Task counter V. 1.0

Descriptions

This function is launched every TIMER1 interrupt, by ENC_RTC() function. It checks overrun task problems, and shifts the enable condition to the new task (TASK1 ..TASK10)

Syntax: TaskCounter()

Input parameters: TASK0

Status: COUNT_TASK

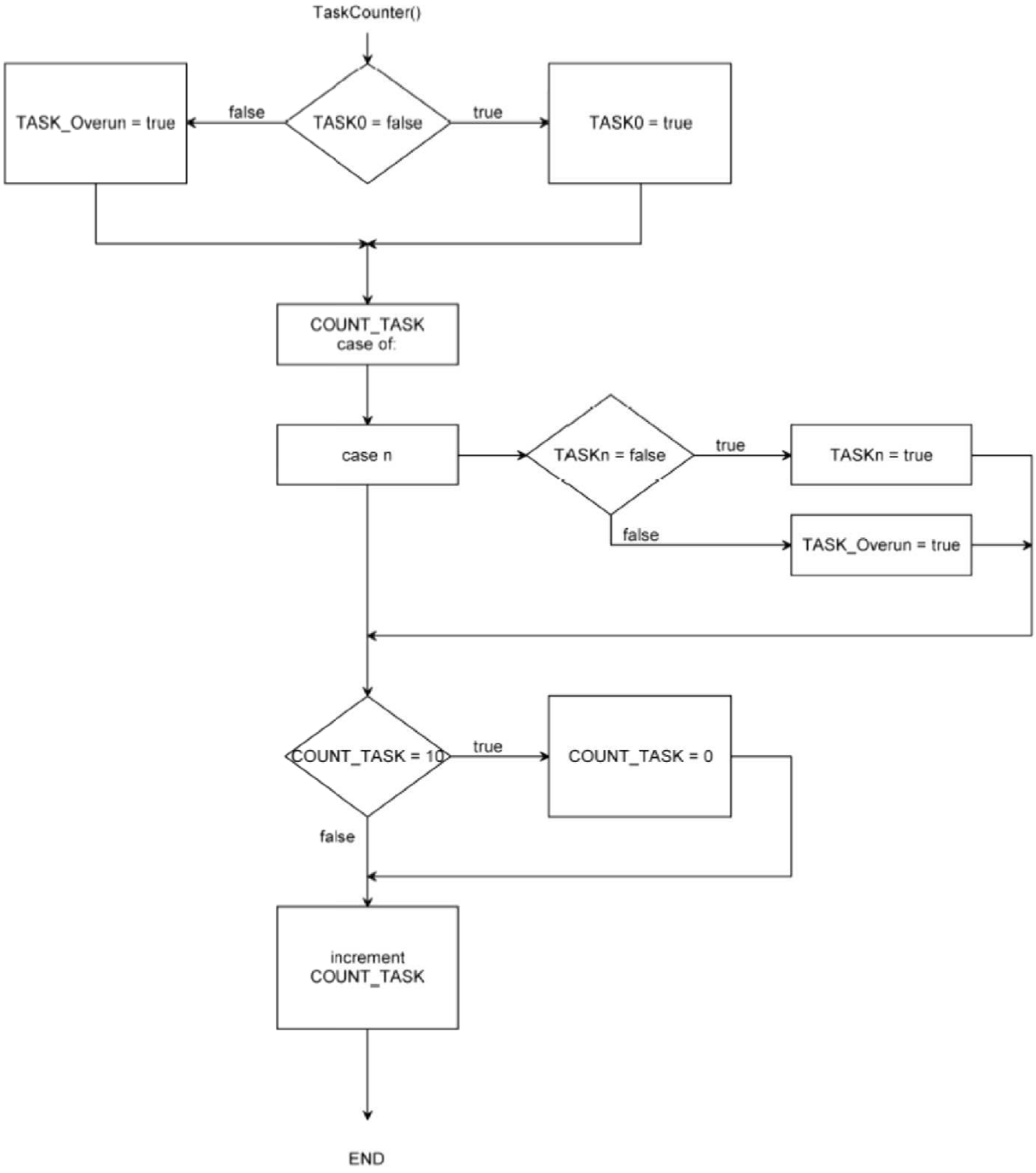
Code:

```
void TaskCounter()

{  if (!TASK0) {TASK0= true;}   else {TASK_OverRun = true;}

 // Switch case instruction (TASK 1 .. TASK 10) tested every 500 micro Sec.
//If enabled each tasks (1 to 10) is launched  at interval of 5 mSec

 switch (COUNT_TASK)
{
 case 1: if (!TASK1){TASK1= true;}   else {TASK_OverRun = true;}  break;
case 2: if (!TASK2){TASK2= true;}   else {TASK_OverRun = true;}  break;
case 3: if (!TASK3){TASK3= true;}   else {TASK_OverRun = true;}  break;
case 4: if (!TASK4){TASK4= true;}   else {TASK_OverRun = true;}  break;
case 5: if (!TASK5){TASK5= true;}   else {TASK_OverRun = true;}  break;
case 6: if (!TASK6){TASK6= true;}   else {TASK_OverRun = true;}  break;
case 7: if (!TASK7){TASK7= true;}   else {TASK_OverRun = true;}  break;
case 8: if (!TASK8){TASK8= true;}   else {TASK_OverRun = true;}  break;
case 9: if (!TASK9){TASK9= true;}   else {TASK_OverRun = true;}  break;
case 10:if(!TASK10){TASK10= true;}  else {TASK_OverRun = true;}  break;
}
 if (COUNT_TASK == 10){COUNT_TASK=0;}
 COUNT_TASK++;                    // COUNT_TASK is set with values from 1 to 10
}
```


TaskCounter()
TASK_Overun = true
false
TASK0 = false
true
TASK0 = true
COUNT_TASK
case of:
case n
TASKn = false
true
TASKn = true
false
TASK_Overun = true
COUNT_TASK = 10
true
COUNT_TASK = 0
false
increment
COUNT_TASK
END