Asi quedó mi archivo al terminar:

	A	B	C	D
1	18F4550			
2	Pin	Name	Type	Side
3		1RC7/RX/DT/SDO	bi	left
4		2RD4/SPP4	bi	left
5		3RD5/SPP5/P1B	bi	left
6		4RD6/SPP6/P1C	bi	left
7		5RD7/SPP7/P1D	bi	left
8		6VSS	pwr	left
9		7VDD	pwr	left
10		8RB0/AN12/INT0/FLT0/SDI/SDA	bi	left
11		9RB1/AN10/INT1/SCK/SCL	bi	left
12		10RB2/AN8/INT2/VMO	bi	left
13		11RB3/AN9/CCP2/VPO	bi	left
14		12NC/ICCK/ICPGC	bi	bottom
15		13NC/ICDT/ICPGD	bi	bottom
16		14RB4/AN11/KBI0/CSSPP	bi	bottom
17		15RB5/KBI1/PGM	bi	bottom
18		16RB6/KBI2/PGC	bi	bottom
19		17RB7/KBI3/PGD	bi	bottom
20		18VPP/RE3/~MCLR	bi	bottom
21		19RA0/AN0	bi	bottom
22		20RA1/AN1	bi	bottom
23		21RA2/AN2/VREF-/CVREF	bi	bottom
24		22RA3/AN3/VREF+	bi	bottom
25		33NC/ICVPP/~ICRST	bi	right
26		32RC0/T1OSO/T13CKI	bi	right
27		31OSC2/CLKO/RA6	bi	right
28		30OSC1/CLKI	bi	right
29		29VSS	pwr	right
30		28VDD	pwr	right
31		27RE2/AN7/OESPP	bi	right
32		26RE1/AN6/CK2SPP	bi	right
33		25RE0/AN5/CK1SPP	bi	right
34		24RA5/AN4/HLVDIN/C2OUT/~SS	bi	right
35		23RA4/T0CKI/C1OUT/RCV	bi	right
36		44RC6/TX/CK	bi	top
37		43RC5/D+/VP	bi	top
38		42RC4/D-/VM	bi	top
39		41RD3/SPP3	bi	top
40		40RD2/SPP2	bi	top
41		39RD1/SPP1	bi	top
42		38RD0/SPP0	bi	top
43		37VUSB	bi	top
44		36RC2/CCP1/P1A	bi	top
45		35RC1/T1OSI/CCP2/~UOE	bi	top
46		34NC/ICPORTS	bi	top

Una vez terminado el archivo, (a mi me tomo cerca de 3 mins. para generar esta lista) guardamos el archivo en formato .csv. Posteriormente nos vamos al directorio donde tenemos dicho archivo y abrimos una terminal y tecleamos lo siguiente:

Código: [Seleccionar]

```
$kipart todopic.csv -s num -f -o 18F4550_test.lib -w
```

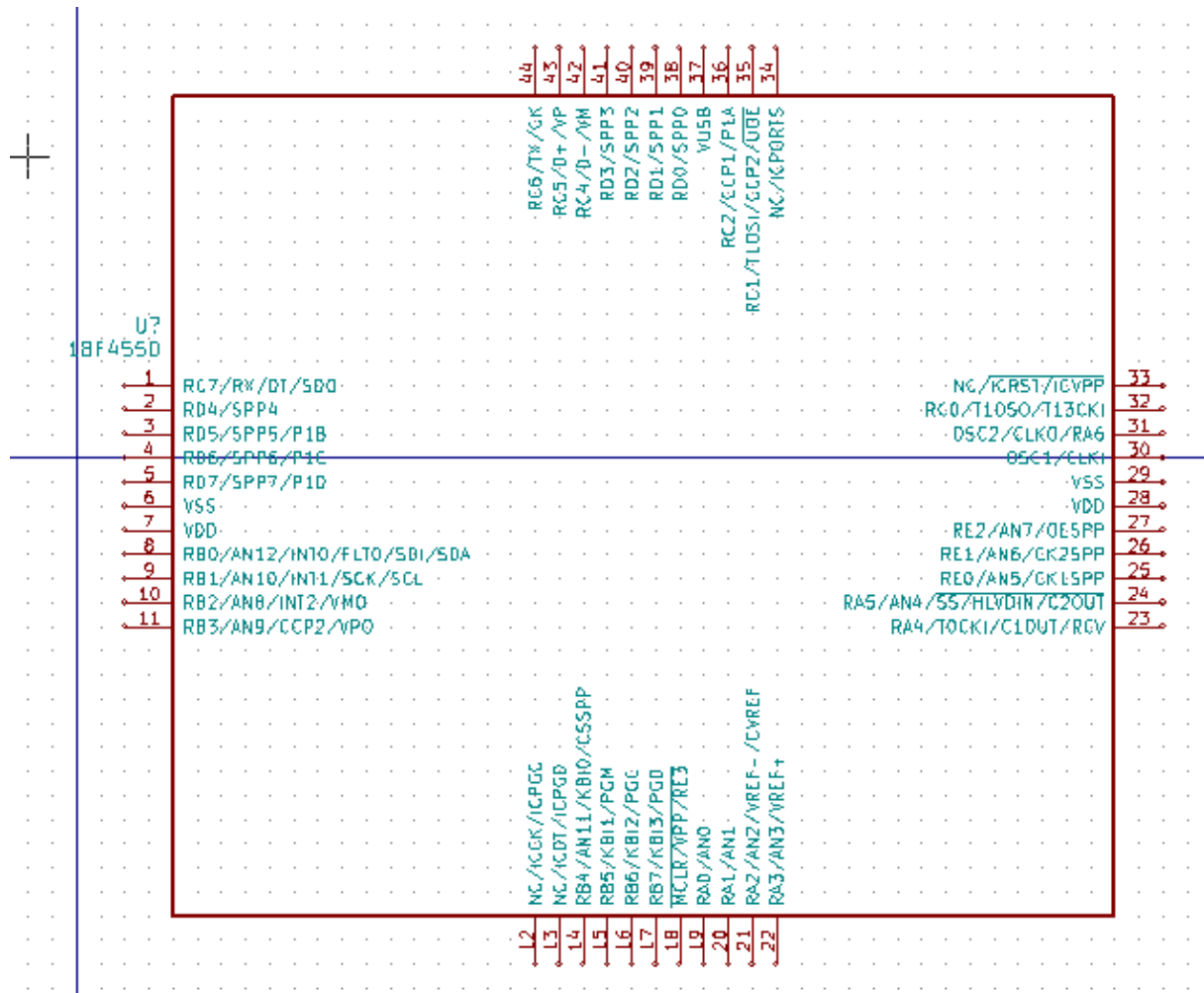
Donde:

```
>kipart todopic.csv -s num -f -o 18F4550_test.lib -w
>_
```

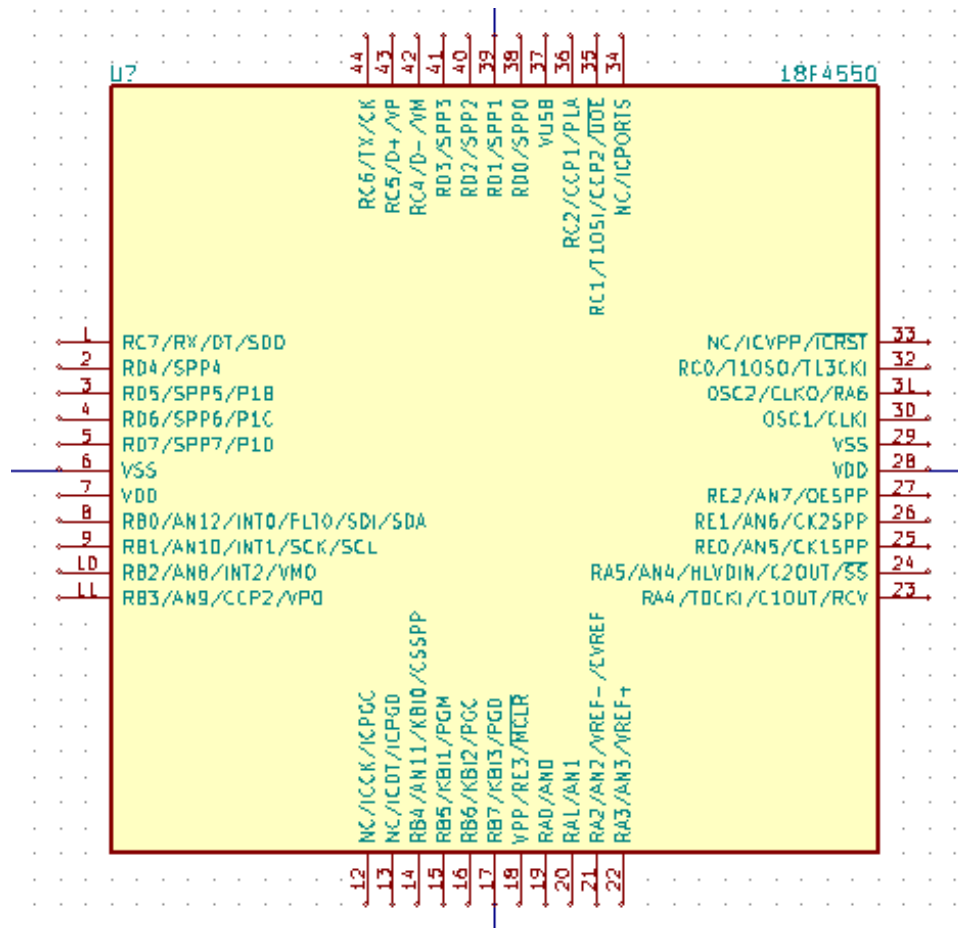
- todopic.csv es nuestro archivo origen
- El flag -s num le indica a KiPart que queremos que ordene los pines por número y en orden ascendente. Podemos también ordenarlos por nombre (name) o por fila (row), estos parámetros los podemos encontrar en la ayuda de KiPart.
- El flag -f le indica a KiPart que queremos que incluya el tipo de pin, orientación o representación en nuestro símbolo.
- El flag -o 18F4550_test.lib le indica a KiPart que queremos que el archivo de salida se llame 18F4550.lib
- El flag -w es para sobrescribir la librería en caso de que ya tengamos una con el

mismo nombre en el mismo directorio.

KiPart procede a armar nuestro simbolo, en mi maquina tardo cerca de 5 segundos asi que en la de ustedes puede tomar mas o menos tiempo. Una vez terminado podemos abrir el archivo xxx.lib en el Eschematic library editor de KiCAD para poder checarlo, añadimos el path del directorio donde tenemos el documento y añadimos la libreria, los pasos para hacer esto estan cubiertos en el excelente [tutorial](#) que escribio JUANJO, seleccionamos la libreria que se genero con KiPart, en mi caso se llama 18F4450_test y elegimos el componente que se genero, obtuve el siguiente simbolo:



Como podemos ver el simbolo se creo bien, tiene el nombre que le asignamos en el archivo .csv, los pines estan ordenados por su número, sin embargo lo podemos editar si asi lo queremos, KiPart ya hizo el trabajo pesado, al final el simbolo me quedo de la siguiente manera:



Procedemos a guardar los cambios y hemos terminado.

KiPart tiene mas flags que podemos usar, sin embargo no he tenido la necesidad de usarlas en todo caso recordar que una mini ayuda se encuentra en [documentación KiPart](#)

PS.

También podemos separar nuestro componente en unidades, es decir poner en una unidad todos los pines de alimentación y en otra todos los bidireccionales por ejemplo. Para hacer esto añadimos otra columna a nuestro archivo, esta columna se llamará Unit y debajo para cada pin especificamos a que unidad pertenecerá.

Guardamos el documento en formato .csv y nos vamos al directorio donde esta, abrimos la terminal y tecleamos el mismo comando que usamos antes para generar el simbolo, el flag -w nos servirá para sobrescribir la libreria vieja.

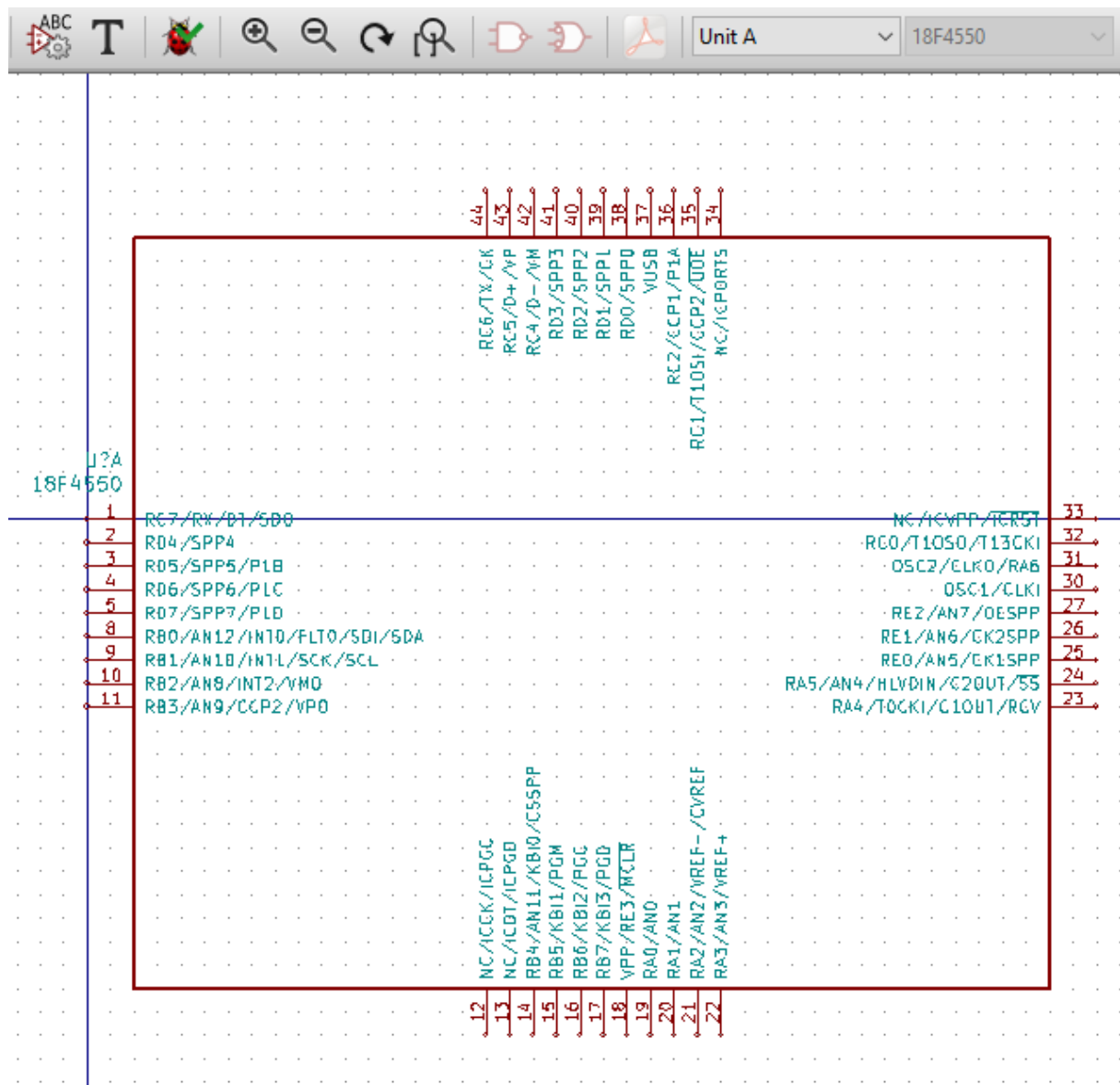
Código: [Seleccionar]

```
$kipart todopic.csv -s num
-f -o 18F4550_test.lib -w
```

y nos genera el siguiente simbolo, la unidad A:

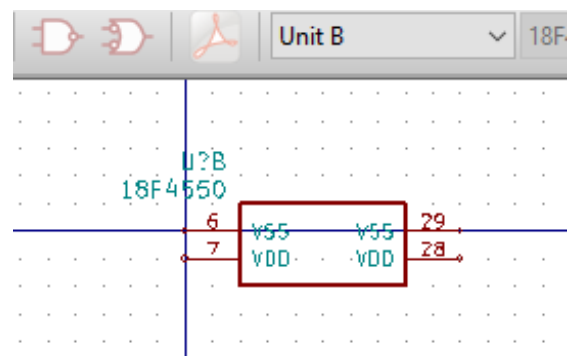
	A	B	C	D	E
1	18F4550				
2	Pin	Name	Type	Side	Unit
3		1RC7/RX/DT/SDO	bi	left	IO
4		2RD4/SPP4	bi	left	IO
5		3RD5/SPP5/P1B	bi	left	IO
6		4RD6/SPP6/P1C	bi	left	IO
7		5RD7/SPP7/P1D	bi	left	IO
8		6VSS	pwr	left	PWR
9		7VDD	pwr	left	PWR
10		8RB0/AN12/INT0/FLT0/SDI/SDO	bi	left	IO
11		9RB1/AN10/INT1/SCK/SCL	bi	left	IO
12		10RB2/AN8/INT2/VMO	bi	left	IO
13		11RB3/AN9/CCP2/VPO	bi	left	IO
14		12NC/ICCK/ICPGC	bi	bottom	IO
15		13NC/ICDT/ICPGD	bi	bottom	IO
16		14RB4/AN11/KBI0/CSSPP	bi	bottom	IO
17		15RB5/KBI1/PGM	bi	bottom	IO
18		16RB6/KBI2/PGC	bi	bottom	IO
19		17RB7/KBI3/PGD	bi	bottom	IO
20		18VPP/RE3/~MCLR	bi	bottom	IO
21		19RA0/AN0	bi	bottom	IO
22		20RA1/AN1	bi	bottom	IO
23		21RA2/AN2/VREF-/CVREF	bi	bottom	IO
24		22RA3/AN3/VREF+	bi	bottom	IO
25		33NC/ICVPP/~ICRST	bi	right	IO
26		32RC0/T1OSO/T13CKI	bi	right	IO
27		31QSC2/CLKO/RA6	bi	right	IO
28		30QSC1/CLKI	bi	right	IO
29		29VSS	pwr	right	PWR
30		28VDD	pwr	right	PWR
31		27RE2/AN7/QESPP	bi	right	IO
32		26RE1/AN6/CK2SPP	bi	right	IO
33		25RE0/AN5/CK1SPP	bi	right	IO
34		24RA5/AN4/HLVDIN/C2OUT/~S	bi	right	IO
35		23RA4/T0CKI/C1OUT/RCV	bi	right	IO
36		44RC6/TX/CK	bi	top	IO
37		43RC5/D+/VP	bi	top	IO
38		42RC4/D-/VM	bi	top	IO
39		41RD3/SPP3	bi	top	IO
40		40RD2/SPP2	bi	top	IO
41		39RD1/SPP1	bi	top	IO
42		38RD0/SPP0	bi	top	IO
43		37VUSB	bi	top	IO
44		36RC2/CCP1/P1A	bi	top	IO
45		35RC1/T1OSI/CCP2/~UOE	bi	top	IO
46		34NC/ICPORTS	bi	top	IO
47					

```
>kipart todopic.csv -s num -f -o 18F4550_test.lib -w
>_
```

Y la unidad B:

Por último no es necesario hacer el empaquetado de forma cuadrada y con los pines en cada uno de los lados, podemos poner los pines donde nosotros queramos. Como ejemplo el simbolo de un PCT2075, esta es la descripcion de sus pines:



6.2 Pin description

Table 3. Pin description for SO8, TSSOP8 and HWSO8

Symbol	Pin	Description
SDA	1	Digital I/O. I ² C-bus serial bidirectional data line; open-drain.
SCL	2	Digital input. I ² C-bus serial clock input.
OS	3	Overtemp Shutdown output; open-drain.
GND	4 ⁽¹⁾	Ground. To be connected to the system ground.
A2	5	Digital input. User-defined address bit 2.
A1	6	Digital input. User-defined address bit 1.
A0	7	Digital input. User-defined address bit 0.
V _{CC}	8	Power supply.

Ahora volcamos toda la información a nuestro archivo de texto:

	A	B	C	D	E
1	PCT2075				
2	Pin	Name	Type	Side	Style
3	1	SDA	bi	left	line
4	2	SCL	clk	left	clk
5	3	OS	oc	left	line
6	4	GND	pwr	left	line
7	5	A2	in	left	line
8	6	A1	in	left	line
9	7	A0	in	left	line
10	8	VCC	pwr	left	line

Podemos ver que el nombre del símbolo será PCT2075, pines del 1 – 8 con sus respectivos nombres, basados en la tabla del datasheet del componente asignamos el tipo de pines, donde bi son pines bidireccionales, clk es un pin de clock, oc es open_collector, pwr es pin de alimentación e in a entradas.

También vamos a especificar el estilo gráfico de nuestros pines, esto se especifica en la columna Style y las opciones que tenemos son las siguientes:

The style column specifies the graphic representation of the pin. The allowable pin styles are:

line,

inverted, inv

clock, clk, rising_clk

inverted_clock, inv_clk

input_low, inp_low, in_lw, in_b

clock_low, clk_low, clk_lw_clk_b

output_low, outp_low, out_lw, out_b

falling_edge_clock, falling_clk

non_logic, nl, analog

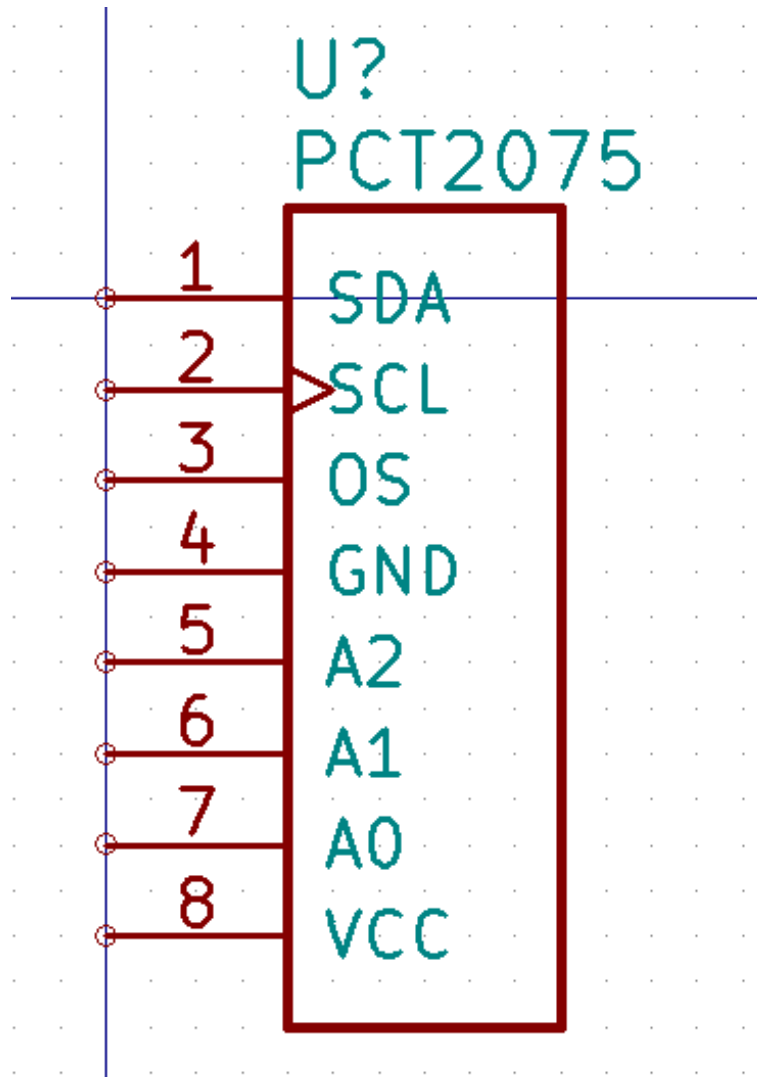
Guardamos nuestro archivo como .csv y abrimos una consola en el directorio donde tenemos el archivo. Para generar simbolo con KiPart hacemos como con el PIC18F4550, cambiando el nombre del archivo .csv y el archivo de salida.

Código: [\[Seleccionar\]](#)

```
$kipart PCT2075.csv -s num -f -o PCT2075.lib -w
```

y se genera nuestro simbolo, podemos abrirlo con el Eschematic library editor y obtuvimos lo siguiente:

```
>kipart PCT2075.csv -s num -f -o PCT2075.lib -w
```



Saludos 🤖