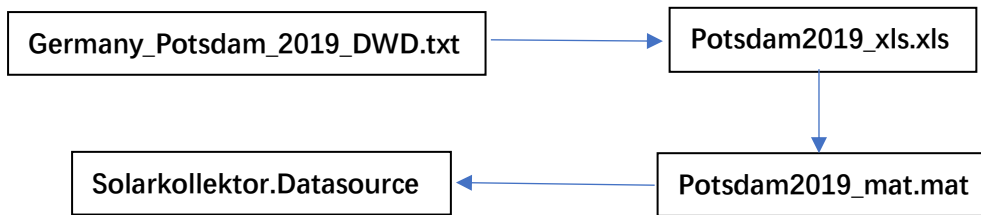


Contents:

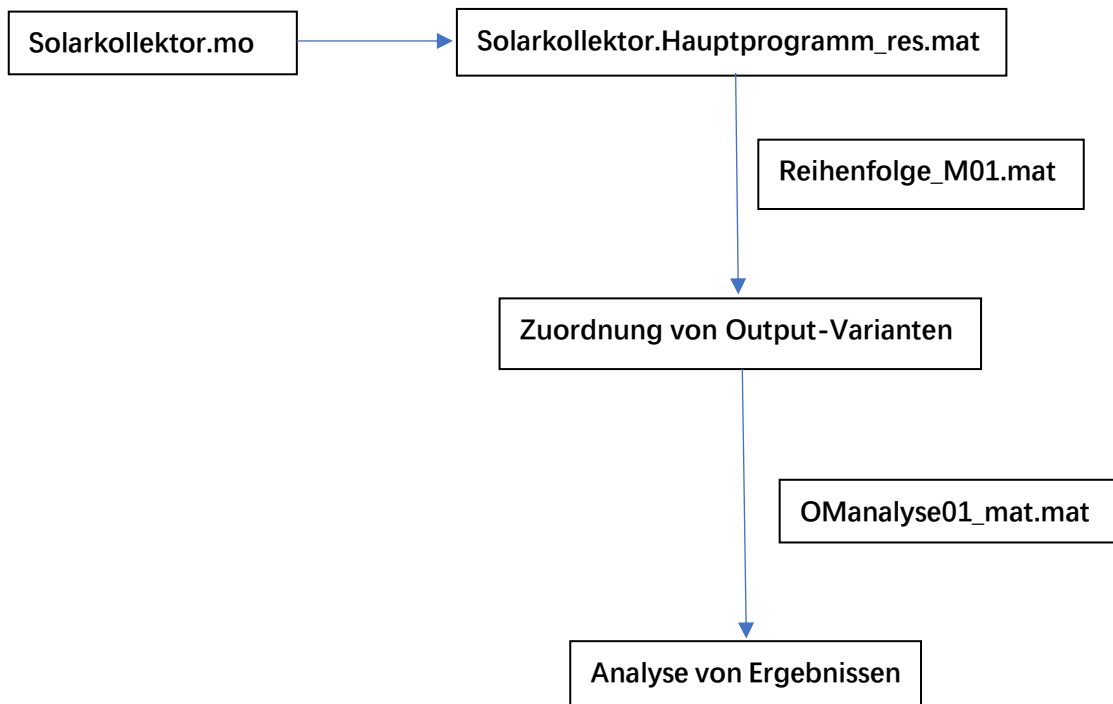
- Struktur von Arbeitspaket01
 - * Weatherdateneinlesen
 - * Darstellung und Analyse von Ergebnisse
- Kollektormodell in openmodelica erstellen
 - * Icon und Model einstellen
 - * Model von Solarkollektor einschreiben und Eingabeswerten importieren
 - * Model von Weatherdaten einschreiben und Dateneinlesen
 - * Hauptprogramme einschreiben und verbinden alle Komponenten
- **Um die Simulation von 'Solarkolektor.mo' zu laufen**
 - * **Herunterladen und Speichern**
 - * **Arbeitsverzeichnis einzustellen**
 - * **Simulation zu laufen**
- Ergebnisse von Kollektormodell als mat-file output
- Ergebnisse von openmodelica in Matlab zu analysieren
 - * Reihenfolge und Name von Output-Varianten zu analysieren
 - * Ergebnisse von openmodelica einlesen
 - * Dateneingabe
 - * Zuordnung der Ergebnisse
 - * Solarertrag-Bildung von Monatswerten

- Struktur von Arbeitspaket01

- Weatherdateneinlesen



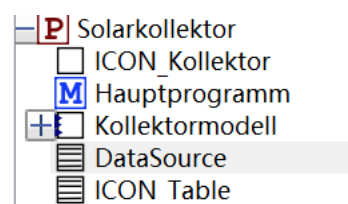
- Analyse und Darstellung von Ergebnissen



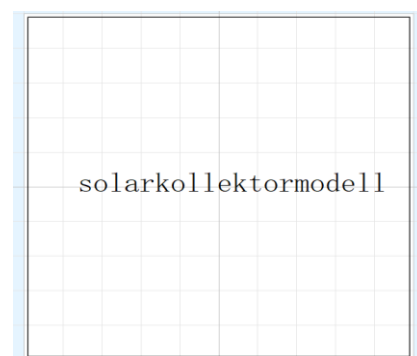
Kollektormodell in openmodelica einstellen

■ Icon und Model einstellen

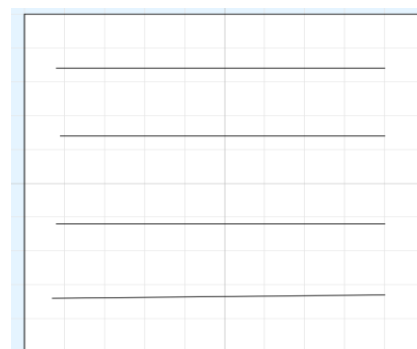
Zuerst ICON von Solarkollektormodell und Weatherdaten erstellen in Package Solarkollektor. ICON von Solarkollektormodell ist als 'ICON_Kollektor' und ICON von Weatherdaten ist als 'ICON_Table' bezeichnet. Danach das Model von Solarkollektor heißt 'Kollektormodell' und das Model von Weatherdaten heißt 'DataSouce' einstellen. Zuletzt das Model von Hauptprogramm einstellen.



ICON von Solarkollektormodell und Weatherdaten sind sowie folgendes:

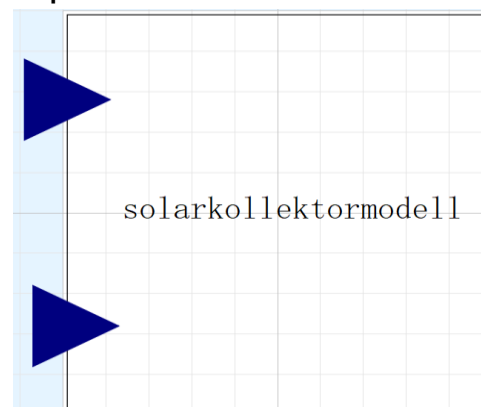


ICON von Solarkollektormodell



ICON von Weatherdaten

Bei Kollektormodell: Kuppeln das Kollektormodell mit Block 'Input' .
'Input' sind diese blaue Dreiecke. Diese 'Input' sind 'Gt', 'Ta'.



■ Model von Solarkollektor einschreiben und verbinden mit Icon

Zuerst verbinden das Model mit ICON, danach beschreiben das model von Solarkollektor. Die Code für die Beschreibung ist sowie folgendes:

```
model Kollektormodell
  extends ICON_Kollektor; "Verbindung von model und ICON"

"Definition der Kollektorparameter 'Types', 'parameters', 'Variables'"
//Types
type HeatLossCoefficient = Real(unit = "w/(m2*K)");
type Temperature = Real(unit = "deg");
type HeatCapacity = Real(unit = "J/(m2*K)");
type area = Real(unit = "m2");
type MassFlow = Real(unit = "kg/h");
//parameters
parameter area A_g = 2.0 "gross area";
parameter Real eta = 0.8 "top efficiency";
parameter MassFlow m_dot = 75.0 "mass flow";
parameter HeatLossCoefficient a_1 = 3.5;
parameter HeatLossCoefficient a_2 = 0.01;
parameter HeatCapacity a_5 = 5000;
parameter Temperature T_ci = 50 "collector entry
temperatur";
//Varialbes
Real Tco "collector outlet temperature";
Real Tm "middle temperature";
Real cp "specifichheat";
Real Qabs "absorbs amout of heat";
Real Quse "using energy";
Real Qvp "Lost Energy";
Real Qv "lost energy";

"Verbindung das Kollektormodell mit Inputsblock von Eingabewerten von , 'GT', 'Ta'"
Modelica.Blocks.Interfaces.RealInput Gt annotation(
  Placement(visible = true, transformation(origin = {-
98, 56}, extent = {{-20, -20}, {20, 20}}, rotation = 0),
  iconTransformation(origin = {-98, 56}, extent = {{-20, -
20}, {20, 20}}, rotation = 0)));
Modelica.Blocks.Interfaces.RealInput ta annotation(
  Placement(visible = true, transformation(origin = {-
94, -56}, extent = {{-20, -20}, {20, 20}}, rotation = 0),
```

```
iconTransformation(origin = {-94, -56}, extent = {{-20, -20}, {20, 20}}, rotation = 0)));
```

"Gleichungen beschreiben"

equation

```
Qabs = eta*Gt*A_g;
Qvp = (a_1*(Tm - ta) + a_2*(Tm - ta)*(Tm - ta))*A_g;
Quse = m_dot* cp*(Tco - T_ci)/3.6;
der(Tm) = (Qabs - Qvp - Quse) / (A_g * a_5);
Tm = (T_ci + Tco) / 2;
cp = 4.21091532 - 1.85646462e-3 * Tm + 3.09300765e-5 *
Tm * Tm - 1.22631409e-7 * Tm * Tm * Tm;
Qv=-Qvp;
end Kollektormodell;
```

■ Model von Weatherdaten einschreiben und Dateneinlesen

Zuerst Stellen eine DataTable in das Ansichtschema von Model der Datasoure



Nennen wir die Filename von DataTable als ,file' , und die TableName ist als ,num' genannt.

Component		
名称:	DataTable	
Class		
路径:	Modelica.Blocks.Sources.CombiTimeTable	
注释:	Table look-up with respect to time and linear/periodic extrapolation methods (data from matrix/file)	
Table data definition		
tableOnFile	<input checked="" type="checkbox"/> true	= true, if table is defined on file or in function use
table	fill(0.0, 0, 2)	Table matrix (time = first column; e.g., table=[0, 0;
tableName	"num"	Table name on file or in function usertab (see docu)
fileName	file	File where matrix is stored
verboseRead	<input checked="" type="checkbox"/> true	= true, if info message that file is loading is to be

,file' bezeichnet die Weatherdatenquelle in Aktendeckel von Openmodelica Arbeitsverzeichnis 'Modell01_v15'. Diese Datenquelle heißt ,Potsdam2019_mat.mat' .

Weatherdatenquelle :

,Germany_Potsdam_2019_DWD.txt' → ,Potsdam2019_xls.xlsx' → ,Potsdam2019_mat.mat'

,Germany_Potsdam_2019_DWD.txt' bleibt in die Bibliothek von OpenModelica . Es zeigt die Weathersituation von Potsdam im Jahr 2019.

此电脑 > 本地磁盘 (D:) > Aufgabe > lib > omilibrary > BuildingSystems 2.0.0-beta > Climate > weather

	名称	修改日期	类型	大小
_v14	Germany_Potsdam_2014_DWD	2020/11/17 8:18	文本文件	390 KB
!_v14	Germany_Potsdam_2015_DWD.nc	2020/11/17 8:18	NC 文件	619 KB
ktor.Hauptproc	Germany_Potsdam_2015_DWD	2020/11/17 8:18	文本文件	390 KB
	Germany_Potsdam_2016_DWD.nc	2020/11/17 8:18	NC 文件	621 KB
	Germany_Potsdam_2016_DWD	2020/11/17 8:18	文本文件	390 KB
	Germany_Potsdam_2017_DWD.nc	2020/11/17 8:18	NC 文件	619 KB
	Germany_Potsdam_2017_DWD	2020/11/17 8:18	文本文件	390 KB
	Germany_Potsdam_2018_DWD.nc	2020/11/17 8:18	NC 文件	619 KB
	Germany_Potsdam_2018_DWD	2020/11/17 8:18	文本文件	390 KB
	Germany_Potsdam_2019_DWD.dat	2020/11/17 8:18	DAT 文件	372 KB
	Germany_Potsdam_2019_DWD.nc	2020/11/17 8:18	NC 文件	310 KB
	Germany_Potsdam_2019_DWD	2020/11/17 8:18	文本文件	501 KB

‚Potsdam2019_xls.xlsx‘ ist von ‚Germany_Potsdam_2019_DWD.txt‘ umgewandelt.

In die erste Spalte von dieser Tabelle bezeichnet die Zeit in Uhr. Die zweite Spalte zeigt den Umgebungsdruck. Die tertiäre Spalte bezeichnet die äußere Temperatur oder Lufttemperatur. Die vierte Spalte zeigt die relative Luftfeuchtigkeit. Die fünfte Spalte bezeichnet die Windrichtung. Die sechste Spalte zeigt die Windgeschwindigkeit. Die siebte Spalte zeigt die Wolkendeckekoeffizient.

Die letzte Spalte zeigt die Globalstrahlung auf geneigte Fläche. GT kommt nicht aus ‚Potsdam2019_xls.xlsx‘, sondern aus die Ergebnissedaten von GT in Modell02.

1	Germany_Potsdam_2019_DWD_3987_DWD							
2	13.06	52.38	81	1				
3								
4	hy	p	Ta	RH	N	FF	DD	GT
5	1	1013.7	7.8	92	8	6.3	260	0
6	2	1012.8	7.9	90	8	7.1	260	0
7	3	1012.4	7.7	90	8	6.5	260	0
8	4	1011	7.4	90	8	6.8	260	0
9	5	1009.7	7.3	88	8	7.8	260	0
10	6	1008.5	7.1	89	8	8.2	260	0
11	7	1007.5	7.1	85	8	8.3	260	0
12	8	1006.7	6.8	87	8	8.6	260	0.462229
13	9	1005.8	6.6	92	8	8.4	250	16.04988
14	10	1005.4	6.6	95	8	8.9	260	27.34366
15	11	1004.7	6.6	96	8	9.1	260	42.68247
16	12	1003.9	7.4	92	7	9.1	270	307.545
17	13	1003.3	7.7	70	2	10.2	280	333.7595
18	14	1002.7	7.5	66	2	10.4	290	155.4903
19	15	1001.6	7.4	71	7	10.2	270	65.37214
20	16	1001.7	5.7	82	7	10.8	280	1.555436
21	17	1001.7	5.3	80	7	11.3	280	0
22	18	1001.4	3.7	88	5	10.8	280	0

‚Potsdam2019_mat.mat‘ ist von ‚Potsdam2019_xls.xlsx‘ umgewandelt.

Wir geben den folgenden Code in Kommandozeile von Matlab ein.

```
fx >> num=xlsread('C:\Users\34067\Desktop\Model_01\Model_01_v15\Potsdam2019_xls.xlsx')
```

Danach bekommen wir die folgenden Tabelle, die TableName ist ,num'.

	1	2	3	4	5	6	7	8
1	1	1.0137e+...	7.8000	92	8	6.3000	260	0
2	2	1.0128e+...	7.9000	90	8	7.1000	260	0
3	3	1.0124e+...	7.7000	90	8	6.5000	260	0
4	4	1011	7.4000	90	8	6.8000	260	0
5	5	1.0097e+...	7.3000	88	8	7.8000	260	0
6	6	1.0085e+...	7.1000	89	8	8.2000	260	0
7	7	1.0075e+...	7.1000	85	8	8.3000	260	0
8	8	1.0067e+...	6.8000	87	8	8.6000	260	0.4622
9	9	1.0058e+...	6.6000	92	8	8.4000	250	16.0499
10	10	1.0054e+...	6.6000	95	8	8.9000	260	27.3437
11	11	1.0047e+...	6.6000	96	8	9.1000	260	42.6825
12	12	1.0039e+...	7.4000	92	7	9.1000	270	307.5450

Danach beschreiben das Model von Weatherdaten. Um die Weatherdaten einzulesen, benutzt man die Datenquelle 'Potsdam2019_mat.mat' und beschreibt file= 'modelica://Solarkollektor/ Potsdam2019_mat.mat' .

'modelica://Solarkollektor/' bedeutet, diese Datenquelle in Package 'Solarkollektor' verwendet werden. Und die Datenquelle 'Potsdam2019_mat.mat' bezeichnet die Weatherdatenquelle in das OM-Arbeitsverzeichnis. Danach kann diese mat-Datei von weatherdaten in model 'DataSource' und model 'Hauptprogramme' eingelesen werden.

```
model DataSource
  extends ICON_Table; "Verbindung von model und ICON"
  import SI = Modelica.SIunits;
```

"Einschreibung von Datenquelle"

```
parameter String file =
"modelica://Solarkollektor/Potsdam2019_mat.mat";
```

"Definition der Ausgabewerten von DataSource"

```
output SI.Irradiance G;
output Real Ta;
```

"Verbindung das model mit DataTable"

```
protected
  Modelica.Blocks.Sources.CombiTimeTable
  DataTable( columns = 1:8, fileName = file, smoothness =
  Modelica.Blocks.Types.Smoothness.ContinuousDerivative,
```

```

tableName = "num",tableOnFile = true, timeScale = 3600)
annotation(
    Placement(visible = true, transformation(origin = {-
62, 62}, extent = {{-10, -10}, {10, 10}}, rotation =
0)));

```

"Beschreibung der Gleichung für die Ausgabewerten"

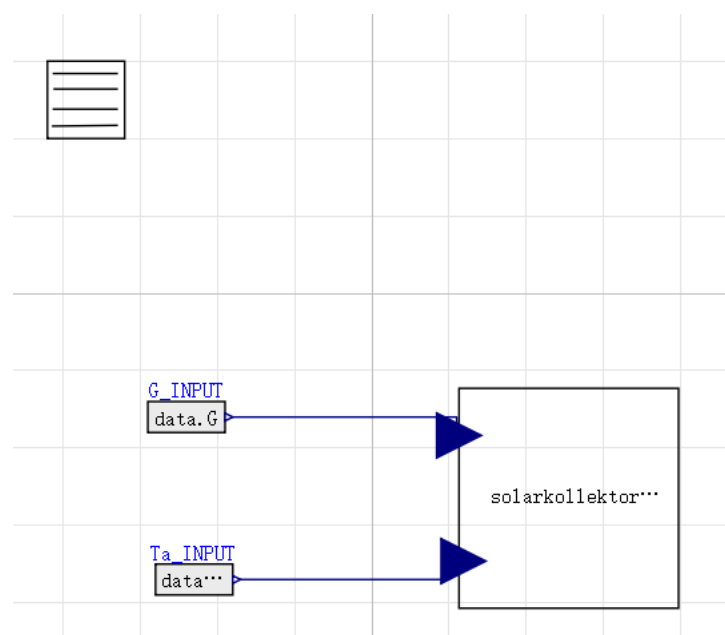
```

equation
    Ta = DataTable.y[3];
    G= DataTable.y[8];
end DataSource;

```

■ Hauptprogramme einschreiben und verbinden alle Komponenten

Zuletzt beschreiben Hauptprogramme, verbinden das Model von Solarkollektor und das Model von Weatherdaten mit die Block'Input'.



```

model Hauptprogramm
parameter String file_weather =
Modelica.Utilities.Files.loadResource("modelica://Solarko
llektor/Potsdam2019_mat.mat"); "Beschreibung der
Datenquelle"

```

"Verbindung mit allen Komponenten"

```

    Solarkollektor.Kollektormodell kollektormodell
annotation(

```



```

    Placement(visible = true, transformation(origin = {51,
-53}, extent = {{-29, -29}, {29, 29}}, rotation = 0)));
    Solarkollektor.DataSource data(file = file_weather)
annotation(
    Placement(visible = true, transformation(origin = {-
74, 50}, extent = {{-10, -10}, {10, 10}}, rotation =
0)));
Modelica.Blocks.Sources.RealExpression Ta_INPUT(y =
data.Ta) annotation(
    Placement(visible = true, transformation(origin = {-
46, -74}, extent = {{-10, -10}, {10, 10}}, rotation =
0)));
Modelica.Blocks.Sources.RealExpression G_INPUT(y =
data.G) annotation(
    Placement(visible = true, transformation(origin = {-
48, -32}, extent = {{-10, -10}, {10, 10}}, rotation =
0)));

equation
    connect(Ta_INPUT.y, kollektormodell.ta) annotation(
        Line(points = {{-34, -74}, {20, -74}, {20, -70}, {24,
-70}}, color = {0, 0, 127}));
    connect(G_INPUT.y, kollektormodell.Gt) annotation(
        Line(points = {{-36, -32}, {22, -32}, {22, -36}},
color = {0, 0, 127}));

    annotation(
        experiment(StartTime = 0, StopTime = 3.1536e+07,
Tolerance = 1e-06, Interval = 3600)); "SimulationsZeit
einstellen"
end Hauptprogramm;

```

- Um die Simulation zu laufen:

- Herunterladen und Speichern

Zuerst herunterladen und speichern Sie diese ‚Modell01_v15.zip‘ an der Desktop oder andere Speicherverzeichnis, danach öffnen Sie diese Solarkollektor.mo.

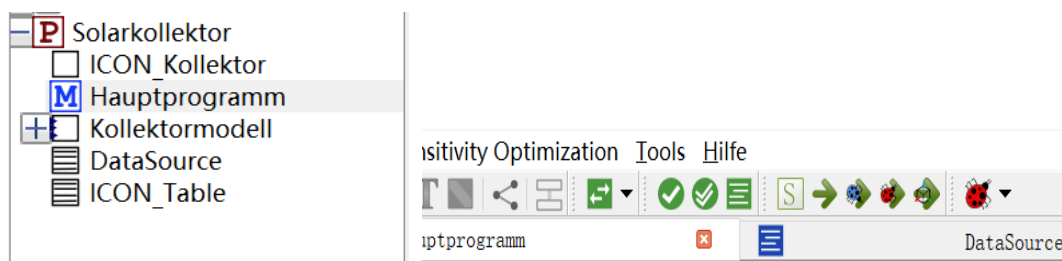
- Arbeitsverzeichnis einzustellen

Zweite klicken Sie ‘Tools’, danach ‘Optionen’, um das Arbeitsverzeichnis einzustellen. Dritte klicken Sie bitte ‘Durchführen’, und wählen Sie bitte Aktendeckel ‘Modell01_v15’.



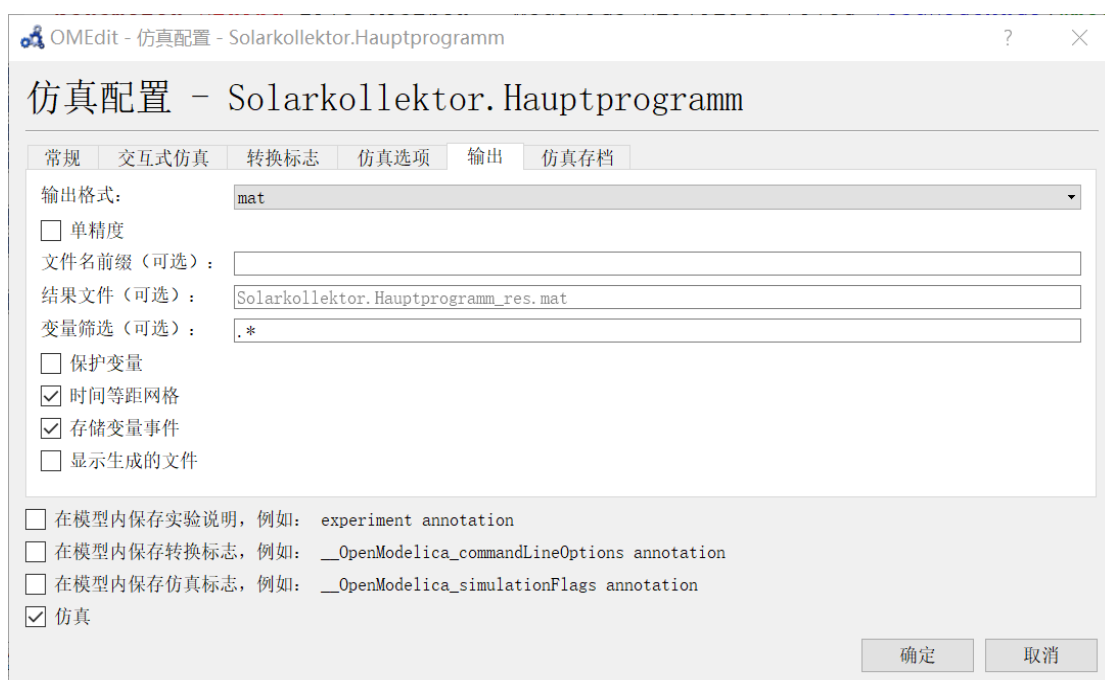
■ Simulation zu laufen

Zuletzt klicken 'Hauptprogramme', danach 'S' (,S' bedeutet ,Simulation'), dann beginnen die Simulation.



● Ergebnisse von Kollektormodell als mat-file output

Klicken Sie bitte 'S' und 'Ausgabe'. Hier habe ich die Mat-file in 'Solarkollektor.Hauptprogramm_output' von ,Modell01_v15' gespeichert, um die Ergebnistabelle von Openmodelica in MatData einzuspeichern.



- Ergebnisse von Openmodelica in Matlab zu analysieren

■ Reihenfolge und Name von Output-Varianten zu analysieren

Ausgegebene Varianten sind wie folgende Bild gezeigt.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	3600	7200	10800	14400	18000	21600	25200	28800	32400	36000	39600	43200
2	0	48.1890	48.1940	48.1846	48.1705	48.1657	48.1562	48.1563	48.1459	48.2713	48.3703	48.5035	50.8414
3	0.8824	3.5800e-06	-1.6000e-...	-3.2700e-...	-2.6300e-...	-1.9900e-...	-2.0900e-...	-2.4100e-...	1.1300e-06	3.1000e-05	2.7100e-05	4.6400e-05	3.6806e-04
4	0	0	0	0	0	0	0	0	0.4622	16.0499	27.3437	42.6825	307.5450
5	7.5500	7.8000	7.9000	7.7000	7.4000	7.3000	7.1000	7.1000	6.8000	6.6000	6.6000	6.6000	7.4000
6	0	0	0	0	0	0	0	0	0.7396	25.6798	43.7499	68.2920	492.0721
7	-8.7727e+...	-315.3842	-314.5140	-316.1400	-318.6115	-319.4401	-321.0852	-321.0821	-322.8829	-301.0587	-283.8090	-260.6151	146.5584
8	51.7100	-315.3483	-314.5300	-316.1727	-318.6378	-319.4600	-321.1061	-321.1062	-323.6112	-326.4287	-327.2874	-328.4427	-341.8330
9	-50	46.3780	46.3880	46.3693	46.3409	46.3314	46.3125	46.3125	46.2918	46.5425	46.7406	47.0070	51.6828
10	4.2109	4.1796	4.1796	4.1796	4.1796	4.1796	4.1795	4.1795	4.1795	4.1796	4.1796	4.1796	4.1804
11	50	50	50	50	50	50	50	50	50	50	50	50	50
12													

Diese Varianten zeigen keine Namen. Und wir können nicht von diese Tabelle wissen, welche Varianten welche Reihenfolge haben. Um die Reihenfolge und Name von Output-Varianten zu analysieren benutzen wir die Matlab-Code ,Reihenfolge_M01.mat' . Die Matlab-Code sind sowie folgende:

```
clc
clear all
%%
% load Sim_out.mat % individueller Name
derAusgabedatei, alle Var: .*
load
'Solarkollektor.Hauptprogramm_output/Solarkollektor.Haupt
programm_res.mat' % alternativ

%% Datenaufbereitung
[r,c] = size(data_2);

%% Header
```

```
col1 = string(name(:,1:r)');
col2 = string(description(:,1:r)');
data = (data_2);
res_0 = table(col1,col2,data);
res_0.Properties.VariableNames{1} = 'Bez_var';
res_0.Properties.VariableNames{2} = 'OM_var';
```

Dann bekommen wir die Reihenfolge und Namen von Varianten.

■ Ergebnisse von Openmodelica einlesen:

Einschreiben die Matlab-Code um die Simulationsergebnissen von OM in Matlab zu analysieren. Diese Code sind in ‚OManalyse01_mat.mat‘ gespeichert.

```
clc
```

```
load Solarkollektor.Hauptprogramm_res.mat “Einlesen die Weatherdaten in MatData”
```

```
data=data_2;
```

```
%%
```

```
“Definition des Parameters”
```

```
(Reihenfolge Von Varianten ist durch ‚Reihenfolge_M01.mat‘ zu bekommen)
```

```
ts      = data(1,:);
```

```
Tm      = data(2,:);
```

```
derTm   = data(3,:);
```

```
Ta      = data(5,:);
```

```
G       = data(4,:);
```

```
Qabs    = data(6,:);
```

```
Quse    = data(7,:);
```

```
Qver    = data(8,:);
```

```
Tco     = data(9,:);
```

■ Dateneingabe

```
T_ci=50;
```

```
th = ts/3600;
```

```
Quse(Quse<0)=0;
```

```
ctr=Quse./Quse;
```

```
ctr(isnan(ctr))=0;
```

■ Zuordnung der Ergebnisse

```
figure(1)
```

```
plot(th,Tco.*ctr,'.r');
```

```
hold on
```

```
plot(th,Tm.*ctr,'.m')
```

```
plot(th,Ta,'.b');
```

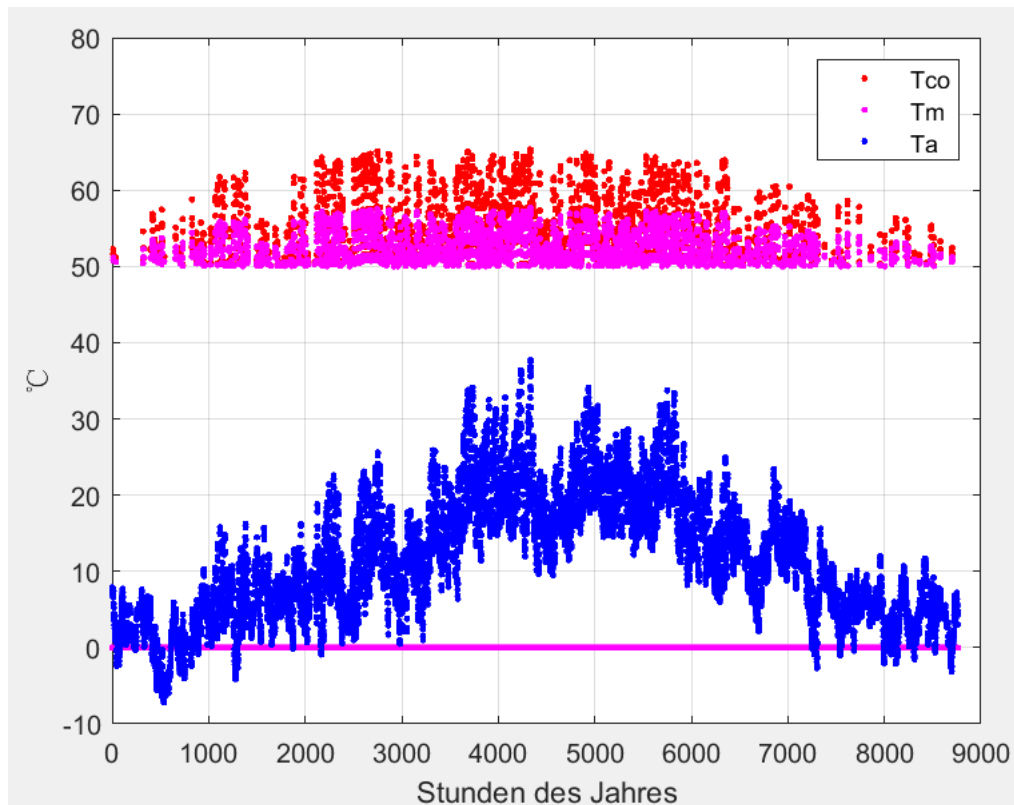
```
hold off
```

```
grid
```

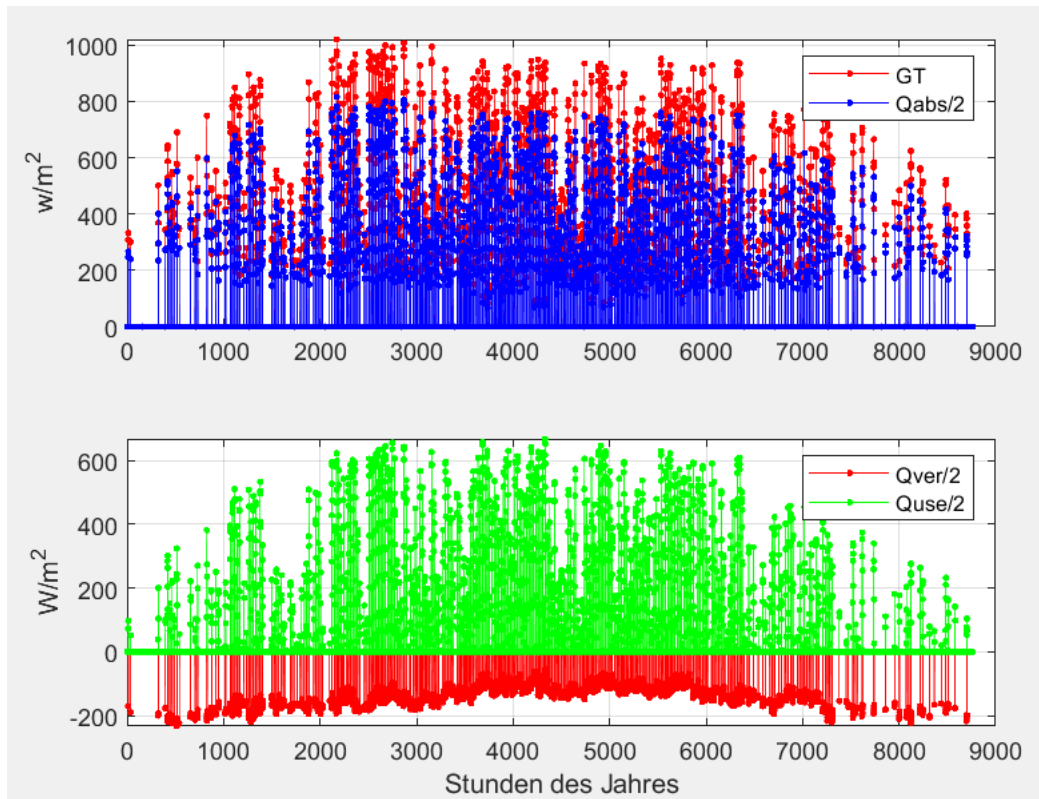
```
legend('Tco','Tm','Ta')
```

```
xlabel('Stunden des Jahres')
```

```
ylabel('jæ')
ylim([-10 80])
```



```
figure(2)
subplot(2,1,1)
plot(ts/3600 ,G.*ctr ,'.-r');
hold on
plot(ts/3600 ,Qabs/2.*ctr,'.-b');
hold off
legend('GT','Qabs/2')
grid
ylabel('w/m^2')
subplot(2,1,2)
plot(ts/3600 ,Qver/2.*ctr,'.-r');
hold on
plot(ts/3600 ,Quse/2.*ctr,'.-g');
hold off
grid
legend('Qver/2','Quse/2')
xlabel('Stunden des Jahres')
ylabel('W/m^2')
```



■ Solarertrag-Bildung von Monatswerten

```

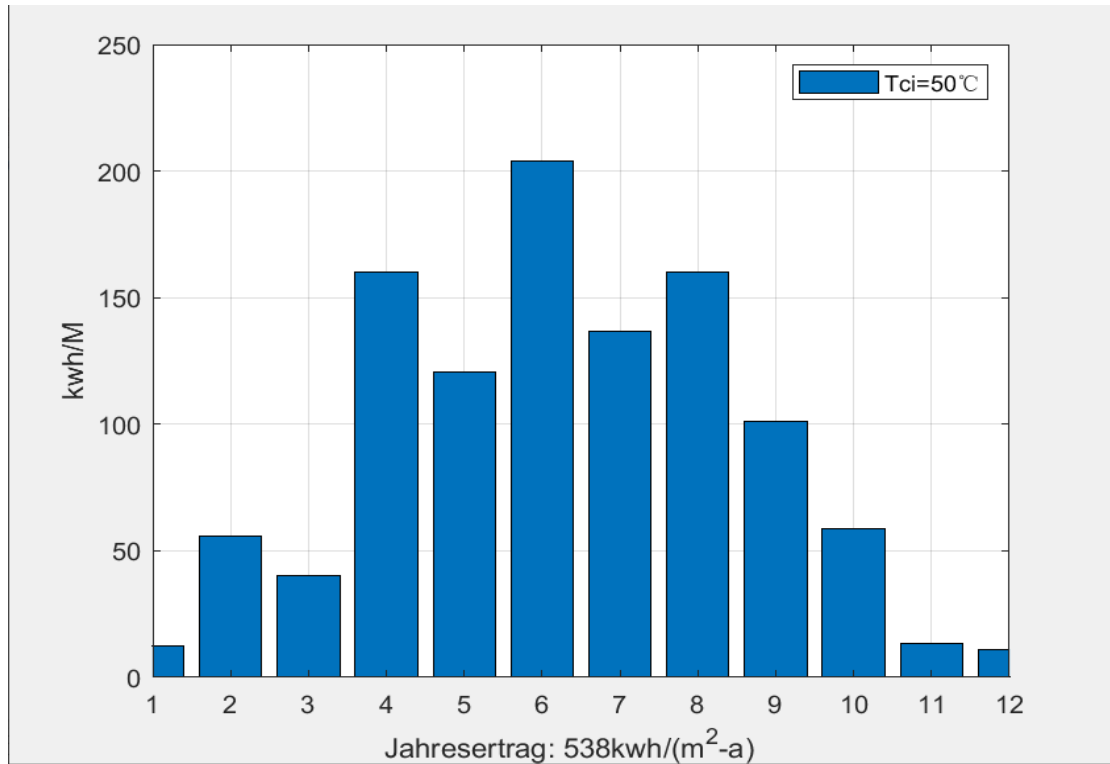
Quse=Quse';
QuseD=sum(reshape(Quse(1:8760,1),[24,365]))./1000;
QuseM(1,1)=sum(QuseD(1:31));
QuseM(1,2)=sum(QuseD(32:59));
QuseM(1,3)=sum(QuseD(60:90));
QuseM(1,4)=sum(QuseD(91:120));
QuseM(1,5)=sum(QuseD(121:151));
QuseM(1,6)=sum(QuseD(152:181));
QuseM(1,7)=sum(QuseD(182:212));
QuseM(1,8)=sum(QuseD(213:243));
QuseM(1,9)=sum(QuseD(244:273));
QuseM(1,10)=sum(QuseD(274:304));
QuseM(1,11)=sum(QuseD(305:334));
QuseM(1,12)=sum(QuseD(335:365));
%%
figure(3)
title 'Monatssummen'
bar(QuseM,'grouped');
grid
legend(['T{ci}=',num2str(Tci),';æ'])
xlim([1 12])

```

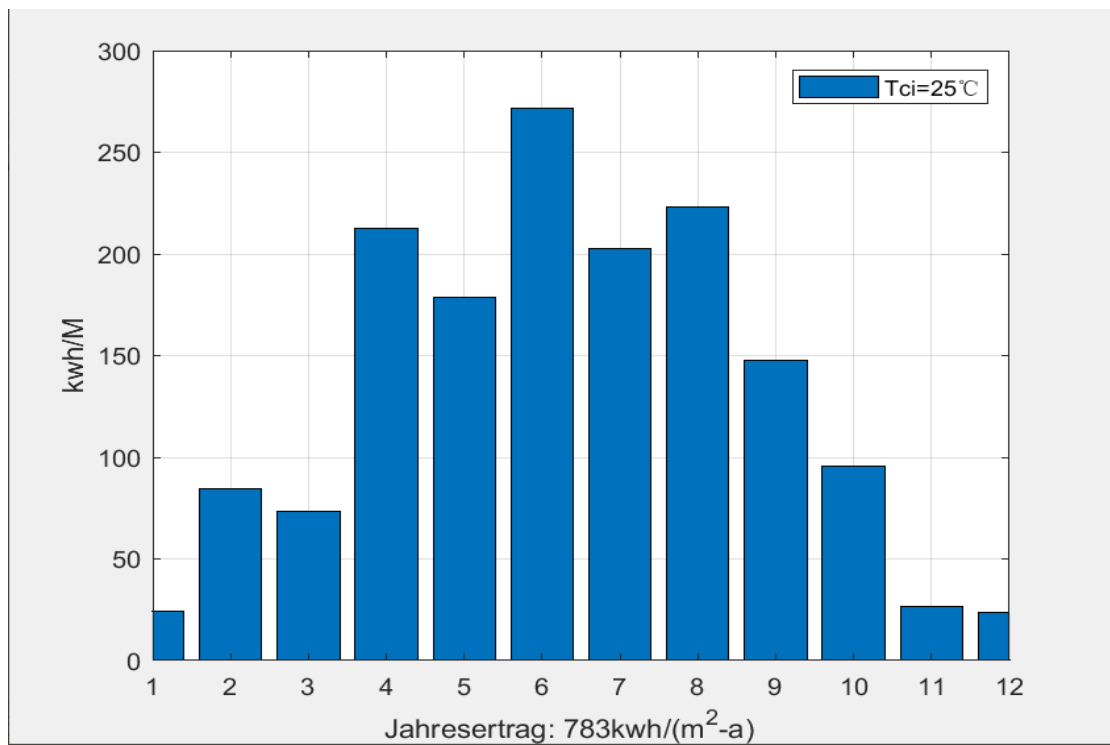
```

q_sol=sum(QuseM)/2;
Tmm=mean(Tm);
xlabel(['Jahresertrag: ',num2str(ceil(q_sol)),...
      'kwh/(m^2-a)   Mitteltemp.:
      ',num2str(ceil(Tmm)),';æ'])
ylabel('kwh/M')

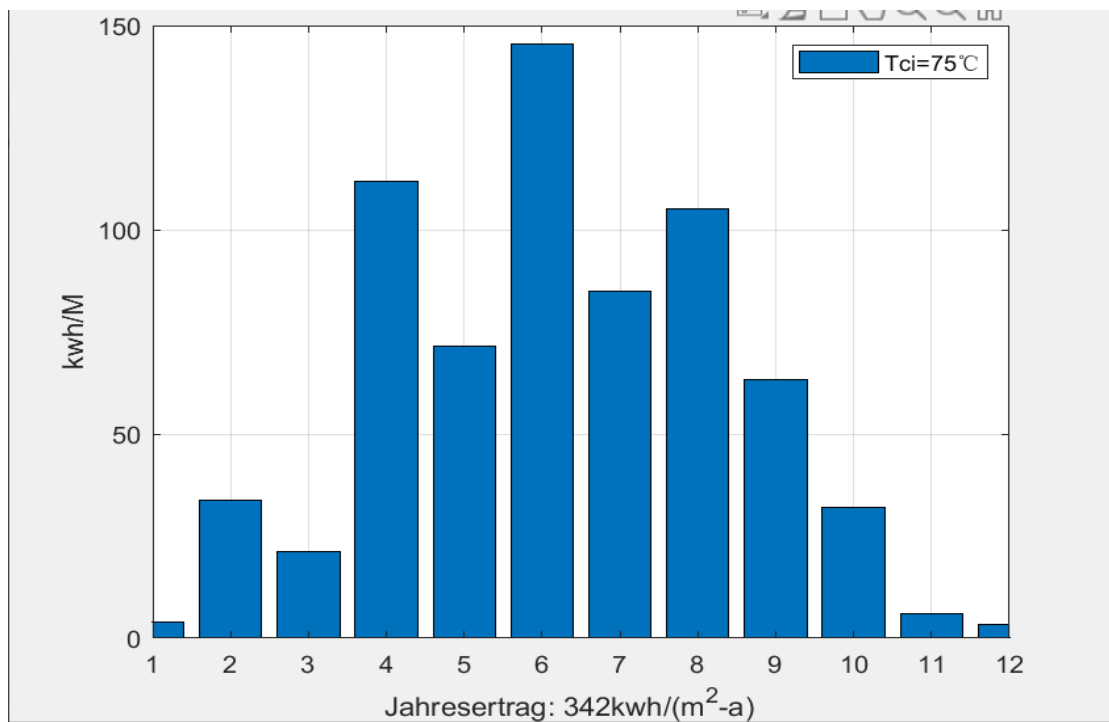
```



Wenn $T_{ci}=25$:



Wenn $T_{ci}=75$:



Folgerung:

Je kleiner die Eintrittstemperatur von Kollektor ist, ist der Jahresertrag je größer.