

## LibreOffice Calc BASICによるダクト静圧計算ソフトウェア 取扱説明書

### 1. はじめに

このプログラムは、既に配布している配管圧力損失計算プログラムとほぼ同じ内容のモジュールにより構成されています。  
配管圧力損失計算プログラムとの主な違いは、重力ヘッドによる負荷がないことです。ダクトの静圧計算にあたっては、重力ヘッドと浮力とがバランスするため、計算においては無視することができます。  
その他については、配管圧力損失計算プログラムのドキュメントを参照しながら、この説明書もお読みください。

### 2. 使用許諾条件

このプログラムは、General Public License (GPL)のライセンス形態で配布します。GPL については、説明を省略しますので、ウェブサイトなどで調べてください。  
このプログラムの内容を理解せずに使用すると、誤った結果を導きます。また、ソースコードが正しいことは証明されていません。バグのないソフトウェアが存在しないことは、既知の事実であり、このプログラムもその例外ではありません。  
このプログラムを使用したことによる直接・間接のいかなる損害についても保証することはできません。このプログラムは、ユーザーの責任で使用して下さい。  
このプログラムは、著作権でも保護されているため、このプログラムやこれを元に修正したプログラムを販売することを禁じます。無償で配布することは、GPL により許諾されています。  
このプログラムに関するご質問は、ウェブサイトのフォームよりご連絡ください。可能であれば対応致します。  
場合によっては、ブログでの回答になることをご了承ください。  
ユーザーがプログラムを改変した場合には、GPLによりプログラムの公開義務があります。第三者または元の開発者から公開を求められた場合には、必ず公開しなければなりません。公開を拒否すると GPL によるライセンスの違反となります。

### 3. プログラムのバグについて

このプログラムは、何例かの計算では問題のないことを確認しましたが、計算していない条件では、誤った数値が入っていないことは、確認できていません。計算した結果、何かおかしいと感じた場合には、各要素ごとに損失値を確認してください。必要に応じ、データベース内の数値やプログラムの中の定数を修正してください。入力ミスも必ず確認してください。  
データベース内の数値やプログラム中の定数を修正する前に、必ず、元のファイルをバックアップしておいてください。

### 4. プログラムに使用された数式

プログラム中に使用された数式は、全て一般的な計算式であり、特殊な数式は使用されていません。数式は、Appendix に記載されたプログラムの解説を参照してください。  
損失値は、一般的なものを使用しています。データに疑問がある場合には、datenbank シートの C 列の値を変更してください。また、英語部分に問題があればユーザーが修正して使用してください。

### 5. マクロのセキュリティ設定

配管圧力損失計算プログラムのドキュメントを参照するか、LibreOffice/OpenOffice のドキュメントを参照してください。

### Appendix プログラムの解説

| ソースコード   | 注意事項   |
|--|--|
| <pre>REM ***** BASIC *****  Dim i, j, reihen_datebank, reihe_rechnung, nom_intestazione As Integer Dim reihe_erst, num_list As Integer Dim aspekt, auflisten(400,2) As String Dim friktion(400) As Double Dim verlust, menge, laenge, durchmesser As Double Dim sostanziale, flaeche, schnellkeit, druckbewegung, flesen As Double Dim pa, mmaq, inaq As Double Dim buch, bankname, rechnung, aSheet as Object Dim sprache As String</pre> | <p>モジュール全体に亘って有効なグローバル変数を宣言する</p>  |
| <pre>Sub auffuelen Click() ' 圧力損失計算実行ボタン      Call auffuelen  End Sub</pre>  | <p>計算実行ボタン GUI ツールでボタンを作成して、実行するサブルーチンを割り当てる。最後にコメントを書き込む。ここでは、言語選択変数"sprache"に"0"を指定し</p> |

|   |  |
|---|--|
| <pre>Sub report_Click() ' レポート出力ボタン ' sprache(言語選択) 0:日本語, 1:英語   sprache=0   Call berichten End Sub</pre>  | <p>て、サブルーチン<br/>"berichten()"を実行。</p>  |
| <pre>Sub report_e_Click() ' レポート出力ボタン ' sprache(言語選択) 0:日本語, 1:英語   sprache=1   Call berichten End Sub</pre>  | <p>計算実行ボタン<br/>ここでは、言語選択変<br/>数"sprache"に"1"を指定し<br/>て、サブルーチン<br/>"berichten()"を実行。</p>  |
| <pre>Sub streichen_Click() ' 計算結果消去ボタン    Call streichen End Sub</pre>  | <p>計算実行ボタン<br/>計算結果消去サブルーチ<br/>ン"streichen"を実行する。</p>  |
| <pre>Sub auffuelen() ' 圧損計算モジュール  ' オブジェクト指定 buch=ThisComponent      ' このファイルをオブジェクトとする bankname=buch.getSheets().getByName("datenbank") ' データベースシート rechnung=buch.getSheets().getByName("rechnung") ' 計算シート aSheet=buch.CurrentController.ActiveSheet      ' アクティブシート  ' 圧損要素開始行の1行前  reihe_erst = 9          ' ヘッダー部最後の行 num_list = bankname.getCellByPosition(2,0).value ' リスト要素の最大数 ' リストの最大数は、"datenbank"シートのC1セルにユーザが記載する  ' 要素の行数を求める For i = reihe_erst+1 To 1000   If rechnung.getCellByPosition(0,i).string = "" Then     reihe_rechnung = i - reihe_erst - 1     GoTo Z10010   End If Next i Z10010:  ' 要素の行数をC1セルに書込む rechnung.getCellByPosition(2,0).value=reihe_rechnung  ' "datebank"タブ内の損失係数表を読み込みメモリ内に保存する  For i = 1 To num_list   auflisten(i,1) = bankname.getCellByPosition(1, i).string ' 日本語   auflisten(i,2) = bankname.getCellByPosition(5, i).string ' 英語訳   friktion(i) = bankname.getCellByPosition(2,i).value      ' 損失係数 Next i  ' 風量を自動計算する ' 正の数：流出、負の数：流入  For i = 2 To reihe_rechnung   rechnung.getCellByPosition(4,reihe_erst+i).value _     =rechnung.getCellByPosition(4,reihe_erst+i-1).value- rechnung.getCellByPosition(5,reihe_erst+i).value Next i  ' A列が空白になるまで、計算を継続する(最大で要素数まで) ' A列セルに空白があるとその下にデータがあっても計算しないので注意  For i = 1 To reihe_rechnung   If rechnung.getCellByPosition(0,reihe_erst+i).string = "" Then     GoTo Z100   End If End Sub  ' ダクト幅 reiheweite = rechnung.getCellByPosition(8,reihe_erst+i).value</pre> | <p>記述を簡素化するため、オブ<br/>ジェクト名を定義する。</p> <p>ヘッダーは、rechnungシートの<br/>10行目まで続き、11行目か<br/>らがユーザーの入力データと<br/>なります。<br/>BASICでは表や列は0から始ま<br/>るので、10行目=9というこ<br/>とになります。</p> <p>行数を数得る場合、最大1001<br/>行目まで数えられます。更に<br/>データを増やす場合には、<br/>For i = reihe_erst+1 To<br/>1000<br/>を "To 2000"等に変更してく<br/>ださい。</p> <p>要素名（日本語+英語）と損失<br/>係数をデータに格納します。</p> <p>rechnungシートのA列は、物<br/>理的な要素名を認識する他、<br/>データ有無を判断するのに使<br/>用されています。</p> |

```

' ダクト深さ
  tiefe = rechnung.getCellByPosition(9,reihe_erst+i).value
' 圧損要素の名称
  aspekt = rechnung.getCellByPosition(1,reihe_erst+i).string
' 円管ダクトの直径
  durchmesser = rechnung.getCellByPosition(7,reihe_erst+i).value
' 流量
  menge = rechnung.getCellByPosition(4,reihe_erst+i).value
' ダクト長
  laenge = rechnung.getCellByPosition(6,reihe_erst+i).value

' 相当直径 -durchmesser
' 円管直径が空白の場合、相当直径を計算し、K列に書込む
If rechnung.getCellByPosition(7,reihe_erst+i).string="" Then
  rechnung.getCellByPosition(10,reihe_erst+i,11).value
    =1.3*((reiheweite * tiefe)^5 / (reiheweite + tiefe)^2)^(1/8)
Else
' 円管直径の値がある場合には、K列に円管直径を書込む
  rechnung.getCellByPosition(10,reihe_erst+i).value = durchmesser
End If

' 直径は相当直径に統一し、K列の値を計算に使う
  durchmesser = rechnung.getCellByPosition(10,reihe_erst+i).value

' 断面積 -flaeche
' 円管の直径が空白の場合、矩形ダクトとして面積を計算する

' 矩形ダクトの場合
If rechnung.getCellByPosition(7,reihe_erst+i).string = "" Then
  rechnung.getCellByPosition(11,reihe_erst+i).value = reiheweite * tiefe
* 0.000001

' 円管ダクトの場合
Else
  rechnung.getCellByPosition(11,reihe_erst+i).value =
3.14/4*durchmesser^2*0.000001
End If

' L列に面積を書込む
  flaeche = rechnung.getCellByPosition(11,reihe_erst+i).value

' 流速,動圧 -schnellkeit, druckbewegung
  schnellkeit = menge / flaeche / 3600
  rechnung.getCellByPosition(12,reihe_erst+i,13).value = schnellkeit
  druckbewegung = schnellkeit ^ 2 * 1.2 / 2
  rechnung.getCellByPosition(13,reihe_erst+i,14).value = druckbewegung

' f 読取値 - flesen
' B列内の選択した要素を読み込み、メモリ内の損失係数表から損失係数を参照する
For j = 1 To num_list
  If aspekt = auflisten(j,1) Then
    flesen = friktion(j)
    ' 参照した損失係数を0列にそのまま記入する
    rechnung.getCellByPosition(14,reihe_erst+i).value = flesen
    rechnung.getCellByPosition(16,reihe_erst+i).string = auflisten(j,2)
    GoTo Z200
  End If
Next j
Z200:
' f - verlust
' 損失係数を修正する (直管のみ)
' 直管の場合は損失係数計算に、管長を含めるため、別処理にする
If flesen = 999 Then
' "datenbank"シートからの損失係数の読値で999となっている場合(直管)

```

datenbankシート（損失係数データベース）の係数には、実際の係数と明白に違うものがあります。その場合は、ユーザーが計算方法を指定することになります。

|  |  |
|--|--|
| <pre> ' 999 ではなく、計算式で損失係数を求める     verlust = laenge / durchmesser * 0.02 * 1000  ' ユーザ計算(VAV) ' メーカーカタログの図表から風量と損失との関係を数式化したもの ' この部分は、製品仕様に合わせて作成すること ElseIf fleser = 998 Then     verlust = Exp(Log(3.675)+2*Log(rechnung.getCellByPosition(4,reihe_erst+i, 5).value/bankname.getCellByPosition(4,j+1).value))     verlust = verlust / druckbewegung  ' "997"は、Pa 単位で与えた損失 (値を動圧で除している) ElseIf fleser = 997 Then     verlust = bankname.getCellByPosition(4,j).value / druckbewegung Else  ' 指定以外の場合は、損失係数はそのまま使用する     verlust = fleser End If  ' 損失係数はD列に記入する     rechnung.getCellByPosition(3,reihe_erst+i).value = verlust  ' 損失 ' 動圧値に f 値を乗じる rechnung.getCellByPosition(2,reihe_erst+i).value = verlust * druckbewegung  Next i Z100:  ' 要素毎の圧力損失値(Pa)を集計する pa = 0 For i = 1 To reihe_rechnung     pa = pa + rechnung.getCellByPosition(2,reihe_erst+i).value Next i  ' 合計圧力損失値(Pa)を書込み、Pa 値を、mmAq 単位とインチ Aq 単位に変換した値も 書込む     mmaq = pa / 9.81     inaq = mmaq / 25.4      rechnung.getCellByPosition(2,4).value = pa     rechnung.getCellByPosition(2,5).value = mmaq     rechnung.getCellByPosition(2,6).value = inaq  Msgbox "計算完了 Calculation Complete!",0,""  End Sub </pre> | <p>例：<br/> 999：直管<br/> 998：VAV（使用するモデルの<br/> カタログから圧力損失値を<br/> 読み計算式を修正してくだ<br/> さい。ここにあるモデルは一<br/> 例です。</p> |
| <pre> Sub streichen()  ' 計算結果消去モジュール ' 自分の入力には消去せず、計算結果部分のみ消去する  buch=ThisComponent rechnung=buch.getSheets().getByName("rechnung")  reihe_rechnung=rechnung.getCellByPosition(2,0).value  For i = 1 To reihe_rechnung      rechnung.getCellByPosition(2,9+i).string = ""     rechnung.getCellByPosition(3,9+i).string = ""     rechnung.getCellByPosition(4,10+i).string = ""     rechnung.getCellByPosition(10,9+i).string = ""     rechnung.getCellByPosition(11,9+i).string = ""     rechnung.getCellByPosition(12,9+i).string = ""     rechnung.getCellByPosition(13,9+i).string = ""     rechnung.getCellByPosition(14,9+i).string = ""     rechnung.getCellByPosition(16,9+i).string = ""  Next i </pre>  |  |

|   |  |
|---|--|
| <pre>rechnung.getCellByPosition(2,4).string = "" rechnung.getCellByPosition(2,5).string = "" rechnung.getCellByPosition(2,6).string = "" rechnung.getCellByPosition(2,0).string = ""  Msgbox "計算値消去完了 Computed Values Are Removed!",0,""  End Sub</pre>   |  |
| <pre>Sub berichten() ' レポート作成  Dim aufsatz As String Dim aSheet As Object Dim objController As Object Dim objSheet As Object  buch=ThisComponent bankname=buch.getSheets().getByName("datenbank") rechnung=buch.getSheets().getByName("rechnung") aSheet=buch.CurrentController.ActiveSheet  num_list = bankname.getCellByPosition(2,0).value ' リスト要素の最大数 ' リストの最大数は、"datenbank"シートの C1 セルにユーザが記載する  ' ダイアログボックスでシート名を入力 ' sprache : 日本語:0, 英語:1, Japanese:0, English:1, ボタンで自動指定 ' 言語は、プッシュボタンを押したときに自動で設定される  If sprache=0 then aufsatz = InputBox("出力するシート名を英数字で入力してください", "出力先 シート名入力", "") Else aufsatz = InputBox("SPECIFY NAME OF REPORT.", "SET OUTPUT SHEET", "") End If  ' 押されたボタンによってレポート書式を自動選択する ' レポート書式をコピーし、コピーしたレポート様式の名称を指定のものに変更す る If sprache=0 then buch.Sheets.copyByName("report_form", aufsatz, 0) Else buch.Sheets.copyByName("report_form_E", aufsatz, 0) End If  With ThisComponent objController=.getCurrentController() objSheet=.Sheets.getByNamed(aufsatz) objController.setActiveSheet(objSheet) End With  ' コピーする行数 ' reihe_erst : "rechnung"シートのヘッダの最後の行数 (ゼロ起算) ' reihe_rechnung : 計算ステップの総数 ' nom_intestazione : レポートシートのヘッダの最後の行数 (ゼロ起算)  reihe_erst = 9 ' 計算シートの最初の行計算シート書式を変更したら変 更する nom_intestazione = 6 ' レポート書式の最初の行 レポート書式を変更したら変 更する reihe_rechnung = rechnung.getCellByPosition(2,0).value ' 計算結果の行数  ' プロジェクト名と系統名をコピー objSheet.getCellByPosition(2,2).string = rechnung.getCellByPosition(1,7).string objSheet.getCellByPosition(2,3).string = rechnung.getCellByPosition(1,8).string  ' 余裕係数をコピー objSheet.getCellByPosition(11,4).value = rechnung.getCellByPosition(2,3).value  ' 結果コピー</pre> |  |

```

' 合計損失値を書込む
objSheet.getCellByPosition(2,4).value _
= rechnung.getCellByPosition(2,4).value
*rechnung.getCellByPosition(2,3).value 'Pa
objSheet.getCellByPosition(4,4).value _
= rechnung.getCellByPosition(2,5).value
*rechnung.getCellByPosition(2,3).value 'mmAq
objSheet.getCellByPosition(6,4).value _
= rechnung.getCellByPosition(2,6).value
*rechnung.getCellByPosition(2,3).value 'inAq

' 計算中の詳細数字を書込む
For i=1 to reihe_rechnung
' 番号(0)
objSheet.getCellByPosition(0,nom_intestazione+i).value= _
rechnung.getCellByPosition(0,reihe_erst+i).value
' ダクト横寸(2)
If rechnung.getCellByPosition(8,reihe_erst+i).string="" then
objSheet.getCellByPosition(2,nom_intestazione+i).string= "-"
Else
objSheet.getCellByPosition(2,nom_intestazione+i).value= _
rechnung.getCellByPosition(8,reihe_erst+i).value
End If
' ダクト縦寸(3)
If rechnung.getCellByPosition(9,reihe_erst+i).string="" then
objSheet.getCellByPosition(3,nom_intestazione+i).string= "-"
Else
objSheet.getCellByPosition(3,nom_intestazione+i).value= _
rechnung.getCellByPosition(9,reihe_erst+i).value
End If
' 相当径(4)
objSheet.getCellByPosition(4,nom_intestazione+i).value= _
rechnung.getCellByPosition(10,reihe_erst+i).value
' 長さ(5)
If rechnung.getCellByPosition(6,reihe_erst+i).value<1E-6 then
objSheet.getCellByPosition(5,nom_intestazione+i).string= "-"
Else
objSheet.getCellByPosition(5,nom_intestazione+i).value= _
rechnung.getCellByPosition(6,reihe_erst+i).value
End If
' 損失係数(6)
objSheet.getCellByPosition(6,nom_intestazione+i).value= _
rechnung.getCellByPosition(3,reihe_erst+i).value
' 風量(7)
objSheet.getCellByPosition(7,nom_intestazione+i).value= _
rechnung.getCellByPosition(4,reihe_erst+i).value
' 風速(8)
objSheet.getCellByPosition(8,nom_intestazione+i).value= _
rechnung.getCellByPosition(12,reihe_erst+i).value
' 動圧(9)
objSheet.getCellByPosition(9,nom_intestazione+i).value= _
rechnung.getCellByPosition(13,reihe_erst+i).value
' 部分損失(10)
objSheet.getCellByPosition(10,nom_intestazione+i).value= _
rechnung.getCellByPosition(2,reihe_erst+i).value

' 要素名(1) 英語対応時には要修正
If sprache=0 then
objSheet.getCellByPosition(1,nom_intestazione+i).string= _
rechnung.getCellByPosition(1,reihe_erst+i).string
Else
objSheet.getCellByPosition(1,nom_intestazione+i).string= _
rechnung.getCellByPosition(16,reihe_erst+i).string
End If

Next i

' 損失値積算計算
' 最初の損失累積値は最初の部分損失値となる
objSheet.getCellByPosition(11,nom_intestazione+1).value = _
objSheet.getCellByPosition(10,nom_intestazione+1).value

' A列が空白の場合には、累積値計算を打切る
For i = 2 To reihe_rechnung
If objSheet.getCellByPosition(0,nom_intestazione+i).string = "" Then
GoTo Z1110

```

|  |  |
|--|--|
| <pre> End If ' 前回値と今回値を加算する objSheet.getCellByPosition(11,nom_intestazione+i).value = _ objSheet.getCellByPosition(11,nom_intestazione+i-1).value + _ objSheet.getCellByPosition(10,nom_intestazione+i).value  Next i  Z1110: ' 記号がなくそれ以下にデータがないと認められる場合 Msgbox "報告書出力完了 Report Complete!",0,""  End Sub </pre>                |  |
| <pre> Sub del_reports() ' レポートを削除する Dim delSheet As Object Dim sheetname as String Dim objController As Object  buch=ThisComponent  delSheet=buch.getSheets().getByIndex(0) ' buch.Sheets.copyByName("report_form",aufsatz,0) If delSheet=buch.getSheets().getByName("rechnung") then Goto Z9990 End If  Z9990: End Sub </pre> |  |