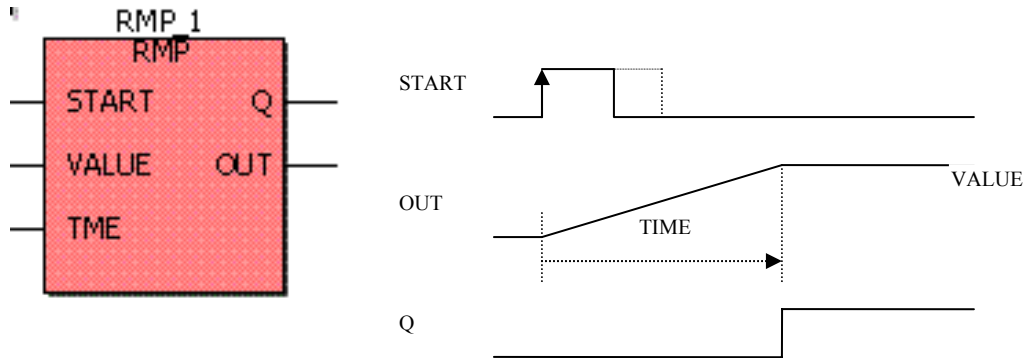


Specific PLC500N Firmware Function Block RMP

There are occasions that you may need to generate ramps in your control program. Although you can write your own POU's to generate ramps, a firmware function block RMP is provided to speed up program developments.

RMP FB has three inputs and two outputs that are described below.



Name	Function	Data Type	Description
START	Input	BOOL	A rising edge on this input starts generating a ramp. The Output "OUT" starts rising/ falling to reach the destination value specified on the "VALUE" input within the time-interval specified on the "TME" input.
VALUE	Input	INT	Destination value to be reached (from initial value)
TME	Input	TIME	Time interval to reach the destination value
Q	Output	BOOL	An output that shows if the destination value is reached
OUT	Output	INT	An output that shows the instantaneous value of ramp

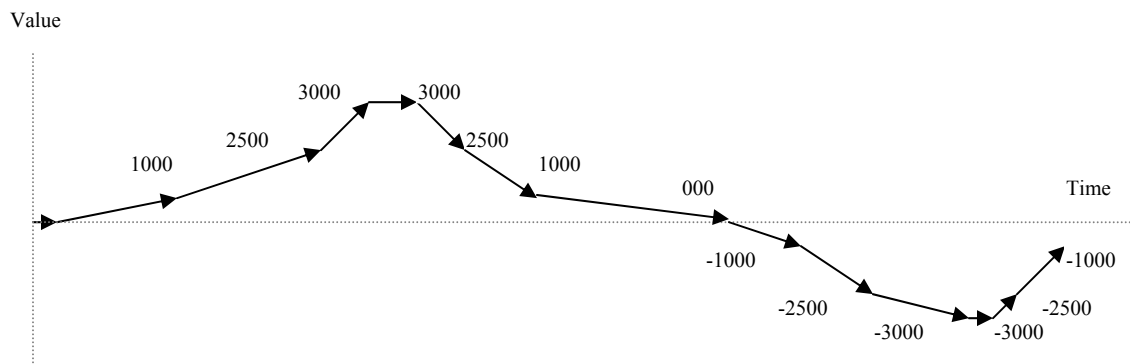
Note1. You may initialize the output "OUT" to a preset value by starting the RMP by introducing a value to the "VALUE" input and a zero time to the "TME" input. The preset value will be preserved for the subsequent operations.

Note2. You may introduce successive values and times to the "VALUE" and "TME" inputs and "START" the ramp generator.

PLC Specific Firmware Function Block RMP (Ramp Generator)

Example: Produce a multi-segment ramp with the integer values and time-intervals as specified below to be repeated continuously.

Ramp	Time interval (Seconds)	Value
Segment 0	0	0
Segment 1	6	1000
Segment 2	7	2500
Segment 3	4	3000
Segment 4	3	3000
Segment 5	3	2500
Segment 6	5	1000
Segment 7	10	0
Segment 8	5	-1000
Segment 9	7	-2500
Segment 10	4	-3000
Segment 11	3	-3000
Segment 12	3	-2500
Segment 13	5	-1000



PLC Specific Firmware Function Block RMP (Ramp Generator)

Please follow the steps below to implement the required parts.

1. Create a function block “Selector” in ST language.
2. Body of the function block “Selector” will be as follows.

```

IF ENABLE = FALSE
    THEN TIMER:=T#5S;
    RETURN;
END_IF;

CASE INPUT      OF
    INT#0      :      OUT:= INT#0;TIMER:=T#0S;
    INT#1      :      OUT:= INT#1000;TIMER:=T#6S;
    INT#2      :      OUT:= INT#2500;TIMER:=T#7S;
    INT#3      :      OUT:= INT#3000;TIMER:=T#4S;
    INT#4      :      OUT:= INT#3000;TIMER:=T#3S;
    INT#5      :      OUT:= INT#2500;TIMER:=T#3S;
    INT#6      :      OUT:= INT#1000;TIMER:=T#5S;
    INT#7      :      OUT:= INT#0;TIMER:=T#10S;
    INT#8      :      OUT:= INT#-1000;TIMER:=T#5S;
    INT#9      :      OUT:= INT#-2500;TIMER:=T#7S;
    INT#10     :      OUT:= INT#-3000;TIMER:=T#4S;
    INT#11     :      OUT:= INT#-3000;TIMER:=T#3S;
    INT#12     :      OUT:= INT#-2500;TIMER:=T#3S;
    INT#13     :      OUT:= INT#-1000;TIMER:=T#5S;
ELSE
    INPUT:=INT#-1;OUT:= INT#0; TIMER:=T#10S;
END_CASE;

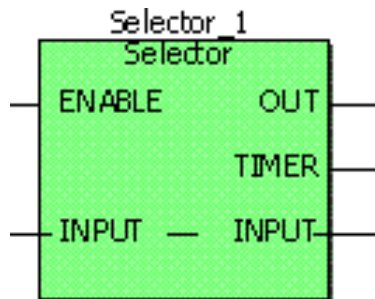
INPUT:=INPUT + 1;
    
```

3. Definition of the variables of the FB is as shown below.

```

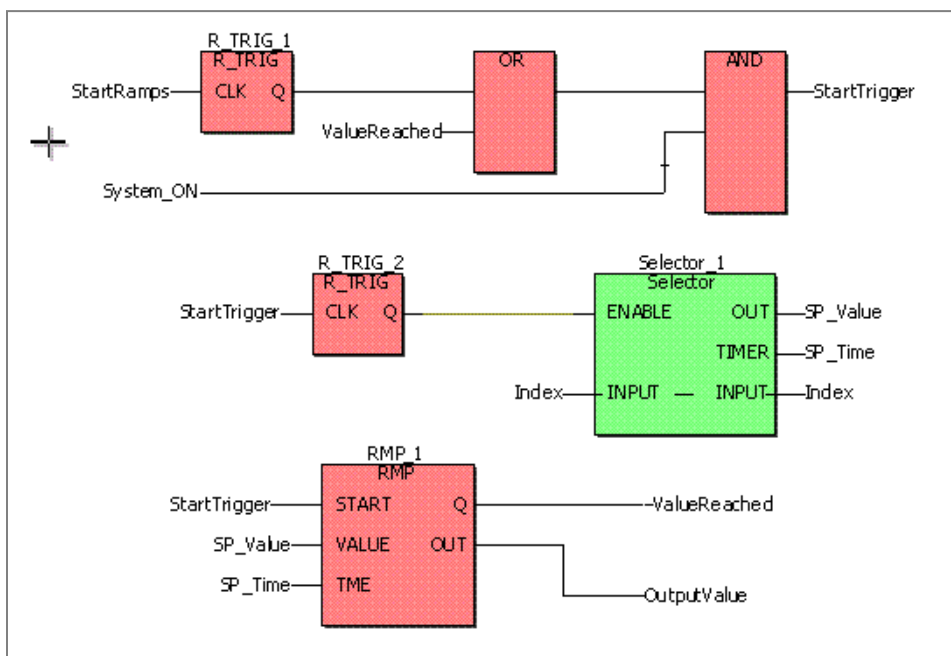
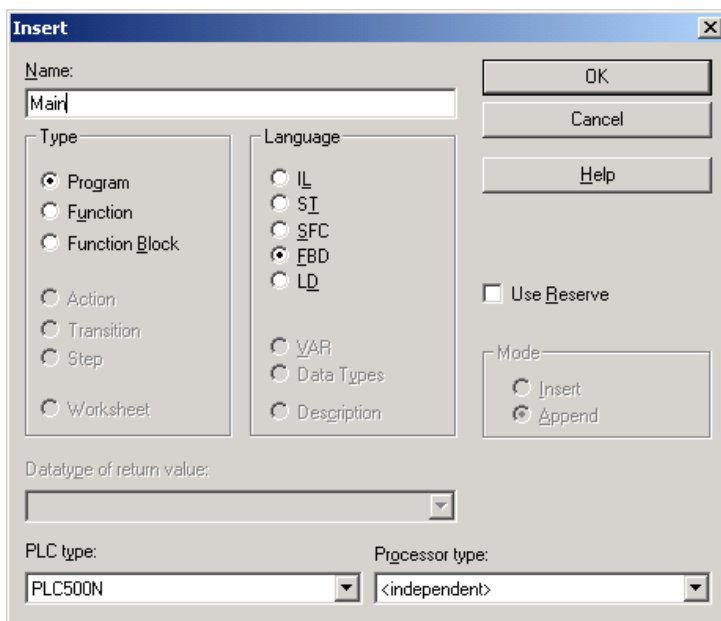
VAR_INPUT (*AUTOINSERT*)
    ENABLE :      BOOL;
END_VAR
VAR_OUTPUT (*AUTOINSERT*)
    OUT :      INT;
    TIMER :      TIME;
END_VAR
VAR_IN_OUT (*AUTOINSERT*)
    INPUT :      INT;
END_VAR
    
```

4. If you call this FB in a POU it will appear as shown below.



PLC Specific Firmware Function Block RMP (Ramp Generator)

5. Create a POU and insert an instance of FB “Selector” and an instance of the FB “RMP” and complete it as shown. Please note that the RMP function block is available from edit wizard when a POU is created with PLC type of PLC500N as shown below.



PLC Specific Firmware Function Block RMP (Ramp Generator)

Definition of the variables is as shown below.

```

VAR_EXTERNAL (*AUTOINSERT*)
    StartRamps      :      BOOL;
    System_ON       :      BOOL;
END_VAR
VAR (*AUTOINSERT*)
    Selector_1 :      Selector;
    Index      :      INT := 0;
    RMP_1      :      RMP;
    R_TRIG_1   :      R_TRIG;
    StartTrigger :      BOOL;
    R_TRIG_2   :      R_TRIG;
    ValueReached :      BOOL;
    SP_Value   :      INT;
    SP_Time    :      TIME;
    OutputValue :      INT;
END_VAR

```

6. There are some global variables that support VAR_EXTERNAL variables. Some of these are shown below. Insert the POU in a cyclic task and test your programs.

```

VAR_GLOBAL (*AUTOINSERT*)
    StartRamps      AT %IX0.3      :      BOOL;
    System_ON       AT %IX0.0      :      BOOL;
END_VAR

```