

## 1. Create a new data type which is a structure

This uses the typedef and struct keywords in combination

```
typedef struct task {
    int state;
    unsigned long period;
    unsigned long elapsedTime;
    int (*TickFct)(int);
} task;
```

## 2. Call the Tick Function as a pointer to a function

In the Structure definition and declaration above:

```
int (*TickFct)(int);
```

Calling the tick function in the TimerISR Scheduler 'for loop'

```
tasks[i].state = tasks[i].TickFct(tasks[i].state);
```

## 3. Create an array of tasks

```
task tasks[2];
const unsigned char tasksNum = 2;
```

## 4. Move the scheduler to the TimerISR

```
for (i = 0; i < tasksNum; ++i) { // Heart of scheduler code
    if ( tasks[i].elapsedTime >= tasks[i].period ) { // Ready
        tasks[i].state = tasks[i].TickFct(tasks[i].state);
        tasks[i].elapsedTime = 0;
    }
    tasks[i].elapsedTime += tasksPeriodGCD;
}
```

## 5. Set up a global guard variable to detect interrupted interrupt

```
if (processingRdyTasks) {
    printf("Period too short to complete tasks\n");
}
```

## 6. In the Main, initialize each state machine's variables in the array. Repeat for each element in the array (each element is a state machine)

```
tasks[i].state = -1;
```

```
tasks[i].period = periodBlinkLed;
tasks[i].elapsedTime = tasks[i].period;
tasks[i].TickFct = &TickFct_BlinkLed;
```

## 7. Initialize the timer for the GCD

```
TimerSet(tasksPeriodGCD);
TimerOn();
```

## 8. Modify each state machine to be called with its state and return its state value

```
int TickFct_BlinkLed(int state) {
    .
    .
    return state;
}
```

## 9. Scheduler code details.... see comments below

```
for (i = 0; i < tasksNum; ++i) { // Heart of scheduler code .. loop for each SM
    if ( tasks[i].elapsedTime >= tasks[i].period ) { // Is elapsed time >= SM period?
        tasks[i].state = tasks[i].TickFct(tasks[i].state); // yes, then tick SM
        tasks[i].elapsedTime = 0; // reinit elapsed time because just ticked
    }
    tasks[i].elapsedTime += tasksPeriodGCD; // increment elapsed time
}
```

## 10. Idle loop now does nothing and can even go to sleep or to Operating System- the TimerISR wakes up and starts tick process

```
while(1) { Sleep(); }
```