The Module's Software Input/Output = SW IO

IN_1 [LABEL] CLOCK_TICK [CALL] TIME_OUT_1 OUT_1
IN_2 [LABEL] INIT_MODULE [CALL] TIME_OUT_2 OUT_2
IN_3 [LABEL] RUN_MODULE [CALL] TIME_OUT_3 OUT_3
IN_4
IN_5 OUT_5
IN_6 OUT_6
IN_7 OUT_7
IN_8 OUT_8

SOFTWARE INPUT_PORTS:
IN_1 = [LABEL] CLOCK_TICK = Is an Interrupt Driven Input Port that should be called periodically, by a timer module such as S5E_TIMER0_CLOCK_MODULE.
When (the Output-Port of another Module Calls Input port IN_2 ) Then
(The code of this modules increments the 3 internal counters.)

SOFTWARE OUTPUT_PORTS:
OUT_1,2,3 = [CALL] TIME_OUT_1,2,3

The Assembler-Code of this module was Automatically created (by ECCM) by decoding user-entry into cells of a Spreadsheet of an Active-Form. That Active-Form guides and assists the user in the data-entry.

The features of the resulting-code:
Schedules ...Calls the input-ports of other modules at user defined TIME_INSTANTS.

More About S11A:
[1] S11A is a Time_Scheduler Module that is used in order to schedule the Execution of other modules.
[2] S11A has two parts. Part_A is an Interrupt driven part. Part_B is a Non_Interrupt driven part.
[2.1] Part_A must be fed at port S11A(IN_1) by Periodic Calls from real time clock [such as Module S5E(OUT_2)].
[2.2] Part_B must be fed at port S11A(IN_3) by Continuous Calls from a module such as S2A_MAIN_INFINITE_LOOP(OUT_5 or 6 or 7 or 8).
[3] S11A Part_A has 3 programmable Time_Out_1,2,3 counters. Each counter is incremented by the Periodic Calls (from IN_1) until it reaches its Time_Out_Value (Value named TO_1,2,3) and then restarts again to count from 0, infinite times.
Each time a counter reaches its Time_Out_Value (TO_1,2,3) it sets a bit named TO_BIT_1,2,3.
[4] S11A Part_B is activated by the Continuous Calls (from IN_3).
When activated, S11A checks the status of the bits TO_BIT_1,2,3.
If [bit TO_BIT_1 is set] Then [S11A issues a call (Named: Run_1) at port OUT_1];
That call (via OUT_1) is a Scheduled call that can be used in order to activate other modules periodically.
Also If [bit TO_BIT_1 is set] Then [S11A Resets bit TO_BIT_1].
[5] Also If [bit TO_BIT_2,3 is set] Then [S11A issues a call (Named: Run_2,3) at port OUT_2,3].
Similarly If [bit TO_BIT_2,3 is set] Then [S11A Resets bit TO_BIT_2,3].

About the Software Input/Output ports of Module S11A_REAL_TIME_SCHEDULER

SOFTWARE INPUT_PORTS:
IN_1 = [LABEL] CLOCK_TICK = Is an Interrupt Driven Input Port that should be called periodically, by a timer module such as S5E_TIMER0_CLOCK_MODULE.
When (the Output-Port of another Module Calls Input port IN_2 ) Then
(The code of this modules increments the 3 internal counters.)

IN_2 = [LABEL] INIT_MODULE = When (the Output-Port of another Module Calls Input port IN_2 ) Then
(This ECCM Module is initialized);

IN_3 = [LABEL] RUN_MODULE = When (the Output-Port of another Module Calls Input port IN_3 ) Then
(The code of this modules checks if TIME_OUT_1 or 2 or 3 occurred, and issues 3 independent periodic calls respectively via output ports OUT_1,2,3).

SOFTWARE OUTPUT_PORTS:
OUT_1,2,3 = [CALL] TIME_OUT_1,2,3
The code of this module issues Periodic Calls via output port OUT_1,2,3 respectively, to other modules > (the Periods = PERIOD_1,2,3, are defined by the user during the creation of the module).

http://www.embedded-controller-code-maker.com/
http://www.game-blocks-code.com/