TRNSYS18

TypeStudioで作るTRNSYSコンポーネント

このドキュメントは以下のライセンスで提供されます。

[クリエイティブ・コモンズ・ライセンス](http://creativecommons.org/licenses/by-nc-sa/4.0/)

quattro corporate design Co., Ltd. 作『TRNSYS.JP Library』は[クリエイティブ・コモンズ 表示 - 非営利 - 継承 4.0 国際 ライセンス](http://creativecommons.org/licenses/by-nc-sa/4.0/)で提供されています。

内容

[1 新しいプロジェクトを作成する 4](#_Toc333867115)

[2 気象データ 5](#_Toc333867116)

[2.1 コンポーネントの配置 5](#_Toc333867117)

[2.2 気象データの設定 6](#_Toc333867118)

[2.3 コネクションを作成する 6](#_Toc333867119)

[2.4 コネクションの設定 7](#_Toc333867120)

[2.5 動作確認 8](#_Toc333867121)

[2.6 日射量の計算 9](#_Toc333867122)

[2.6.1 方位数の設定 9](#_Toc333867123)

[2.6.2 方位の設定 9](#_Toc333867124)

[2.7 方位別日射量を確認する 10](#_Toc333867125)

このドキュメントではTRNSYS、TypeStudioの基本的な操作方法を説明しています。

以下の環境を前提として記載しています。

TRNSYS 18.00.0019(64bit)

OS Windows10 Pro(64bit, 1803)

# はじめに

TRNSYSには標準的なコンポーネントが多数用意されています。基本的なシミュレーションはそれだけでも十分に行えますが、特殊な機器や、研究開発中の装置などを組み入れたい場合にはオリジナルのコンポーネントが必要になることがあります。

従来コンポーネントの開発にはIntel FORTRANを使用が前提になっていました。Intel FORTRANは本格的なプログラミングが可能な非常に強力な開発ツールです。開発者向けに様々な機能が提供され、詳細な設定が可能です。しかし一方では気軽にプログラムを書いて試してみたいという用途には少々ハードルが高いツールであるとも言えます。

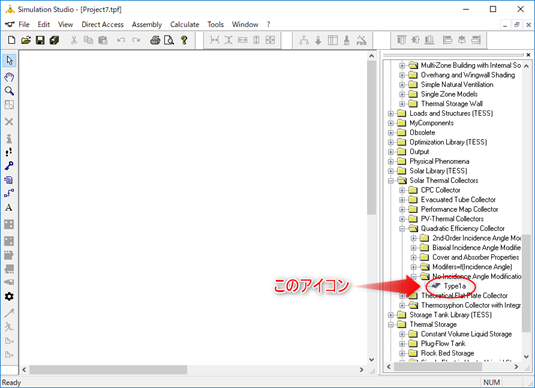
* 1. TypeStudio登場！

TRNSYS18では、コンポーネントの開発用の専用ツール、TypeStudioがリリースされました。こちらはTRNSYS専用のFORTRANコンパイラとエディタで構成されています。TRNSYSコンポーネントの開発に特化しているため、特に面倒な設定も不要で気軽にコンポーネントの開発を始めることができます。

**[](file:///C:\Users\yasuda\AppData\Local\Temp\OpenLiveWriter1641855762\supfiles10EA2100\image8.png)**

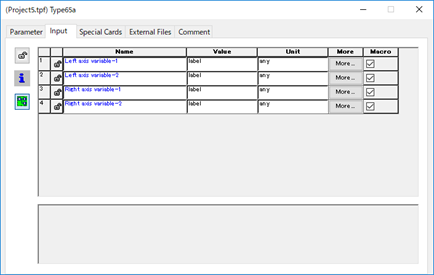
# Proforma（プロフォルマ）の作成

さて、コンポーネントを作成する前に**Proforma**を用意します。普段はあまり意識しないProformaですが、Simulation Studioでコンポーネントとして見ているアイコンの実体が**Proforma**です。



* 1. そもそもProformaとは？

Simulation Studioでコンポーネントの設定を行う際に図のような画面（Variables Window）が表示されます。この画面に表示される内容を定義しているのが**Proforma**ファイル（拡張子.TMF）です。



TRNSYSのコンポーネントの実体はFORTRANのサブルーチン（C言語だと関数）に相当します。

ProformaはSimulation Studioとサブルーチンのデータ（Parameters,Inputs,Outputs)の橋渡しルールを定義しています。

* 1. 計算式の準備

まずは、これから作成するコンポーネントの式を用意します。今回はドキュメントに載っているシンプルなヒーターの式を例にProformaを作成します。

[image](file:///C:\Users\yasuda\AppData\Local\Temp\OpenLiveWriter1641855762\supfiles10F2E2D2\image4.png)

Tset  設定温度  
Tin    入口温度  
ｍ      流量  
Cp     比熱  
Q      負荷

**※TRNSYS18, Vol.7 Developer’s / Programmer’s Guide,  p7-17, Eq.7.3.1-1 より**

さて、この式を元にコンポーネントのProformaを作っていくわけですが、はじめにそれぞれの値を**Prameter,Input,Output**に割り当てます。

**Q**は計算結果なのでOutputで問題なさそうです。**ｍ, Tset, Tin**は時系列で変化する値として**Input**に割り当てます。**Cp**は時系列で変化するなら**Input**で、変化しないのであれば固定値で**Paramter**とします。今回は変化しないものとして**Parameter**で扱うことにします。

値の割り当てを整理すると次の表のようになります。

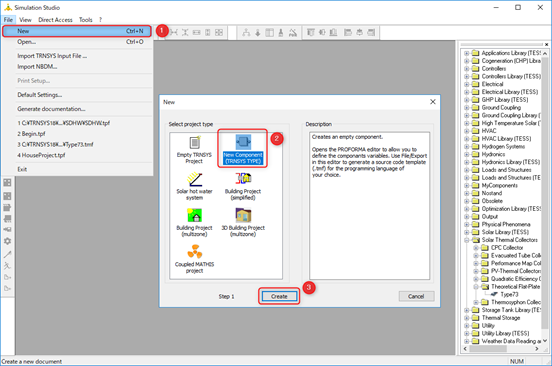
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Role | Dimension | Unit | Def. |
| m | input | Flow Rate | kg/hr | 100.0 |
| Tset | input | Temperature | C | 50.0 |
| Tin | input | Temperature | C | 20.0 |
| Cp | parameter | Specific Heat | kJ/kgK | 4.190 |
| Q | output | Power | kJ/hr | 0.0 |

※デフォルト値（Def.）はInputでは初期値として使われるので、適切は値に設定します。Cpはここでは水の比熱です。

* 1. Proformaの新規作成

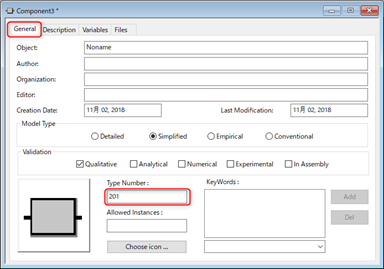
Simulation StudioでProformaを作成します。

1. メニューから**[File]-[New]**を選択する。
2. 表示されるウィンドウで**「New Component」**を選択して、
3. **Createボタン**をクリックする。

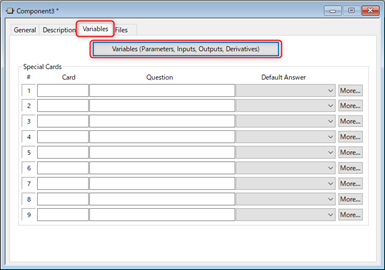
[](file:///C:\Users\yasuda\AppData\Local\Temp\OpenLiveWriter1641855762\supfiles10F2E2D2\image9.png)

Proformaのウィンドウが表示されるので、式に合せて設定を行います。

はじめに「**General**」タブのType Numberの項目にType番号を指定します。ここでは**201**を指定してください。

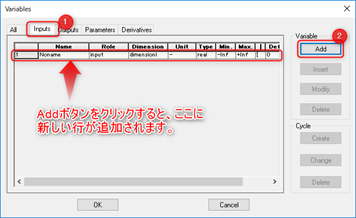


次に、「**Variable**」タブで、「**Variables(Parameters, Inputs, Outputs, Derivatives)**」ボタンをクリックして、設定画面を表示します。

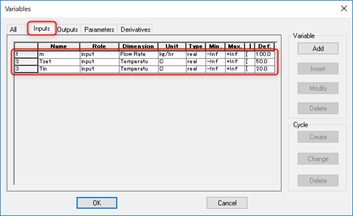


Inputの項目を追加します。

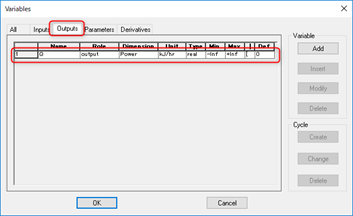
1. 「**Inputs**」タブを選択して
2. 画面右のAddボタンをクリックして新しい項目を追加します。



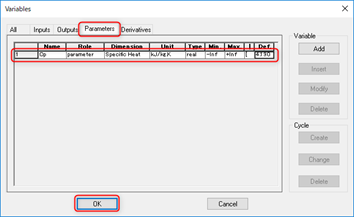
先ほどの表に合せて**Inputs**の項目を3つ追加、設定します。



つづいて、同じように「**Outputs**」タブで項目を一つ追加、設定します。



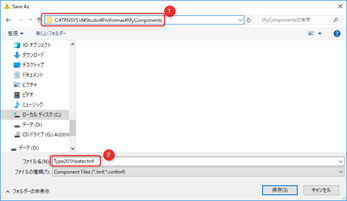
そして最後に「**Parameters**」タブで項目を一つ追加、設定したらOKボタンをクリックしてウィンドウを閉じます。



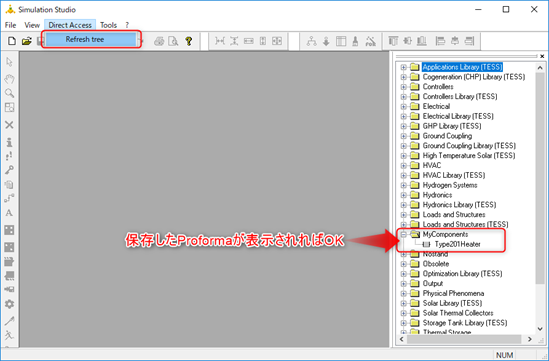
メニューから**[File]-[Save]**を選択してProformaを保存します。

① 保存先として新しいフォルダ”C:\TRNSYS18\Studio\Proformas\**MyComponents** “を作成します。

② Proformaを**Type201Heater.tmf**という名前で保存します。

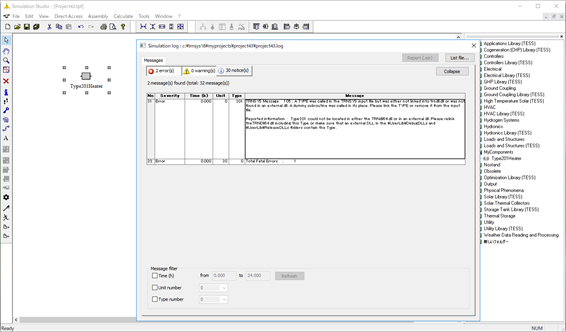


Simulation Studioのメニューから[**Direct Access**]-[**Refresh tree**]を選んで、ツールバーにType201Heaterが表示されればProformaの完成です。



この段階で、すでにコンポーネントとして配置、他のコンポーネントと接続できる状態です。

ただし、実行すると図のようなエラーが発生します。Proformaは出来上がったものの、コンポーネントの本体、つまり計算処理の部分が作成されていないためです。



# 計算実装

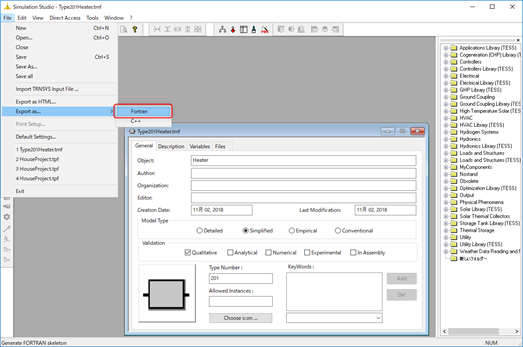
* 1. ソースコードのエクスポートと編集

**前頁で作成したProforma**から**ソースコードの雛形**をエクスポートします。

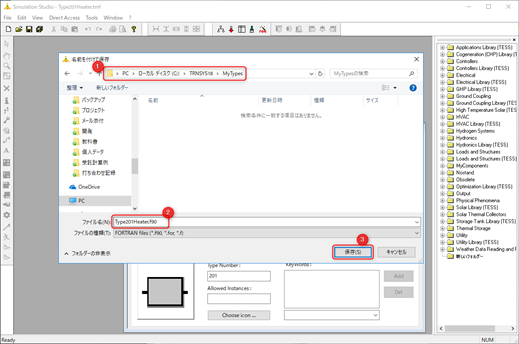
はじめに、Simulation Studioを起動してProformaを開きます。

（"C:\TRNSYS18\Studio\Proformas\MyComponents\Type201Heater.tmf"）

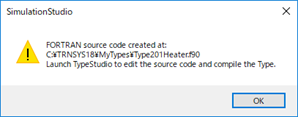
次にメニューから [**File**]-[**Export** **as**]-[**Fortran**] を選んでソースコードをエクスポートします。



1. エクスポート先のフォルダとして"**C:\TRNSYS18\MyTypes**"を選択。
2. ファイル名、"**Type201Heater.f90**"を指定。
3. 「保存」ボタンをクリックします。



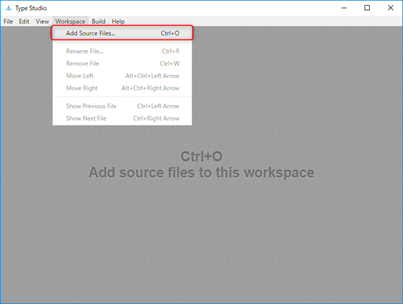
エクスポートが終わると**TypeStudio**で編集を促す、図のようなメッセージが表示されます。



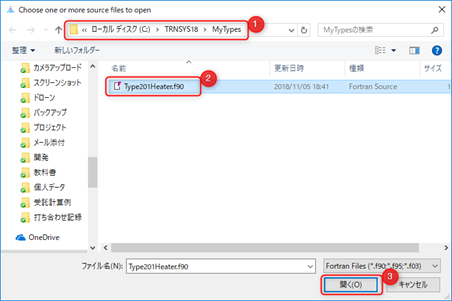
* 1. TypeStudioの起動とソースコードのインポート

**スタートメニュー**から**TypeStudio**を起動して、エクスポートしたソースコードを読み込みます。

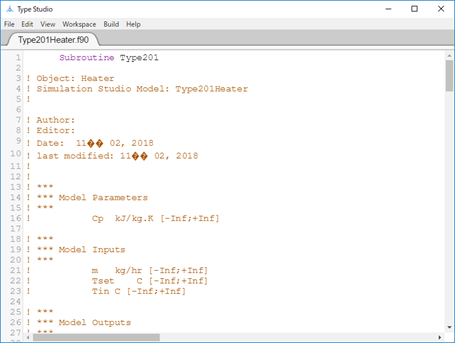
TypeStudioのメニューから[**Workspace**]-[**Add Source Files**]を選択して、



1. 先ほどソースコードをエクスポートしたフォルダ "**C:\TRNSYS18\MyTypes**"で、
2. "**Type201Heater.f90**"を選択して、
3. 「開く」ボタンのクリックしてソースコードを読み込みます。

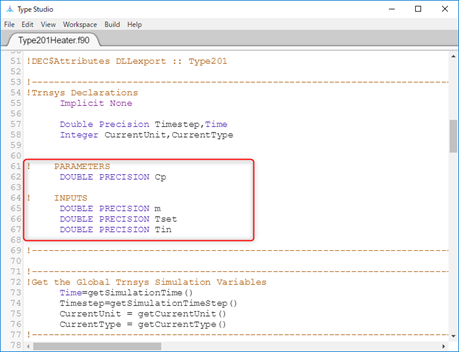


TypeStudioの画面が図のように表示されていれば準備は完了です。



* 1. ソースコードの確認

ソースコードの中身を確認すると、61行目から67行目にProformaの**Parameters**, **Inputs**の値に相当する変数が宣言されている事が確認できます。Proformaの情報が反映されていることが分かります。



* 1. 処理の実装

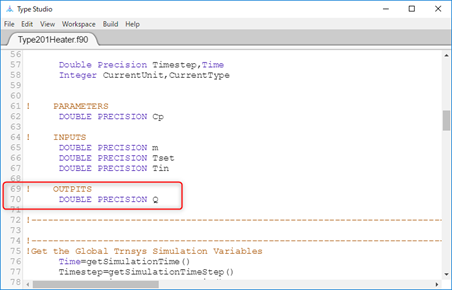
雛形のソースコードへ計算処理や出力を実装します。

* + 1. 変数の追加

前回用意した式の"**Q**"に対応する変数を追加します。

[image](file:///C:\Users\yasuda\AppData\Local\Temp\OpenLiveWriter1641855762\supfiles11061C06\image83.png)

図の赤枠のようにParameters, Inputsの変数宣言の後へ変数"**Q**"を追加します。



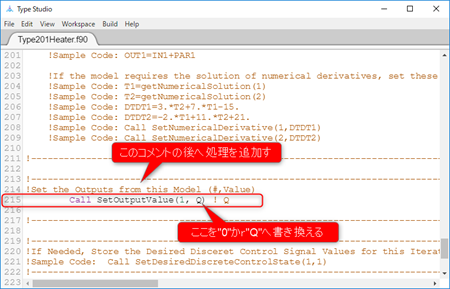
* + 1. 計算処理の追加

コンポーネントのソースコードでは処理を追加する位置が決められています。ソースコードのコメントで"**PERFORM ALL THE CALCULATION HERE FOR THIS MODEL.** "と書かれた位置に処理を追加します。この例では**180行付近**にコメントがあるので、前回準備した計算式に沿って、図のように処理を追加してください。



* + 1. 出力処理の追加

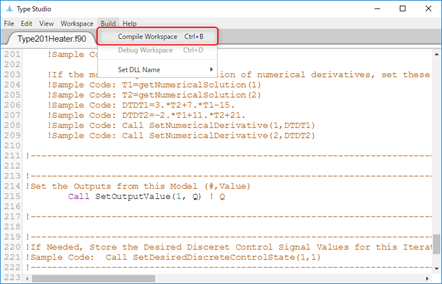
最後に、計算した値を**Outputs**へ出力処理を追加します。これも追加する位置が決まっています。ソースコードのコメント"**Set the Outputs from this Model (#,Value)**"を探して、その後に出力処理を記述します。  
この例では**214行目付近**にコメントが見つかります。すでに出力処理の雛形が用意されているので、出力する値の部分のみを図のように書き換えます。



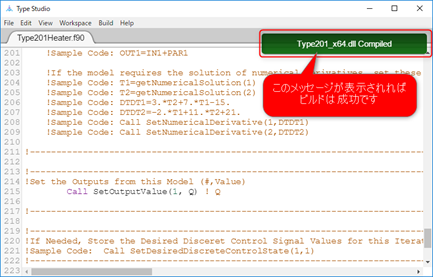
* + 1. ビルド

ソースコード修正が終わったら、いよいよビルドです。（ソースコードからコンポーネントの実体を生成します）

メニューから[Build]-[Compile Workspace]を選択します。



下の図のように画面右上に緑色のメッセージが表示されればビルドは成功です。もし違うメッセージが表示されていたら、どこかに間違いがあります。ソースコードを見直して修正、再度ビルドを行ってください。



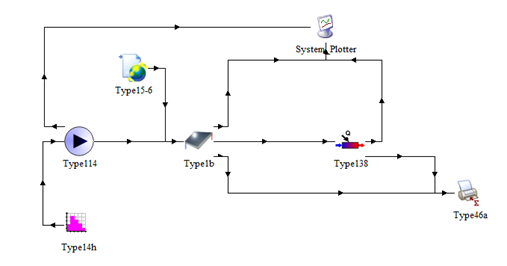
ビルドが成功したらコンポーネントが利用できる状態です。

# コンポーネントの動作検証

ビルドしたコンポーネントの動作を検証します。

* 1. プロジェクトを用意する

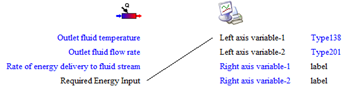
テスト用にプロジェクトを用意します。今回はExamplesフォルダのBegin.tpf（"C:\TRNSYS18\Examples\Begin\Begin.tpf"）を利用してテスト用のプロジェクトを作成します。



コンポーネントを整理して、前回作成したコンポーネント（Type201Heater)を図のように配置、接続します。



**このプロジェクトで使用されているType138**では作成したヒーターと**同じ計算**を行っています。計算結果を比較するため**Type138**から**Required Energy Input**の値をType65へ出力します。

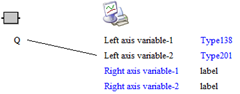


* 1. Type201Heaterの接続

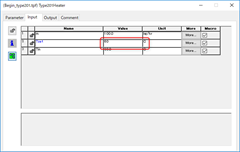
**Input**は集熱パネルから出口温度（Outlet temperature）、流量(Outlet flowrate)の値をそれぞれ対応する**Tin,m**へ接続します。



そして**Output**はType65へ負荷（**Q**)を接続しておきます。

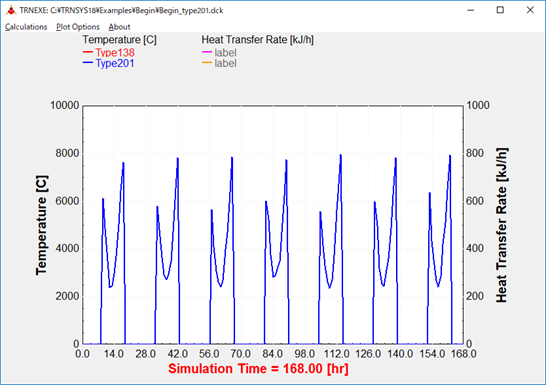


最後に、配置されているType138の設定温度（Setpoint temperature）に合せて、Type201Heaterの設定温度（**Tset**）を**60℃**へ変更します。



* 1. 結果を確認する

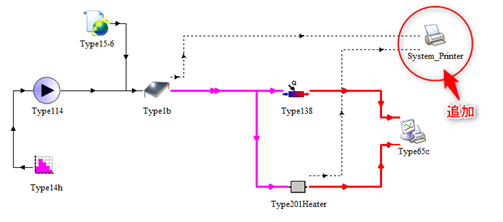
計算を実行して計算結果の確認を行います。



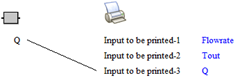
図のようにType138とType201の**結果がまったく同じ**になるため、グラフが重なって描かれています。少々分かりにくい画面になっていますが、コンポーネントは正しく動作しているようです。  
（注：画面はType65/Prameters/Left axis maximumを**10000**へ変更しています）

* 1. 計算結果の検証

この例では同じ計算を行うコンポーネントを使って比較検証しました。同じ計算を行うコンポーネントは通常は存在しないのでなんらかの方法で結果を検証する必要があります。  
ここでは関連するデータを書き出してExcelで検証する方法をご紹介します。まずは検証用のデータを書き出すためType25を配置して、図のように接続します。

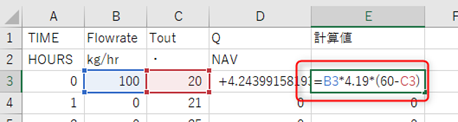


コンポーネントが計算に使用するため、Inputsで受け取っているパネルの出口温度、流量、それとコンポーネントの計算結果をType25で出力します。



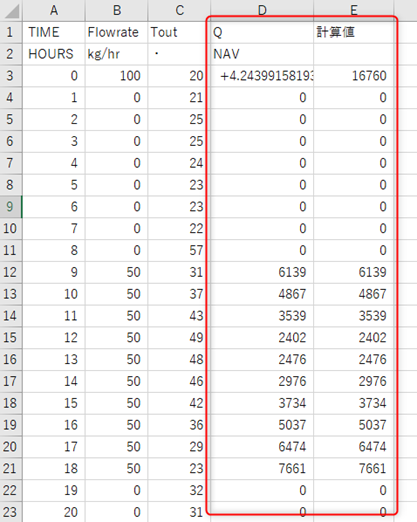
Type25de書き出された値を使ってExcelで計算して同じ結果になるか比較することで処理内容を検証します。

ファイルをExcelで開いたらコンポーネントが内部で行っているのと同じ計算式をセルに入力します。



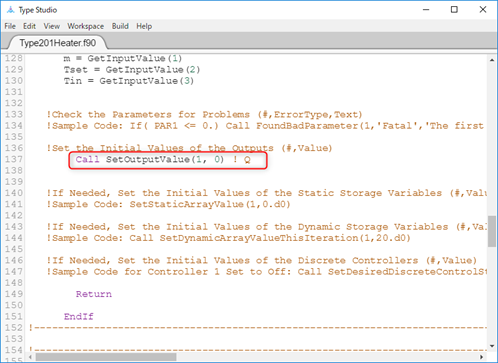
注：上の図では、0hourのQの値が変な値になっていますが、これは出力値の初期化が原因です。理由と対策は後述します。

**Qの値**と**Excelで計算した値**を比べると、同じ値になっている事が確認できます。



# Outputの初期化について

さて、0hourでおかしな値が出力されていた件ですが、これはOutputの初期化が原因です。コンポーネントのソースコードを再確認すると Output の初期化を行っている箇所があります。（下図）

[](file:///C:\Users\yasuda\AppData\Local\Temp\OpenLiveWriter1641855762\supfiles111A5D82\image103.png)

ここで、Outputの1番目、つまり**Q**の値を”**0**”に初期化しています。一見”0”で問題ないように見えますが、この書き方はFORTRANでは整数値として扱われます。やや専門的な話になりますが、整数値から実数値へのキャストによる誤差が発生しています。  
さきほどの図では値の部分が見切れていていますが、"+4.2439915819305446-314"（指数表記）で書き出されています。値としては限りなく"0"に近いのですが、計算上はこういうケースが発生します。

対策はごくシンプルで初期化の値を"**0**"から"**0.0**"へ書き換えて**実数**としてFORTRANが認識できるようにすれば解決できます。

[image](file:///C:\Users\yasuda\AppData\Local\Temp\OpenLiveWriter1641855762\supfiles111A5D82\image108.png)