

Données en table projet

Nom :	Note :	/ 20
--------------	---------------	-------------

Classe :

1 L'entreprise de mon cousin

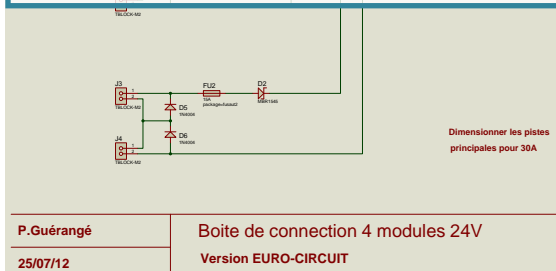
Votre cousin décide de se lancer dans la fabrication de cartes électroniques pour l'industrie solaire photovoltaïque. Il a déjà beaucoup de cartes électroniques à son catalogue, les commandes commencent à affluer et il a du mal à planifier la gestion des fournitures de pièces détachées, la fabrication des cartes électroniques, le contrôle qualité, la vente, les stocks, les clients

Bref il vous appelle à l'aide pour la gestion de son stock et de sa fabrication de cartes électroniques. Bien sûr le projet vous enthousiasme et vous avez accepté de relever ce challenge.

Pour vous mettre dans le bain votre cousin vous explique le processus de conception et de fabrication de ses cartes électroniques.

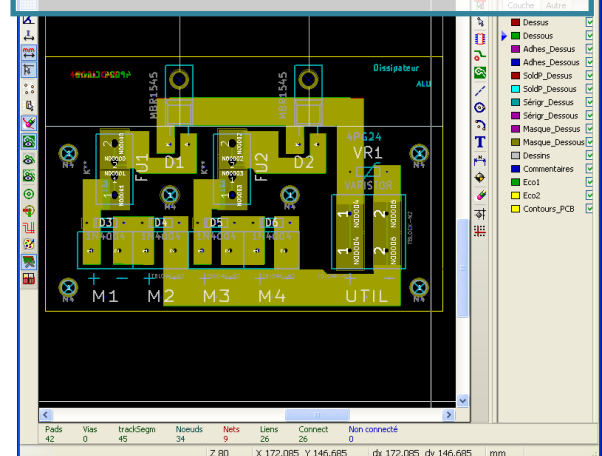
1.1 Le processus de fabrication d'une carte électronique

1) Le schéma électronique d'une carte de connexion de modules solaires



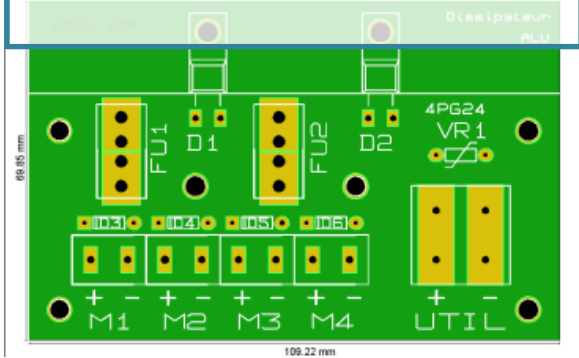
Logiciel de saisie de schéma électronique Proteus Isis.

2) Le routage du circuit imprimé avec un logiciel spécialisé



Logiciel de routage de cartes électroniques, (IUT de Grenoble PCBnew).

3) La fabrication des circuits imprimés sous-traitée.



4) L'achat des composants électroniques parmi des millions de références.

MBR1545 - SCHOTTKY DIODE, 15A, 45V, TO-220AB



Accueil > Semiconducteurs - Composants discrets > Diodes > Diodes Schottky > Diodes de redressement Schottky
Diodes de redressement Schottky: 3 866 produit(s) trouvés(s)

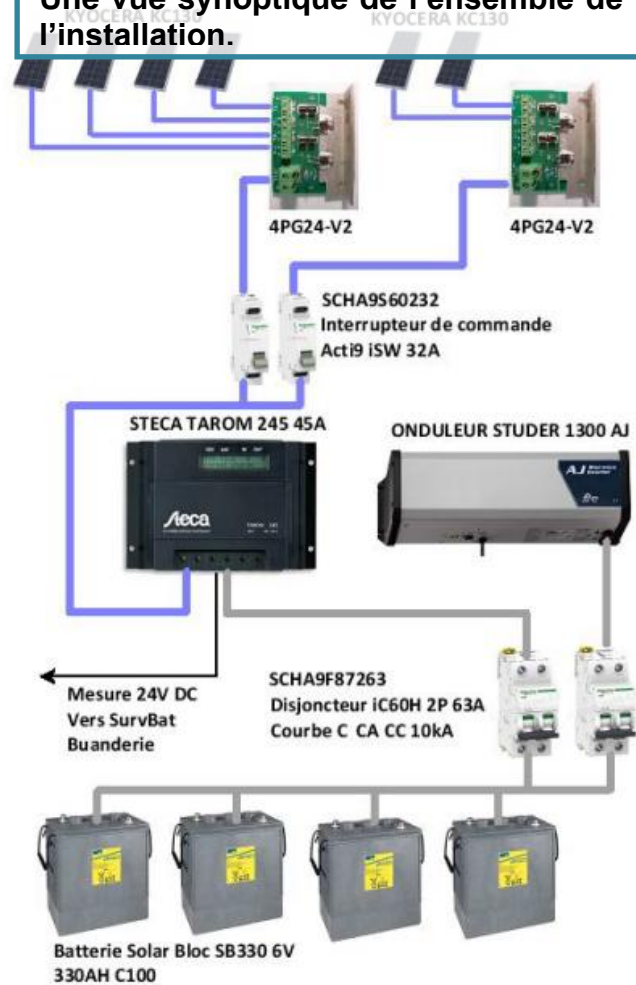
Disposition des filtres: GAUCHE

Fabricant	Tension, Vrrm	Courant, If moy.	Configuration diode	Type de boîtier de diode	Nombre de broches	Tension, Vf max.	Courant, Ifs max.	Température de fonctionnement max.	Gamme de produit
ON SEMICONDUCTOR (800)	10V (25)	1mA (2)	Common Cathode (1)	0603 (1)	2Broche(s) (2535)	250mV (2)	200mA (13)	70°C (1)	10BQ Series (2)
VISHAY (725)	12V (2)	10mA (2)	Double (2)	3-2A1A (10)	3Broche(s) (1197)	290mV (1)	500mA (36)	85°C (1)	IN5810 Series (9)
ROHM (634)	15V (26)	30mA (15)	Double anode commune (38)	Axial à broche (9)	4Broche(s) (33)	300mV (2)	310mA (1)	125°C (760)	IN5817 Series (6)
STMICROELECTRONICS (365)	20V (243)	70mA (2)	Double cathode commune (781)	CFP15 (SOT-1289) (2)	5Broche(s) (24)	310mV (2)	500mA (36)	150°C (2301)	IN5818 Series (6)
	25V (14)	75mA (2)		D-61 (1)	6Broche(s) (13)	320mV (4)	600mA (11)		IN5819 Series (7)
	30V (551)	100mA		DFN (19)	8Broche(s) (45)	330mV (9)			IN5820 Series (9)
	Min.	Min.	Min.	Min.	Min.	Min.	Min.	Min.	Min.
	Max.	Max.	Max.	Max.	Max.	Max.	Max.	Max.	Max.

5) La pose des panneaux solaires



Une vue synoptique de l'ensemble de l'installation.

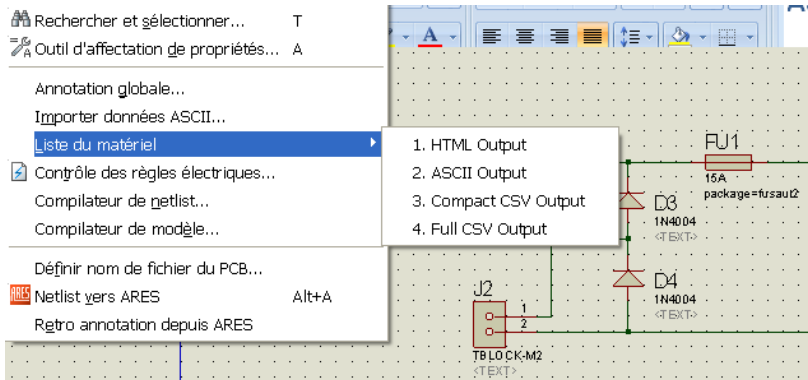


6) Câblage et essais du prototype de la carte de connexion.



1.2 Les données pour réaliser votre tâche

Le logiciel de conception de schéma électronique établit la liste des composants nécessaires pour chacune des cartes automatiquement. Le format Full CSV Output est utilisé.



Dans les chemins d'accès le caractère "." indique le répertoire courant.

.\DATA est donc un sous-répertoire de ce répertoire courant.

Les fichiers produits sont dans le dossier .\DATA

- Bill Of Materials For CARTE_4_MODULES_SOLAIRES_4in_2out.CSV
- Bill Of Materials For CARTE_DECODEUR_DCC.CSV
- Bill Of Materials For CARTE_I2C_8574.CSV
- Bill Of Materials For CARTE_I2C_8591.CSV
- Bill Of Materials For CARTE_I2C_LM75.CSV
- Bill Of Materials For CARTE_MyAVR.CSV

Un extrait de l'un des fichiers :

1	Category, Reference, Value, Code
2	Resistors, "R1", 10k,
3	Resistors, "R2", 10k,
4	Resistors, "R3", 150,



2 Données en table : mise en œuvre

2.1 Lire un fichier de données

Voilà le code qui permet de lire le contenu de l'un des fichiers CSV produit par Proteus :

```
1 #-*- coding: utf-8 -*-
2 #
3 # donnees_en_table_lecture_csv.py
4
5 from os import chdir
6 from os import listdir
7
8 # Se déplacer dans le dossier en cours si besoin
9 path = r'P:\PRO\USB\NSI\NSI DONNEES EN TABLE\EXEMPLES POUR TP'
10 chdir(path)
11
12 import csv
13
14 # On créer la liste vide
15 table=[]
16
17 # Lecture de la table csv, le résultat est dans une
18 # liste de listes
19
20 sous_dossier = '.\DATA\'
21
22 nom_carte = 'Bill Of Materials For CARTE_I2C_LM75.CSV'
23
24 with open(sous_dossier + nom_carte, encoding="utf8") as myFile:
25     reader = csv.reader(myFile)
26     for row in reader:
27         table.append(row)
28
29
30 for x in range(0, len(table)):
31     print(table[x])
```

Ce fichier est disponible :


 donnees_en_table_lecture_CSV.py

Nous obtenons la liste brute de la table voir sur la page suivante :

Liste des composants de la carte

```
'Bill Of Materials For CARTE_I2C_LM75.CSV' >>> (executing lines 1 to 31 of "donnees_en_table_corrige_1.py")
['Category', 'Reference', 'Value', 'Code']
['Resistors', 'R1', '10k', '']
['Resistors', 'R2', '10k', '']
['Resistors', 'R3', '150', '']
['Resistors', 'R4', '150', '']
['Integrated Circuits', 'U3', 'LM75', '']
['Miscellaneous', 'J1', 'I2C_VOX', '']
['Miscellaneous', 'J2', 'I2C_VOX', '']
['Miscellaneous', 'J3', 'BVIS3', '']
['Miscellaneous', 'JP1', 'JUMPER', '']
['Miscellaneous', 'JP2', 'JUMPER', '']
['Miscellaneous', 'JP3', 'JUMPER2', '']
['Miscellaneous', 'JP4', 'JUMPER2', '']
['Miscellaneous', 'JP5', 'JUMPER2', '']
['Miscellaneous', 'PT0S', 'PTEST', '']
['Miscellaneous', 'PTSCL', 'PTEST', '']
['Miscellaneous', 'PTSDA', 'PTEST', '']
```

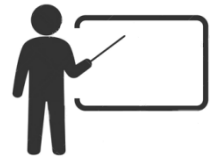
2.2 Utiliser des sélections

Nous travaillons à partir du script précédent :  donnees_en_table_lecture_CSV.py

Il n'est pas demandé d'affichage sophistiqué pour le moment.



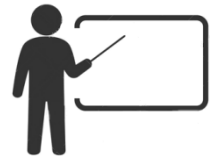
Script_projet_data_en_table_1. Créer un script qui donne la liste de toutes les résistances présentes dans la carte CARTE_I2C_LM75



Résultat attendu : Liste des résistances
['Resistors', 'R1', '10k', '']
['Resistors', 'R2', '10k', '']
['Resistors', 'R3', '150', '']
['Resistors', 'R4', '150', '']



Script_projet_data_en_table_2. Créer un script qui donne la liste de toutes les valeurs différentes de la carte. On évitera d'afficher le descripteur. On utilisera une écriture en compréhension (voir cours types construits 2.3.a p.9).

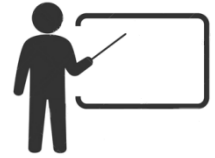


Résultat attendu :

Liste des composants
{'I2C_VOX', '10k', 'JUMPER', 'BVIS3', 'PTEST', 'JUMPER2', '150', 'LM75'}



Script_projet_data_en_table_3. Créer un script qui donne pour chaque valeur des composants la quantité nécessaire à la fabrication de la carte.



Résultat attendu :

```
Liste avec quantité pour chaque valeur de composants
Composant      I2C_VOX      quantité : 2
Composant      10k          quantité : 2
Composant      JUMPER       quantité : 2
Composant      BVIS3        quantité : 1
Composant      PTEST        quantité : 3
Composant      JUMPER2      quantité : 3
Composant      150          quantité : 2
Composant      LM75         quantité : 1
```

2.3 Améliorations

Vous avez montré votre travail à votre cousin, il est très satisfait de l'aide que vous lui apportez. Pour pouvoir immédiatement utiliser vos résultats il vous demande d'en améliorer la présentation, il veut faire des fiches pour aider à la réalisation des cartes électroniques, ces fiches seront distribuées aux ouvriers chargés de câbler les prototypes.

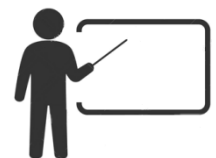
Voilà le résultat attendu :

LISTE DES COMPOSANTS DE LA CARTE : CARTE_I2C_LM75

Category	Reference	Value
Resistors	R1	10k
Resistors	R2	10k
Resistors	R3	150
Resistors	R4	150
Integrated Circuits	U3	LM75
Miscellaneous	J1	I2C_VOX
Miscellaneous	J2	I2C_VOX
Miscellaneous	J3	BVIS3
Miscellaneous	JP1	JUMPER
Miscellaneous	JP2	JUMPER
Miscellaneous	JP3	JUMPER2
Miscellaneous	JP4	JUMPER2
Miscellaneous	JP5	JUMPER2
Miscellaneous	PT0S	PTEST
Miscellaneous	PTSC	PTEST
Miscellaneous	PTSDA	PTEST



Script_projet_data_en_table_4. Créer un script qui réalise la fiche demandée.



Vous aidez votre cousin, mais il n'est pas informaticien. Dans l'analyse plus approfondie des programmes à réaliser vous souhaitez développer une bibliothèque de fonctions dédiées à la gestion des stocks de l'entreprise.


```

#-*- coding: utf-8 -*-
#
#*****
# LECTURE D'UNE TABLE CSV DANS UNE LISTE DE LISTES ET EXPLOITATION
#*****
#
# Liste de fichiers.py

import csv
from os import chdir
from os import listdir

# Se déplacer dans le dossier en cours si besoin
path = r'P:\PRO\USB\NSI\NSI DONNEES EN TABLE\EXEMPLES POUR TP\DATA'
chdir(path)

liste_des_fichiers_CSV = []

liste_des_fichiers = listdir()

for x in range(len(liste_des_fichiers)):
    if liste_des_fichiers[x][-4:] == '.CSV':
        liste_des_fichiers_CSV.append(liste_des_fichiers[x])

for x in range(len(liste_des_fichiers_CSV)):
    print(liste_des_fichiers_CSV[x])

```

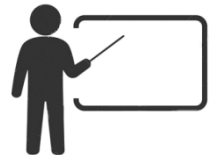


Q1. Que réalise la fonction listdir du module os ?

3.2 Fournir la liste de tous les composants de toutes les cartes



Script_projet_data_en_table_6. Créer un script qui réalise la liste des composants de toutes les cartes stockées dans le répertoire ./DATA/.



Il n'est pas nécessaire de réaliser une impression fouillée à ce stade. On demande la liste de tous les composants sans gérer ni les doublons ni ordonner les résultats.

Liste de tous les composants

```

['Resistors', 'R1', '330', '']
['Resistors', 'R2', '330', '']
['Capacitors', 'C1', '10uF', '']
['Capacitors', 'C2', '100nF', '']
['Capacitors', 'C3', '100nF', '']
['Capacitors', 'C4', '100nF', '']
['Integrated Circuits', 'U3', '7805', '']
['Diodes', 'D1', '1N4001', '']
['Miscellaneous', 'BP0', 'BPKSA', '']
['Miscellaneous', 'BP1', 'BPKSA', '']
['Miscellaneous', 'BP2', 'BPKSA', '']
['Miscellaneous', 'BP3', 'BPKSA', '']
['Miscellaneous', 'GND1', 'D2MM', '']
['Miscellaneous', 'GND2', 'D2MM', '']
['Miscellaneous', 'VALIM', 'D2MM', '']
['Miscellaneous', 'VCC', 'D2MM', '']

['Resistors', 'R3', '150', '']
['Resistors', 'R4', '150', '']
['Integrated Circuits', 'U3', 'LM75', '']
['Miscellaneous', 'J1', 'I2C_VOX', '']
['Miscellaneous', 'J2', 'I2C_VOX', '']
['Miscellaneous', 'J3', 'BVIS3', '']
['Miscellaneous', 'JP1', 'JUMPER', '']
['Miscellaneous', 'JP2', 'JUMPER', '']
['Miscellaneous', 'JP3', 'JUMPER2', '']
['Miscellaneous', 'JP4', 'JUMPER2', '']
['Miscellaneous', 'JP5', 'JUMPER2', '']
['Miscellaneous', 'PT0S', 'PTEST', '']
['Miscellaneous', 'PTSCL', 'PTEST', '']
['Miscellaneous', 'PTSDA', 'PTEST', '']

```

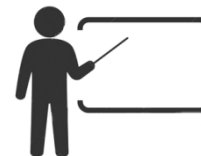
4 Synthèse : éditions de factures

Pour finir nous allons éditer les factures clients. Les prix des différents composants sont dans le fichier CATALOGUE PRIX COMPOSANTS.csv

Attention ce sont des prix professionnels donc hors taxes. Pour calculer un prix TTC il faut multiplier par 1,20.



Script_projet_data_en_table_7. Créer un script qui réalise la liste des composants et la facturation d'une carte au choix, présenter les résultats selon l'exemple ci-dessous.



Exemple de résultat :

```
-----
| Bill Of Materials For CARTE_I2C_8574.CSV |
-----
|          Category -      Valeur -      Quantité |
-----
|  Integrated Circuits      PCF8574          1 |
|  Miscellaneous           I2C_VOX          2 |
|    Resistors              150            2 |
|  Miscellaneous           JUMPER2          3 |
|  Miscellaneous           JUMPER          2 |
|    Resistors              330            8 |
|  Miscellaneous           PTEST          2 |
|    Resistors              10k            2 |
|    LED                    JAUNE           8 |
-----
```

Et la facturation :

```
-----
Facture de la carte   :      CARTE_I2C_8574
-----
|  Integrated Circuits |      PCF8574 |  1 |  1.220 €HT |  1.220 €HT |
|  Miscellaneous     |      I2C_VOX |  2 |  0.450 €HT |  0.900 €HT |
|    Resistors        |         150  |  2 |  0.053 €HT |  0.106 €HT |
|  Miscellaneous     |      JUMPER2 |  3 |  0.250 €HT |  0.750 €HT |
|  Miscellaneous     |      JUMPER  |  2 |  0.230 €HT |  0.460 €HT |
|    Resistors        |         330  |  8 |  0.058 €HT |  0.464 €HT |
|  Miscellaneous     |      PTEST  |  2 |  0.120 €HT |  0.240 €HT |
|    Resistors        |         10k  |  2 |  0.051 €HT |  0.102 €HT |
|    LED              |      JAUNE  |  8 |  0.247 €HT |  1.976 €HT |
-----
Total HT              :      6.22 €HT
Total TTC net à payer :      7.46 €TTC
-----
```



