

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Meccanica



Allegato 1

Codici per automatizzazione del processo

Relatore:

Prof. Fabio Freschi

Candidato:

Andrea Ravetti

Azienda:

Roj srl

Anno Accademico 2023-2024

Contesto

Il processo di conversione dati prevede l'utilizzo di tre file contenuti nella stessa cartella. Essi devono essere:

- un file sorgente in formato foglio di calcolo (.xls);
- un file ausiliario in formato foglio di calcolo con attivazione macro (.xlsm), che contiene diverse macro;
- un file Python (.py) che governa la programmazione.

Il file sorgente è un file con struttura matriciale che possiede su ogni riga l'indicazione di uno specifico test point sulla scheda, la sua tipologia, il suo posizionamento (in x e in y), la dimensione della sua piazzola, il canale della testata a cui deve essere collegato oltre che alcuni dati non necessari per le operazioni da realizzare.

Il funzionamento dei codici di conversione prevede inizialmente l'attivazione del codice Python, che trasferisce i dati presenti nel file sorgente nel file ausiliario e successivamente comanda l'attivazione di opportune macro, richiedendo opportuni dati di input.

L'output che devono generare tali file sono due file di testo, in formato .kyn e .emn.

Nelle sezioni di seguito si riportano i codici redatti.

Codice Python

```
import openpyxl
```

```
import xlrd
```

```
import win32com.client as win32
```

```
import os
```

```
#DEFINIZIONE DELLE FUNZIONI
```

```
#prima funzione: macro che pulisce foglio di lavoro
```

```

def pulizia(destination_file, macro_pulizia):
    # Verifica se il file di destinazione esiste
    if not os.path.exists(destination_file):
        print(f"Il file di destinazione generatore_netlist non esiste.")
        return

    # Apri il file di destinazione con Excel
    excel = win32.Dispatch('Excel.Application')
    wb = excel.Workbooks.Open(destination_file)
    # applica la macro per la pulizia del foglio
    excel.Application.Run(macro_pulizia)
    # chiudi e esci da Excel
    wb.Close(SaveChanges=True)
    excel.Application.Quit()
    print(f"La macro per la pulizia del foglio è stata eseguita. 1/5.")

# seconda funzione: copia il foglio esportato da tool di cablaggio nel file Excel lavorazione_dati
def copy_sheet_xls_to_xlsm(source_file, destination_file, destination_sheet_name):
    # Verifica di nuovo se il file di destinazione esiste
    if not os.path.exists(destination_file):
        print(f"Il file di destinazione {destination_file} non esiste, per la copiatura.")
        return

    # Controlla l'estensione del file di origine che potrebbe essere .xls o .xlsm
    if source_file.endswith('.xls'):
        # Carica il file di origine .xls
        src_wb = xlrd.open_workbook(source_file)
        src_ws = src_wb.sheet_by_index(0)
        # Ottieni i dati dal foglio di origine
        src_data = []
        for row_idx in range(src_ws.nrows):
            row = src_ws.row_values(row_idx)
            src_data.append(row)

```

else:

```
# Carica il file di origine .xslm
```

```
src_wb = openpyxl.load_workbook(source_file, keep_vba=True)
```

```
src_ws = src_wb.active
```

```
# Ottieni i dati dal foglio di origine
```

```
src_data = []
```

```
for row in src_ws.iter_rows(values_only=True):
```

```
    src_data.append(row)
```

```
# Carica il file di destinazione .xslm
```

```
dest_wb = openpyxl.load_workbook(destination_file, keep_vba=True)
```

```
dest_ws = dest_wb[destination_sheet_name]
```

```
# Copia i dati dal foglio di origine al foglio di destinazione
```

```
for row in src_data:
```

```
    dest_ws.append(row)
```

```
# Salva il file di destinazione
```

```
dest_wb.save(destination_file)
```

```
print(f'Il foglio è stato copiato dal file 'TPList' al file 'lavorazione_posizioni' nel foglio  
'{destination_sheet_name}'. 2/5.")
```

terza funzione: attiva tutte le macro del foglio Excel che convertono il file di partenza per generare la maschera del file di testo .kyn

```
def macro_lavoro(destination_file, macro2, macro3, macro4, macro5, macro6, macro7):
```

```
    #apre excel
```

```
    excel = win32.Dispatch('Excel.Application')
```

```
    wb = excel.Workbooks.Open(destination_file)
```

```
    #macro sfoltimento file sorgente
```

```
    excel.Application.Run(macro2)
```

```
    #macro generatore net
```

```
    excel.Application.Run(macro3)
```

```
    #macro generatore part
```

```
    excel.Application.Run(macro4)
```

```
    #macro unione fogli netlist
```

```

excel.Application.Run(macro5)
#macro generatore posizioni
excel.Application.Run(macro6)
#macro unione fogli posizioni
excel.Application.Run(macro7)
#chiude excel
wb.Close(SaveChanges=True)
excel.Application.Quit()
print(f"Le macro per la conversione dei dati sono state eseguite. 3/5. ")

```

quarta funzione: esporto il foglio finale generato nell'Excel in un file di testo .kyn, adatto per la lettura da parte del CAD elettronico

faccio inoltre delle conversioni di formato

```

def export_fifth_sheet_to_text(destination_file, text_file_1):
    try:
        # Apre il file Excel
        w_b = openpyxl.load_workbook(destination_file, read_only=True)
        w_s = w_b[netlist_sheet]
        # Prende l'ultimo foglio di lavoro
        # Prende i dati dal foglio di lavoro e fa le conversioni
        data = []
        for row in w_s.iter_rows(values_only=True):
            converted_row = [str(cell).replace('None','') for cell in row]
            data.append(''.join(converted_row))
        # Scrive i dati in un file di testo
        with open(text_file_1, 'w') as file:
            for row in data:
                file.write(row + '\n')
        print("I dati sono stati esportati correttamente nel file di testo", text_file_1,"4/5.")
    except Exception as e:
        print("Errore durante l'esportazione dei dati nel file di testo:", e)

```

quinta funzione: esporto il foglio finale generato in Excel in un file di testo .emn, adatto per la lettura da parte del Software CAD elettronico

faccio inoltre delle conversioni di formato

def export_last_sheet_to_text(destination_file, text_file_2):

try:

Apre il file Excel

wb = openpyxl.load_workbook(destination_file, read_only=True)

Prende l'ultimo foglio di lavoro

last_sheet_name = wb.sheetnames[-1]

ws = wb[last_sheet_name]

Prende i dati dal foglio di lavoro e fa le conversioni

data = []

for row in ws.iter_rows(values_only=True):

converted_row = []

for cell in row:

if isinstance(cell, float):

converted_cell = f'{cell:.3f}'.replace('None', '').replace('\n', '\n')

else:

converted_cell = str(cell).replace('None', '').replace('\n', '\n')

converted_row.append(converted_cell)

data.append(''.join(converted_row))

Scrive i dati in un file di testo

with open(text_file_2, 'w') as file:

for row in data:

file.write(row + '\n')

print("I dati sono stati esportati correttamente nel file di testo", text_file_2, "5/5.")

except Exception as e:

print("Errore durante l'esportazione dei dati nel file di testo:", e)

#INSERIMENTO DEI PERCORSI AI FILE

source_file = r' *Inserire il percorso del file sorgente* '

destination_file = r' *Inserire il percorso del file destinazione* '

destination_sheet_name = ' *Inserire nome del foglio di destinazione (in questo caso “grezzi”)* '

netlist_sheet = ' *Inserire nome del foglio contenente le netlist (in questo caso “n_finale”)* '

Inserire nomi delle macro da eseguire, con il relativo modulo. Esempio per il file utilizzato:

macro_pulizia = 'Modulo2.assetta_1'

macro2 = 'Modulo1.estrappola_dati_2'

macro3 = 'Modulo3.genera_net_3_1'

macro4 = 'Modulo4.genera_part_4_1'

macro5 = 'Modulo5.genera_posizioni_3_2'

macro6 = 'Modulo6.unisci_parti_5_1'

macro7 = 'Modulo7.unisci_parti_5_2'

text_file_1 = "netlist.kyn"

text_file_2 = "posizioni.emn"

#ESECUZIONE DEL FILE

Esegui prima la macro di pulizia

pulizia(destination_file, macro_pulizia)

Poi copia il contenuto dal file originale a quello di lavorazione

copy_sheet_xls_to_xlsm(source_file, destination_file, destination_sheet_name)

#esegui altre macro per l'elaborazione

macro_lavoro(destination_file, macro2, macro3, macro4, macro5, macro6, macro7)

#converti in file di testo .kyn

export_fifth_sheet_to_text(destination_file, text_file_1)

#converti in file di testo .emn

export_last_sheet_to_text(destination_file, text_file_2)

Macro 1: assetta_1

```
Sub assetta_1()  
    'pulisci tutti i fogli  
    ThisWorkbook.Sheets("grezzi").Cells.ClearContents  
    ThisWorkbook.Sheets("n_net").Cells.ClearContents  
    ThisWorkbook.Sheets("n_part").Cells.ClearContents  
    ThisWorkbook.Sheets("n_finale").Cells.ClearContents  
    ThisWorkbook.Sheets("p_posizioni").Cells.ClearContents  
    ThisWorkbook.Sheets("p_finale").Cells.ClearContents  
End Sub
```

Macro 2: estrapola_dati_2

```
Sub estrapola_dati_2()  
    ThisWorkbook.Sheets(1).Activate  
  
    'PRIMA PARTE: ELIMINA IMMAGINI DAL FILE  
    Dim shp As Shape  
    ' Cicla attraverso tutte le forme presenti nel foglio  
    For Each shp In ActiveSheet.Shapes  
        ' Verifica se la forma è un'immagine  
        If shp.Type = msoPicture Then  
            ' Elimina l'immagine  
            shp.Delete  
        End If  
    Next shp  
  
    'SECONDA PARTE: ELIMINA I BOTTONI DI COMANDO  
    Dim btn As OLEObject
```

' Cicla attraverso tutti i bottoni di comando presenti nel foglio

For Each btn In ActiveSheet.OLEObjects

' Verifica se l'oggetto è un bottone di comando

If btn.ProgID = "Forms.CommandButton.1" Then

' Elimina il bottone di comando

btn.Delete

End If

Next btn

'TERZA PARTE: ELIMINA LE COLONNE CON LE INFO CHE NON MI INTERESSANO

Dim col As Integer

Dim limite_colonne As Integer

limite_colonne = ActiveSheet.UsedRange.Columns.Count

For col = limite_colonne To 1 Step -1

If Cells(11, col).Value = "." Or Cells(11, col).Value = "Z [µm]" Or Cells(11, col).Value = "Checked" Or Cells(11, col).Value = "Extra Down [µm]" Or Cells(11, col).Value = "Pad Code" Or Cells(11, col).Value = "Case" Or Cells(11, col).Value = "Pin" Or Cells(11, col).Value = "Drawing Ref." Or Cells(11, col).Value = "TP Name" Or Cells(11, col).Value = "Net Name" Or Cells(11, col).Value = "Long Net Name" Or Cells(11, col).Value = "Digital Class" Or Cells(11, col).Value = "Contact" Or Cells(11, col).Value = "Probe Type" Or Cells(11, col).Value = "Capacitor list" Or Cells(11, col).Value = "Source" Or Cells(11, col).Value = "Board Side" Or Cells(11, col).Value = "Net Class" Or Cells(11, col).Value = "Low Capacitor" Or Cells(11, col).Value = "Insulated" Or Cells(11, col).Value = "Message" Or Cells(11, col).Value = "Referral Conn. Name" Or Cells(11, col).Value = "Referral Conn. Pin" Then

Columns(col).Delete

End If

Next col

'QUARTA PARTE: ELIMINA LE PRIME 10 RIGHE

' Seleziona e elimina le prime 10 righe del foglio attivo

Rows("1:10").Delete

'QUINTA PARTE: ELIMINA RIGHE IN BASE AI TP TYPE INUTILI

Dim lastRow As Long

```

Dim typeCol As Long
Dim i As Long
Dim j As Long
Dim targetValue As Long
' Trova l'ultima riga nella colonna "A"
lastRow = Cells(Rows.Count, "A").End(xlUp).Row
'Cerca la colonna con titolo "TP Type"
typeCol = 0
For i = 1 To Cells(1, Columns.Count).End(xlToLeft).Column
    If Cells(1, i).Value = "TP Type" Then
        'Trovata la colonna col titolo "TP Type" salvo l'indice di quella colonna ed esco dal for
        typeCol = i
        Exit For
    End If
Next i
'Verifico con un flag di aver trovato la colonna "TP type"
If typeCol > 0 Then
    ' Cicla sulle righe della colonna "TP type" a partire dalla seconda riga (che è quella da cui
    iniziano i numeri)
    ' In realtà parto dal fondo perchè conviene per la programmazione
    For i = lastRow To 2 Step -1
        ' Verifica se il "TP type" è non automatizzabile e in quel caso lo elimina
        If Cells(i, typeCol).Value = "Digital" Or Cells(i, typeCol).Value = "Power" Or Cells(i,
        typeCol).Value = "Master unpositioned" Or Cells(i, typeCol).Value = "Kelvin unpositioned" Or
        Cells(i, typeCol).Value = "Electroscan MUX" Or Cells(i, typeCol).Value = "Dummy Board" Then
            ' Se il valore corrisponde, elimina la riga
            Rows(i).Delete
        ElseIf Cells(i, typeCol).Value = "Dummy" Then
            ' Se ho un dummy devo cancellare anche il suo master
            targetValue = Cells(i, 2).Value - 2
            Rows(i).Delete
        End If
    Next i
    ' Elimina tutte le righe che hanno il valore target nella seconda colonna

```

```

lastRow = Cells(Rows.Count, "A").End(xlUp).Row
For j = lastRow To 2 Step -1
    If Cells(j, 2).Value = targetValue Then
        Rows(j).Delete
    Exit For
End If
Next j
End If
Next i
End If

```

'QUINTA PARTE BIS: SELEZIONO TRA PROBE SIDE TOP E BOTTOM

'chiedo all'utente se vuole la scheda top o bottom e in base a quella cancello i tp opposti

```
Dim SideCol As Long
```

```
Dim Lato As String
```

```
lastRow = Cells(Rows.Count, "A").End(xlUp).Row
```

'Cerca la colonna con titolo "Probe Side"

```
SideCol = 0
```

```
For i = 1 To Cells(1, Columns.Count).End(xlToLeft).Column
```

'Trovata la colonna col titolo "Probe Measure" salvo l indice di quella colonna ed esco dal for

```
If Cells(1, i).Value = "Probe Side" Then
```

```
    SideCol = i
```

```
Exit For
```

```
End If
```

```
Next i
```

'Verifico con un flag di aver trovato la colonna "Probe Measure"

```
If SideCol > 0 Then
```

```
    Lato = InputBox("Inserisci 't' per generare scheda TOP, 'b' per generare scheda BOTTOM ",
"Lato scheda")
```

```
For i = lastRow To 2 Step -1
```

```
    If Lato = "t" Then
```

```

    If Cells(i, SideCol).Value = "Bottom" Then
        ' Se il valore corrisponde, elimina la riga
        Rows(i).Delete
    End If
ElseIf Lato = "b" Then
    If Cells(i, SideCol).Value = "Top" Then
        ' Se il valore corrisponde, elimina la riga
        Rows(i).Delete
    End If
End If
Next i
End If

```

'QUINTA PARTE TRIS: CONVERTO LE QUOTE IN MM

```

Cells(1, 4).Value = "X [mm]"
Cells(1, 5).Value = "Y [mm]"
lastRow = Cells(Rows.Count, "A").End(xlUp).Row
For j = 2 To lastRow
    Cells(j, 4).Value = Cells(j, 4).Value / 1000
    Cells(j, 5).Value = Cells(j, 5).Value / 1000
Next j

```

'QUINTA PARTE QUATRIS: OFFSETTO LE QUOTE

' sulle quote sottraggo gli offset di Leonardo e aggiungo lo spessore della bandella + il vuoto tra scheda e bandella

```

Dim offsetX As Double
Dim offsetY As Double
Dim bandellaX As Double
Dim bandellaY As Double
Dim k As Long

```

offsetX = -10000

offsetY = -10000

bandellaX = -10000

bandellaY = -10000

' Chiedi all'utente il valore da sottrarre

offsetX = InputBox("Inserisci l'offset in X in mm (formato abc,de):", "offsetX")

' Controlla se l'utente ha premuto Annulla o ha lasciato vuoto

If offsetX = -10000 Then

 MsgBox "Errore! Se l'offset è nullo inserire il valore 0", vbInformation

 Exit Sub

End If

offsetY = InputBox("Inserisci l'offset in Y in mm (formato abc,de):", "offsetY")

If offsetY = -10000 Then

 MsgBox "Errore! Se l'offset è nullo inserire il valore 0", vbInformation

 Exit Sub

End If

' Chiedi all'utente il valore da sottrarre

bandellaX = InputBox("Inserisci lo spessore di bandella + spazio vuoto LATO ESTERNO in mm (formato abc,de):", "bandella lato esterno")

' Controlla se l'utente ha premuto Annulla o ha lasciato vuoto

If bandellaX = -10000 Then

 MsgBox "Errore! Se la bandella non è presente inserire il valore 0", vbInformation

 Exit Sub

End If

bandellaY = InputBox("Inserisci lo spessore di bandella + spazio vuoto LATO TESTATA in mm (formato abc,de):", "bandella lato testata")

If bandellaY = -10000 Then

 MsgBox "Errore! Se la bandella non è presente inserire il valore 0", vbInformation

 Exit Sub

End If

' Sottrai il valore da ogni cella nella colonna C

'NB: siccome ho ruotato la scheda di 90°, l'offset su x va su y e viceversa

For k = 2 To lastRow

 If IsNumeric(Cells(k, 4).Value) Then

 Cells(k, 4).Value = Cells(k, 4).Value - offsetY + bandellaY - 1.5

 End If

 If IsNumeric(Cells(k, 5).Value) Then

 Cells(k, 5).Value = Cells(k, 5).Value - offsetX + bandellaX

 End If

Next k

'SESTA PARTE: SE TROVA DELLE PROBE MEASURE 0 SUI KELVIN LI ASSEGNA AI
MASTER CORRISPONDENTI, SE NO LI CHIEDE IN INPUT

'valuta se eliminare

Dim measureCol As Long

lastRow = Cells(Rows.Count, "A").End(xlUp).Row

'Cerca la colonna con titolo "Probe Measure"

measureCol = 0

For i = 1 To Cells(1, Columns.Count).End(xlToLeft).Column

 'Trovata la colonna col titolo "Probe Measure" salvo l'indice di quella colonna ed esco dal for

 If Cells(1, i).Value = "Probe Measure" Then

 measureCol = i

 Exit For

 End If

Next i

'Verifico con un flag di aver trovato la colonna "Probe Measure"

If measureCol > 0 Then

 For i = lastRow To 2 Step -1

 If Cells(i, measureCol).Value = "0" Or Cells(i, measureCol).Value = "" Then

 ' Se il valore corrisponde, elimina la riga

 For j = lastRow To 2 Step -1

```
        If Cells(i, 3).Value = "Kelvin" And Cells(j, 3).Value = "Master" And Cells(j, 1).Value =  
Cells(i, 1).Value And Cells(j, 4).Value = Cells(i, 4).Value And Cells(j, 5).Value = Cells(i, 5).Value  
Then
```

```
        Cells(i, 6).Value = Cells(j, 6).Value
```

```
    End If
```

```
Next j
```

```
End If
```

```
If Cells(i, measureCol).Value = "0" Or Cells(i, measureCol).Value = "" Then
```

```
    Cells(i, 6).Value = InputBox("Inserisci la dimensione del ricettacolo del TP " & Cells(i,  
1).Value & ", di coordinate X " & Cells(i, 4).Value & " mm e Y " & Cells(i, 5).Value & " mm",  
"Probe Dimension")
```

```
End If
```

```
Next i
```

```
End If
```

```
'SETTIMA PARTE: ELIMINA COLONNA TP TYPE e PROBE SIDE
```

```
For col = 1 To 7
```

```
    If Cells(1, col).Value = "TP Type" Or Cells(1, col).Value = "Probe Side" Then
```

```
        Columns(col).Delete
```

```
        col = col - 1
```

```
    End If
```

```
Next col
```

```
'OTTAVA PARTE: CONTROLLA E INVERTE LE PRIME 2 COLONNE
```

```
Dim rng As Range
```

```
Dim riga As Long
```

```
Dim temp As Variant
```

```
' Imposta il range della tua matrice
```

```
Set rng = ActiveSheet.UsedRange
```

```
' Verifica se ci sono abbastanza colonne
```

```
If rng.Columns.Count < 3 Then
```

```
    MsgBox "La matrice deve avere almeno tre colonne.", vbExclamation
```

```
Exit Sub
```



```

End If
' Inverti le prime due colonne
For riga = 1 To rng.Rows.Count
    ' Memorizza il valore temporaneo della prima colonna
    temp = rng.Cells(riga, 1).Value
    ' Scambia i valori tra la prima e la seconda colonna
    rng.Cells(riga, 1).Value = rng.Cells(riga, 2).Value
    rng.Cells(riga, 2).Value = temp
Next riga

```

'NONA PARTE: ORDINA LE RIGHE IN BASE AI CANALI

```

' Verifica se la colonna "Channel" esiste
Dim channelCol As Range
On Error Resume Next
Set channelCol = rng.Rows(1).Find(What:="Channel", LookIn:=xlValues, LookAt:=xlWhole)
On Error GoTo 0
If channelCol Is Nothing Then
    MsgBox "La colonna con il titolo ""Channel"" non è stata trovata.", vbExclamation
    Exit Sub
End If
' Ordina le righe in base alla colonna "Channel"
With rng
    .Sort key1:=channelCol, Header:=xlYes
End With
End Sub

```

Macro 3: genera_net_3_1

```

Sub genera_net_3_1()
    Dim FoglioInput As Worksheet

```

```

Dim FoglioOutput As Worksheet
Dim DatiInput As Range
Dim DatiOutput As Range
Dim UltimaRiga As Long
Dim riga As Long
Dim Contatore As Integer
' Imposta il foglio di lavoro di input
Set FoglioInput = ThisWorkbook.Sheets("grezzi")
' Imposta il foglio di lavoro di output
Set FoglioOutput = ThisWorkbook.Sheets("n_net")
' Trova l'ultima riga nella colonna di input
UltimaRiga = FoglioInput.Cells(FoglioInput.Rows.Count, "A").End(xlUp).Row
' Imposta il range dei dati di input
Set DatiInput = FoglioInput.Range("A2:B" & UltimaRiga)
' Imposta il range dei dati di output nella colonna parallela
Set DatiOutput = FoglioOutput.Range("A1:S" & UltimaRiga)
' Inizializza il contatore
Contatore = 1
Dim foundOne As Boolean
foundOne = False
Dim connettore As Integer
connettore = 1
Dim u As Integer
u = 0
' Loop attraverso ogni riga nella colonna di input
' genero i connettori in base al canale in cui capitano con queste linee un po' complesse
For riga = 1 To UltimaRiga - 1
    If DatiInput(riga, 1).Value <= Contatore * 64 Then
        DatiOutput(riga, 2).Value = "\CONN" & connettore & "\"
        foundOne = True
    Else

```

```

    If foundOne Then
        connettore = connettore + 1
        foundOne = False
    End If
    Contatore = Contatore + 1
    riga = riga - 1
    u = 1
End If
' genero il pin del connettore corrispondente al canale su quel connettore
' correzione con la variabile u necessaria quando ci sono salti di canali e il flag non è ancora
impostato
DatiOutput(riga + u, 4) = DatiInput(riga, 1) - 64 * (Contatore - 1)
u = 0
Next riga
For riga = 1 To UltimaRiga - 1
    ' Popola la matrice di output con i valori appropriati
    DatiOutput(riga, 1).Value = "\NET" & riga & "\  "
    DatiOutput(riga, 3).Value = "-\"
    DatiOutput(riga, 5).Value = "\ \TP"
    DatiOutput(riga, 6).Value = DatiInput(riga, 2).Value
    DatiOutput(riga, 7).Value = "\-1\ "
Next riga
End Sub

```

Macro 4: genera_part_4_1

```

Sub genera_part_4_1()
    Dim FoglioInput As Worksheet
    Dim FoglioAltroInput As Worksheet
    Dim FoglioOutput As Worksheet
    Dim DatiInput As Range

```

```

Dim DatiOutput50 As Range
Dim DatiOutput75 As Range
Dim DatiOutput100 As Range
Dim UltimaRiga As Long
Dim riga As Long
' Imposta il foglio di lavoro di input
Set FoglioInput = ThisWorkbook.Sheets("grezzi")
Set FoglioAltroInput = ThisWorkbook.Sheets("n_net")
' Imposta il foglio di lavoro di output
Set FoglioOutput = ThisWorkbook.Sheets("n_part")
FoglioOutput.Cells(1, 1).Value = " "
FoglioOutput.Cells(2, 1).Value = "%Part"
UltimaRigaDest = 2

```

'GENERO LISTA CONNETTORI

```

Dim UltimaCella As Range
Dim ContenutoUltimaCella As String
Dim Numero As Integer
Dim i As Integer
Dim j As Integer
' Trova l'ultima cella nella colonna di input
Set UltimaCella = FoglioAltroInput.Cells(FoglioAltroInput.Rows.Count, "B").End(xlUp)
' Leggi il contenuto dell'ultima cella
ContenutoUltimaCella = UltimaCella.Value
' Estrai il numero dall'ultimo valore della cella
Numero = Val(Mid(ContenutoUltimaCella, InStrRev(ContenutoUltimaCella, "N") + 1))
FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = "\JUMP_32X2\  "
' Crea una riga di celle con i valori da "\CONN1\" fino a "\CONNN\"
j = 1
For i = 1 To Numero

```

```
FoglioOutput.Cells(UltimaRigaDest + 1, j + 1).Value = "\CONN" & i & "\"
```

```
j = j + 1
```

```
If j = 15 Then
```

```
    UltimaRigaDest = UltimaRigaDest + 1
```

```
    j = 1
```

```
    FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = "* "
```

```
End If
```

```
Next i
```

```
If j = 1 Then
```

```
    FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = ""
```

```
    UltimaRigaDest = UltimaRigaDest - 1
```

```
End If
```

```
UltimaRigaDest = UltimaRigaDest + 1
```

```
'GENERO LISTE TP
```

```
' Trova l'ultima riga nella colonna di input
```

```
UltimaRiga = FoglioInput.Cells(FoglioInput.Rows.Count, "A").End(xlUp).Row
```

```
' Imposta il range dei dati di input
```

```
Set DatiInput = FoglioInput.Range("B2:E" & UltimaRiga)
```

```
FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = "\TP_1.70_PF_To1_0.03\ "
```

```
Dim Contatore100 As Long
```

```
Contatore100 = 1
```

```
' Loop attraverso ogni riga nella colonna di input
```

```
For riga = 1 To UltimaRiga
```

```
    ' Se il valore di associato è 100, copia il valore numerico nella colonna 3 dell'output
```

```
    If DatiInput(riga, 4).Value = 100 Then
```

```
        FoglioOutput.Cells(UltimaRigaDest + 1, Contatore100 + 1).Value = "\TP" & DatiInput(riga, 1).Value & "\"
```

```
        Contatore100 = Contatore100 + 1
```

```
    End If
```

```

If Contatore100 = 15 Then
    UltimaRigaDest = UltimaRigaDest + 1
    Contatore100 = 1
    FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = "*"
End If
Next riga
If Contatore100 = 1 Then
    FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = ""
    UltimaRigaDest = UltimaRigaDest - 1
End If
UltimaRigaDest = UltimaRigaDest + 1
FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = "\TP_1.35_PF_Tol_0.03\ "
Dim Contatore75 As Long
Contatore75 = 1
' Loop attraverso ogni riga nella colonna di input
For riga = 1 To UltimaRiga
    ' Se il valore di associato è 75, copia il valore numerico nella colonna 3 dell'output
    If DatiInput(riga, 4).Value = 75 Then
        FoglioOutput.Cells(UltimaRigaDest + 1, Contatore75 + 1).Value = "\TP" & DatiInput(riga,
1).Value & "\ "
        Contatore75 = Contatore75 + 1
    End If
    If Contatore75 = 15 Then
        UltimaRigaDest = UltimaRigaDest + 1
        Contatore75 = 1
        FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = "*"
    End If
Next riga
If Contatore75 = 1 Then
    FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = ""
    UltimaRigaDest = UltimaRigaDest - 1

```

```

End If
UltimaRigaDest = UltimaRigaDest + 1
FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = "\TP_1.00_PF_To1_0.03\  "
Dim Contatore50 As Long
Contatore50 = 1
' Loop attraverso ogni riga nella colonna di input
For riga = 1 To UltimaRiga
    ' Se il valore di associato è 50, copia il valore numerico nella colonna 3 dell'output
    If DatiInput(riga, 4).Value = 50 Then
        FoglioOutput.Cells(UltimaRigaDest + 1, Contatore50 + 1).Value = "TP" & DatiInput(riga,
1).Value & "\"
        Contatore50 = Contatore50 + 1
    End If
    If Contatore50 = 15 Then
        UltimaRigaDest = UltimaRigaDest + 1
        Contatore50 = 1
        FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = "* "
    End If
Next riga
If Contatore50 = 1 Then
    FoglioOutput.Cells(UltimaRigaDest + 1, 1).Value = ""
    UltimaRigaDest = UltimaRigaDest - 1
End If
End Sub

```

Macro 5: genera_posizioni_3_2

```

Sub genera_posizioni_3_2()
    Dim FoglioInput As Worksheet
    Dim FoglioOutput As Worksheet
    Dim DatiInput As Range

```

```

Dim DatiOutput As Range
Dim UltimaRiga As Long
Dim riga As Long

' Imposta il foglio di lavoro di input
Set FoglioInput = ThisWorkbook.Sheets("grezzi")

' Imposta il foglio di lavoro di output
Set FoglioOutput = ThisWorkbook.Sheets("p_posizioni")

' Trova l'ultima riga nella colonna di input
UltimaRiga = FoglioInput.Cells(FoglioInput.Rows.Count, "A").End(xlUp).Row - 1

' Imposta il range dei dati di input
Set DatiInput = FoglioInput.Range("B2:E" & UltimaRiga)

' Imposta il range dei dati di output nella matrice 3 colonne
Set DatiOutput = FoglioOutput.Range("A1:R" & UltimaRiga)

' Loop attraverso ogni riga nella colonna di input
For riga = 1 To UltimaRiga
    ' Popola la matrice di output con i valori appropriati
    DatiOutput(riga, 1).Value = ""
    DatiOutput(riga, 3).Value = "" & " " & ""
    DatiOutput(riga, 5).Value = "" & " " & "" & "TP"
    DatiOutput(riga, 6).Value = DatiInput(riga, 1).Value
    DatiOutput(riga, 7).Value = "" & "\n"
    DatiOutput(riga, 8).Value = DatiInput(riga, 2).Value
    DatiOutput(riga, 9).Value = "  "
    DatiOutput(riga, 10).Value = DatiInput(riga, 3).Value
    DatiOutput(riga, 11).Value = "  "
    DatiOutput(riga, 12).Value = "0.0"
    DatiOutput(riga, 13).Value = "  "
    DatiOutput(riga, 14).Value = "0.0"
    DatiOutput(riga, 15).Value = "  "
    DatiOutput(riga, 16).Value = "TOP"
    DatiOutput(riga, 17).Value = "  "

```



```

DatiOutput(riga, 18).Value = "PLACED"
If DatiInput(riga, 4).Value = 100 Then
    DatiOutput(riga, 2).Value = "TP_1.70_PF_Tol_0.03"
    DatiOutput(riga, 4).Value = "TP_1.70_PF_Tol_0.03"
ElseIf DatiInput(riga, 4).Value = 75 Then
    DatiOutput(riga, 2).Value = "TP_1.35_PF_Tol_0.03"
    DatiOutput(riga, 4).Value = "TP_1.35_PF_Tol_0.03"
ElseIf DatiInput(riga, 4).Value = 50 Then
    DatiOutput(riga, 2).Value = "TP_1.00_PF_Tol_0.03"
    DatiOutput(riga, 4).Value = "TP_1.00_PF_Tol_0.03"
End If
Next riga
End Sub

```

Macro 6: unisci_parti_5_1

```

Sub unisci_parti_5_1()
    Dim wsDestinazione As Worksheet
    Dim wsOrigine As Worksheet
    Dim Escluso1 As Worksheet
    Dim Escluso2 As Worksheet
    Dim Escluso3 As Worksheet
    Dim Escluso4 As Worksheet
    Dim Escluso5 As Worksheet
    Dim UltimaRigaDest As Long
    Dim UltimaColonnaOrig As Long
    Dim RigaDaCopiare As Long
    Dim ColonnaDaCopiare As Long

    'COPIO TUTTI I DATI NEL FOGLIO DI DESTINAZIONE

```

```

' Imposta il foglio di destinazione
Set wsDestinazione = ThisWorkbook.Sheets("n_finale")
Set Escluso1 = ThisWorkbook.Sheets("p_posizioni")
Set Escluso2 = ThisWorkbook.Sheets("p_chiusura")
Set Escluso3 = ThisWorkbook.Sheets("p_finale")
Set Escluso4 = ThisWorkbook.Sheets("grezzi")
Set Escluso5 = ThisWorkbook.Sheets("p_intest")
wsDestinazione.Cells.ClearContents

' Inizializza l'ultima riga nel foglio di destinazione
UltimaRigaDest = wsDestinazione.Cells(wsDestinazione.Rows.Count, "A").End(xlUp).Row

' Loop attraverso tutti i fogli
For Each wsOrigine In ThisWorkbook.Worksheets
    ' Escludi il foglio di destinazione stesso
    If wsOrigine.Name <> wsDestinazione.Name And wsOrigine.Name <> Escluso1.Name And
wsOrigine.Name <> Escluso2.Name And wsOrigine.Name <> Escluso3.Name And
wsOrigine.Name <> Escluso4.Name And wsOrigine.Name <> Escluso5.Name Then
        ' Loop attraverso tutte le celle con dati nel foglio di origine
        For RigaDaCopiare = 1 To wsOrigine.UsedRange.Rows.Count
            For ColonnaDaCopiare = 1 To wsOrigine.UsedRange.Columns.Count
                ' Copia i dati nella riga successiva nel foglio di destinazione
                wsDestinazione.Cells(UltimaRigaDest + RigaDaCopiare - 1,
ColonnaDaCopiare).Value = wsOrigine.Cells(RigaDaCopiare, ColonnaDaCopiare).Value
            Next ColonnaDaCopiare
        Next RigaDaCopiare
        ' Aggiorna l'ultima riga nel foglio di destinazione
        UltimaRigaDest = wsDestinazione.Cells(wsDestinazione.Rows.Count, "A").End(xlUp).Row
+ 1
    End If
Next wsOrigine
End Sub

```

Macro 7: unisci_parti_5_2

```
Sub unisci_parti_5_2()  
    Dim wsDestinazione As Worksheet  
    Dim wsOrigine As Worksheet  
    Dim Escluso1 As Worksheet  
    Dim Escluso2 As Worksheet  
    Dim Escluso3 As Worksheet  
    Dim Escluso4 As Worksheet  
    Dim Escluso5 As Worksheet  
    Dim UltimaRigaDest As Long  
    Dim UltimaColonnaOrig As Long  
    Dim RigaDaCopiare As Long  
    Dim ColonnaDaCopiare As Long  
    ' Imposta il foglio di destinazione  
    Set wsDestinazione = ThisWorkbook.Sheets("p_finale")  
    Set Escluso1 = ThisWorkbook.Sheets("grezzi")  
    Set Escluso2 = ThisWorkbook.Sheets("n_intest")  
    Set Escluso3 = ThisWorkbook.Sheets("n_net")  
    Set Escluso4 = ThisWorkbook.Sheets("n_part")  
    Set Escluso5 = ThisWorkbook.Sheets("n_finale")  
    wsDestinazione.Cells.ClearContents  
    ' Inizializza l'ultima riga nel foglio di destinazione  
    UltimaRigaDest = wsDestinazione.Cells(wsDestinazione.Rows.Count, "A").End(xlUp).Row  
    ' Loop attraverso tutti i fogli  
    For Each wsOrigine In ThisWorkbook.Worksheets  
        ' Escludi il foglio di destinazione stesso  
        If wsOrigine.Name <> wsDestinazione.Name And wsOrigine.Name <> Escluso1.Name And  
wsOrigine.Name <> Escluso2.Name And wsOrigine.Name <> Escluso3.Name And  
wsOrigine.Name <> Escluso4.Name And wsOrigine.Name <> Escluso5.Name Then  
            ' Loop attraverso tutte le celle con dati nel foglio di origine
```

```

For RigaDaCopiare = 1 To wsOrigine.UsedRange.Rows.Count
    For ColonnaDaCopiare = 1 To wsOrigine.UsedRange.Columns.Count
        ' Copia i dati nella riga successiva nel foglio di destinazione
        wsDestinazione.Cells(UltimaRigaDest + RigaDaCopiare - 1,
ColonnaDaCopiare).Value = wsOrigine.Cells(RigaDaCopiare, ColonnaDaCopiare).Value
    Next ColonnaDaCopiare
Next RigaDaCopiare
' Aggiorna l'ultima riga nel foglio di destinazione
UltimaRigaDest = wsDestinazione.Cells(wsDestinazione.Rows.Count, "A").End(xlUp).Row
+ 1
End If
Next wsOrigine
End Sub

```