



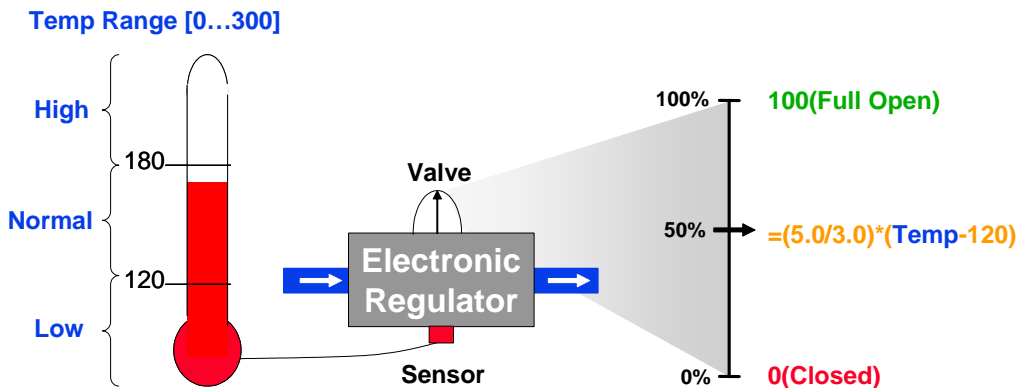
チュートリアル Regulator Exercise

<目的>

TTM の Mode Machines テーブルを用いて簡単な状態遷移をモデル化。 イベント式@T(式) や Guards Event(WHEN、WHERE、WHILE) を用いて状態遷移を定義します。

<要求仕様>

センサーで温度を監視しバルブを開閉する単純なレギュレータがあります。



Operational Summary

If Temp is less than Low, then valve is **Closed**

If Temp is greater than High, then valve is **Full Open**

If Temp is between High and Low, then valve is a function of Temp

操作： もし温度が Low (120 度) より小さいなら、バルブを閉じる。
もし温度が High (180 度) より大きいなら、バルブを最大に開く。
もし温度が High と Low の間なら、バルブの開度は温度の関数に依存。

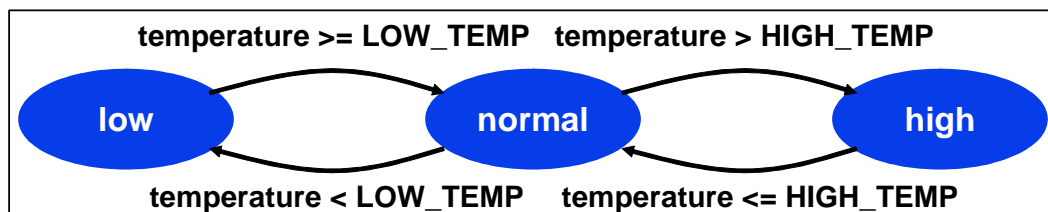
<分析>

入力タイプ：integer 型 0~300

出力タイプ：0、 $(5.0/3.0) * (\text{温度} - 120)$ 、100 モードに依存

定数：LOW_TEMP = 120, HIGH_TEMP = 180

動作の状態遷移は、以下のようなになるでしょう。



<設定>

TTM を起動し、regulator.ttm で保存します。

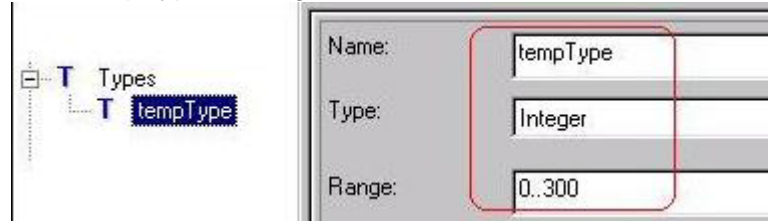
File メニューの Save As を使用します。

TVEC_Tutorial\exercises\regulator\regulator.ttm

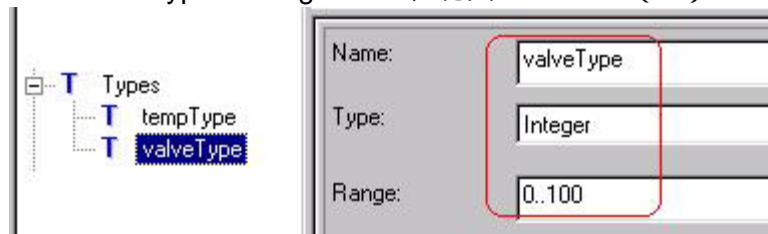
< インタフェース >

型 (Types) を作成します。

tempType : integer 型で、範囲は 0 ~ 300 (度)

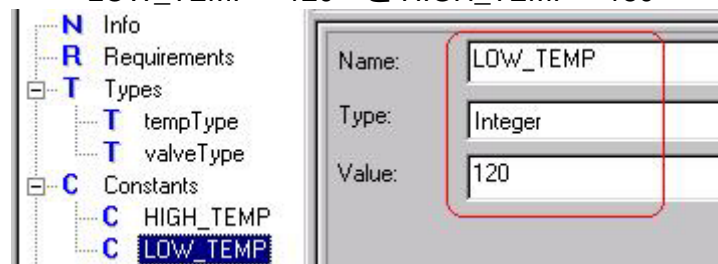


valveType : integer 型で、範囲は 0 ~ 100 (%)



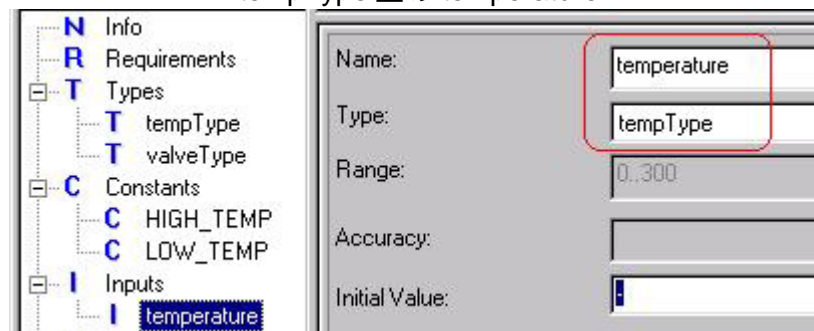
定数 (Constants) を作成します (Integer 型)

LOW_TEMP = 120 と HIGH_TEMP = 180

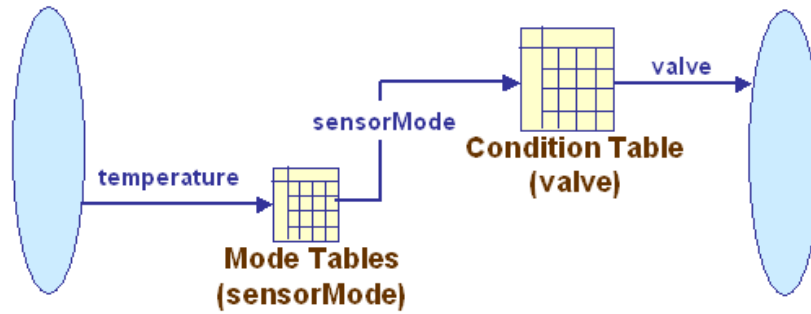


入力 (Inputs) を作成します。

tempType 型の temperature



< 振舞い > valve 出力は Mode Tables の出力（状態値）に依存します。



Mode Machines テーブルで sensorMode を作成します。

Modes タブで、下記モードを追加します。

#	Name	Initial	Final	Numerical Value
		<input type="checkbox"/>	<input type="checkbox"/>	
+ Add				

low, normal, high の 3 モードを追加します。

初期モードは low とします。（initial にチェックを入れます）

#	Name	Initial	Final	Numerical Value
1	low	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	normal	<input type="checkbox"/>	<input type="checkbox"/>	
3	high	<input type="checkbox"/>	<input type="checkbox"/>	

sensorMode テーブル内の Main タブで画面切替後、イベント式 @T(式) を使用して各状態遷移を定義。

例：low の時、@T(temperature >= LOW_TEMP) なら normal へ遷移
 @T(式)：式が偽から真、@F(式)：式が真から偽、
 @C(式)：式が変化（偽から真、あるいは真から偽）

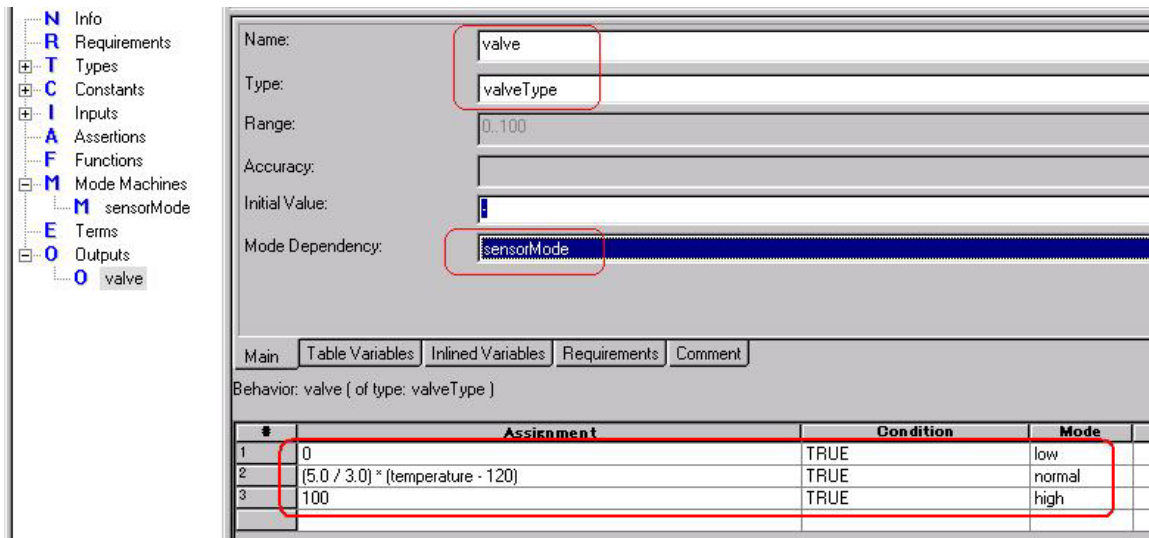
Transitions: sensorMode				
#	Source	Guard	Destination	Re
1	low	@T(temperature >= LOW_TEMP)	normal	
2	normal	@T(temperature > HIGH_TEMP)	high	
3	high	@T(temperature <= HIGH_TEMP)	normal	
4	normal	@T(temperature < LOW_TEMP)	low	

valveType 型の valve という出力テーブルを作成します。
 sensorMode に依存するよう Mode Dependency タブで、sensorMode を指定し、
 sensorMode の状態を基に、valve の値を制御するテーブルを完成させます。

low ならバルブを閉じる。(0)

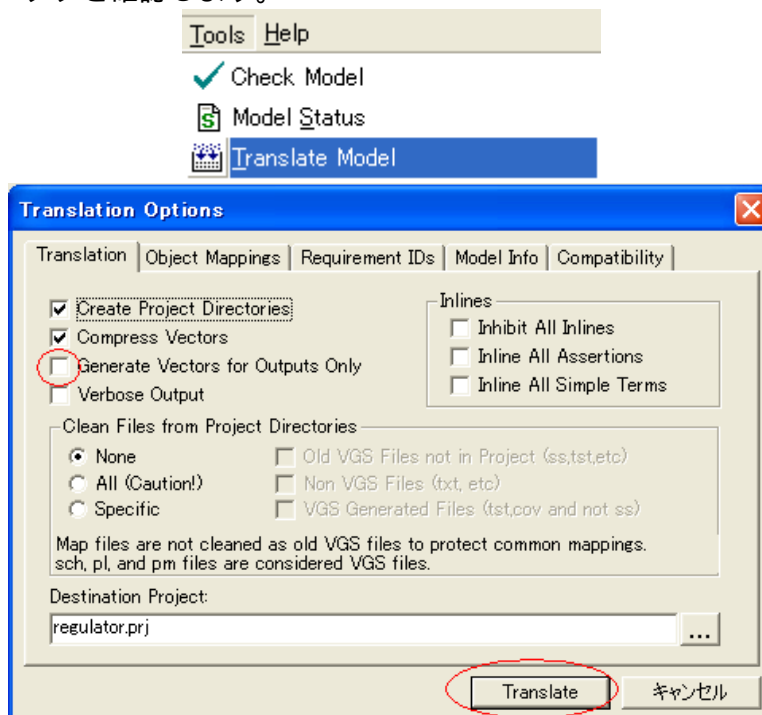
high ならバルブを最大に開く。(100)

nomal ならバルブの開度は関数に依存 ($(5.0/3.0) * (\text{temperature} - 120)$)

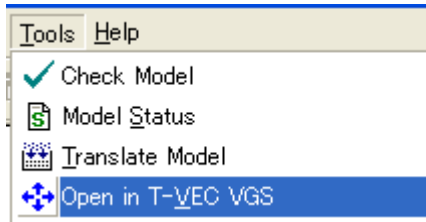


< モデルの変換とビルド >

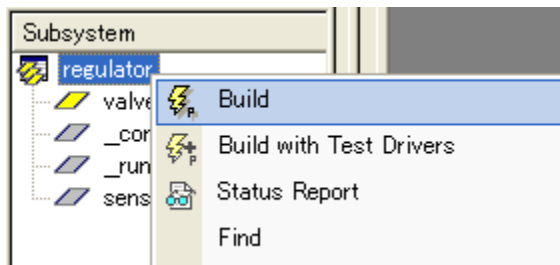
TTM の Translate Model を実行し、T-VEC を起動してプロジェクトをビルドします。その後、テストベクタを確認します。



T-VEC 起動



ビルド



sensorMode テストベクタを開く

#	sensorMode_OUT	_sensorMode	_temperature	temperature	Requirement IDs
1	low = 0	low = 0	0	0	ReqID(s) : No-change for mode low
2	low = 0	low = 0	119	119	ReqID(s) : No-change for mode low
3	normal = 1	normal = 1	120	120	ReqID(s) : No-change for mode normal
4	normal = 1	normal = 1	300	300	ReqID(s) : No-change for mode normal
5	normal = 1	normal = 1	0	0	ReqID(s) : No-change for mode normal
6	normal = 1	normal = 1	180	180	ReqID(s) : No-change for mode normal
7	high = 2	high = 2	181	181	ReqID(s) : No-change for mode high
8	high = 2	high = 2	300	300	ReqID(s) : No-change for mode high
9	normal = 1	low = 0	0	120	
10	normal = 1	low = 0	119	300	
11	high = 2	normal = 1	0	181	
12	high = 2	normal = 1	180	300	
13	low = 0	normal = 1	120	0	
14	low = 0	normal = 1	300	119	
15	normal = 1	high = 2	181	0	
16	normal = 1	high = 2	300	180	

NO CHANGE Tests

状態変数の現在値にはそのままの名前(例: temperature)で、前回値には接頭語 "_" が使用されます(例: _temperature)。下記ベクタでは、_temperature 119 から temperature 300 と温度変化することで、sensorMode が low から normal に遷移しています。

9	normal = 1	low = 0	0	120
10	normal = 1	low = 0	119	300

現在 Mode Machines テーブルでは low から normal への遷移は、@T(temperature >= LOW_TEMP) とだけ定義されています。しかし low から normal への遷移は、temperature >= LOW_TEMP かつ temperature < HIGH_TEMP です。

Guards Event(WHEN、WHERE、WHILE) を使うことで、このような条件を上手く記述することができます。

- E WHEN (D) : E が生じ、D が現状態で TRUE であった
- E WHERE (D) : E が生じ、D が次の状態で TRUE である
- E WHILE (D) : E が生じ、D が現状態と次の状態の両方で TRUE である

low から normal への遷移では、入力温度 temperature の範囲が 120 以上かつ 180 未満であることから、

@T(temperature >= LOW_TEMP) WHERE(temperature < HIGH_TEMP) となります。

また normal から high への遷移では、入力温度 temperature の範囲が 180 より大きいかつ前の温度 (normal 状態) が 120 以上であるから、

@T(temperature > HIGH_TEMP) WHEN(temperature >= LOW_TEMP) となります。

よって、SensorMode テーブルは、以下のようになります。

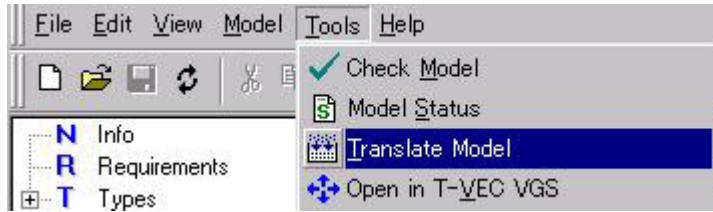
The screenshot shows the T-VEC Tabular Modeler interface for a file named 'regulator.ttm'. The left sidebar shows a tree view with 'sensorMode' selected under 'Mode Machines'. The main window displays two tables:

#	Name	Initial	Final	Enum Value
1	low	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	normal	<input type="checkbox"/>	<input type="checkbox"/>	
3	high	<input type="checkbox"/>	<input type="checkbox"/>	

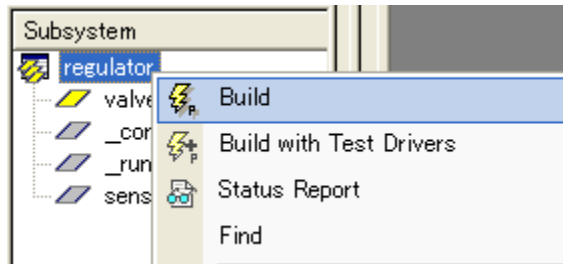
Below this is a 'Transitions: sensorMode' table:

#	Source	Guard	Destination
1	low	@T(temperature >= LOW_TEMP) WHERE(temperature < HIGH_TEMP)	normal
2	normal	@T(temperature > HIGH_TEMP) WHEN(temperature >= LOW_TEMP)	high
3	high	@T(temperature <= HIGH_TEMP) WHERE(temperature >= LOW_TEMP)	normal
4	normal	@T(temperature < LOW_TEMP) WHEN(temperature <= HIGH_TEMP)	low

モデルを保存後、再変換します。



T-VEC VGS へ再ロードしてテストベクタを再生成します。



生成された Guards を使ったベクタと Guards なしのベクタを比較します。

Guards 使用時のベクタ

6	normal = 1	normal = 1	180	180	ReqID(s) : NO-CHANGE CASE for mode normal
7	high = 2	high = 2	181	181	ReqID(s) : NO-CHANGE CASE for mode high
8	high = 2	high = 2	300	300	ReqID(s) : NO-CHANGE CASE for mode high
9	normal = 1	low = 0	0	120	
10	normal = 1	low = 0	119	179	
11	high = 2	normal = 1	120	181	
12	high = 2	normal = 1	180	300	
13	low = 0	normal = 1	120	0	
14	low = 0	normal = 1	180	119	
15	normal = 1	high = 2	181	120	
16	normal = 1	high = 2	300	180	

以上

ツールのデモンストレーションを行っています。

ご興味いただける場合は、お手数ですが下記までご依頼頂けると幸いです。



富士設備工業株式会社 電子機器事業部

〒591-8025 大阪府堺市北区長曾根町1928-1

Tel: 072-252-2128 www.fuji-setsu.co.jp