

LESSON 5



Workflow

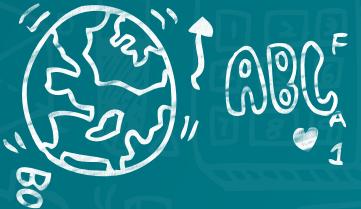
Program for embedded system



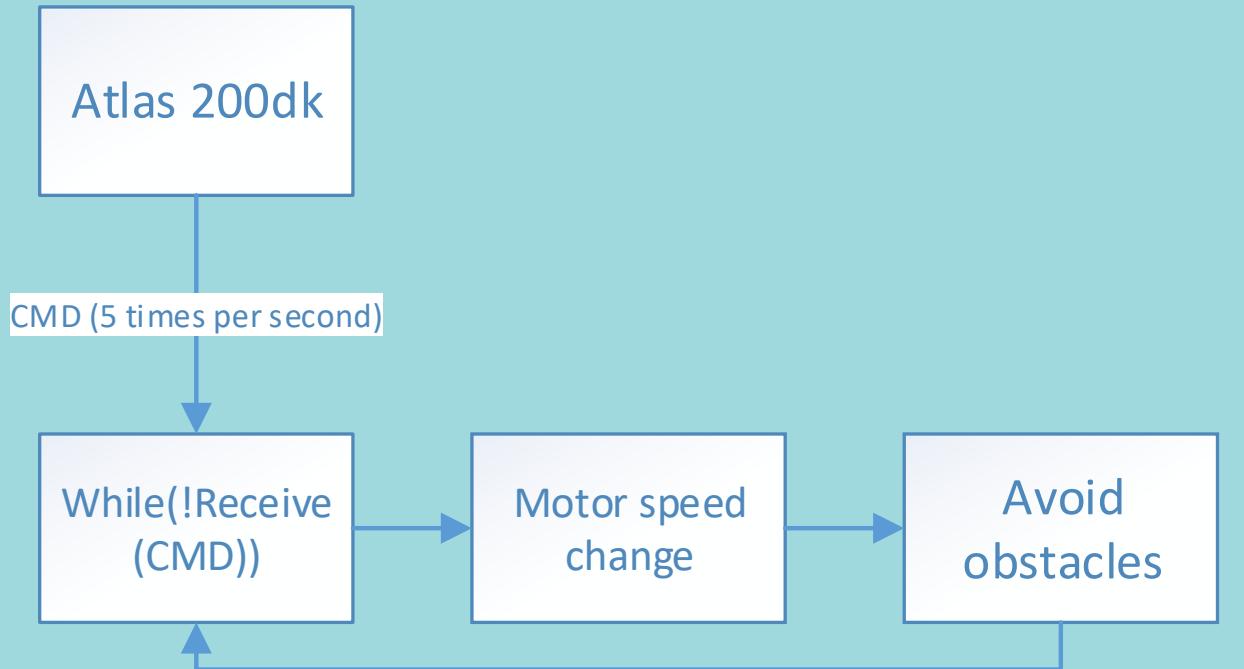
01

Problem scenario

Let embedded system to handle various tasks



Problem scenario



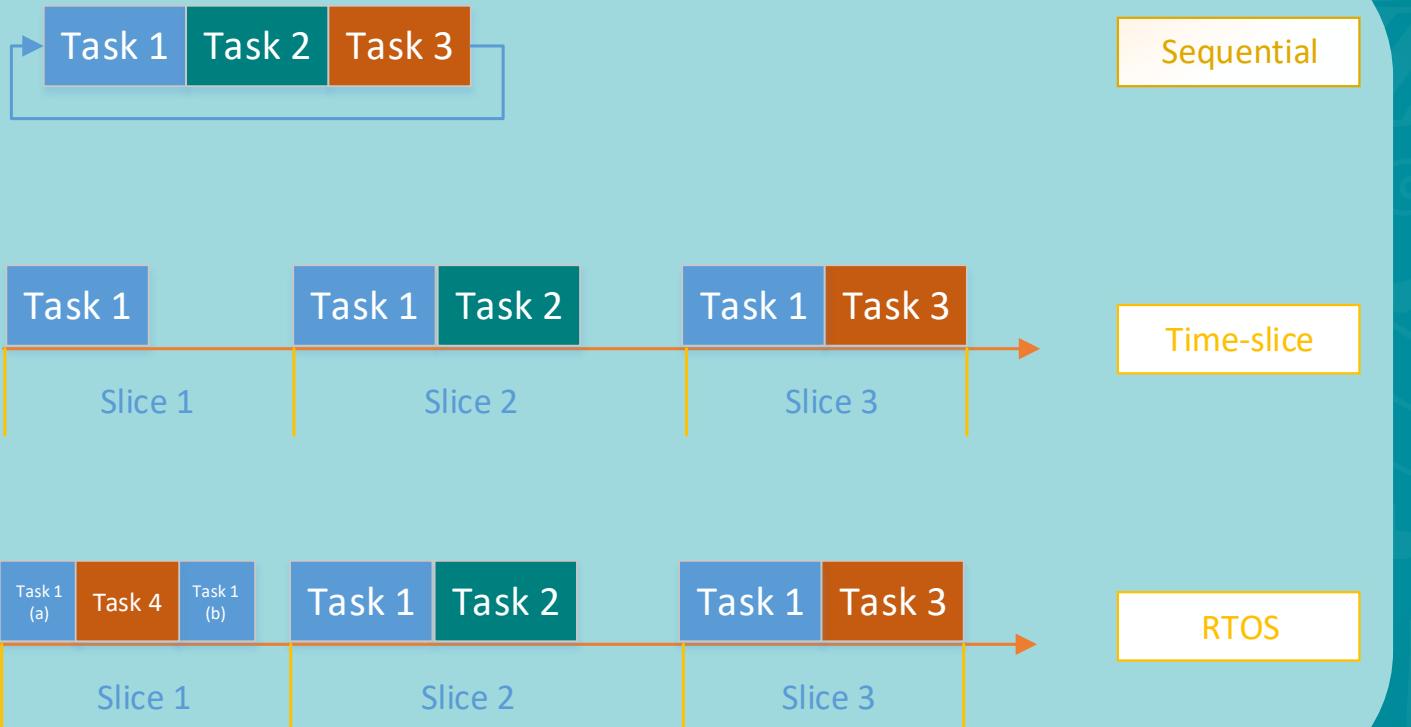
1. If the Bluetooth disconnect for few second,
2. If there was an obstacle in front of the car,
 1. The car will wait for the command from the Bluetooth Module and cannot avoid obstacles;
 2. The car cannot move unless move the car manly;

Disadvantage:

- Highly coupled between tasks
- Poor Maintainability
- Low efficiency (Barrel effect)



Program for embedded system



Sequential

Time-slice

RTOS

	Sequential	Time slice	RTOS
Development difficulty	+	++	+++
Resource	+	++	+++
Real Time	+++ (depend)	+	++
Maintainability	+	++	+++



02

Time slices

How to program using timing slices;



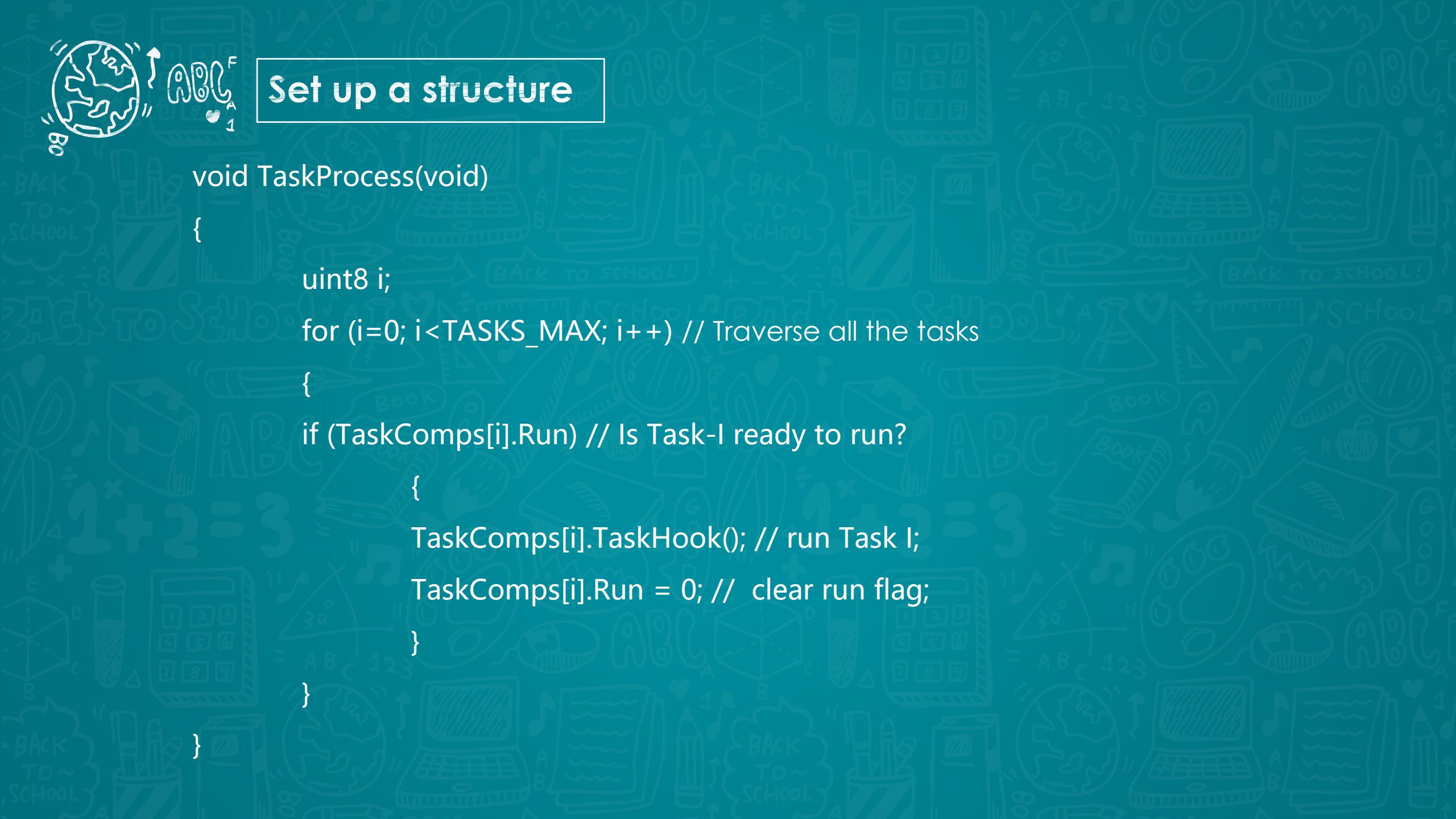
Set up a structure

```
// Task structure
typedef struct _TASK_COMPONENTS
{
    uint8 Run;          // 1:Task are ready to run
    uint8 Timer;        // Current Time
    uint8 ItvTime;      // Task run interval
    void (*TaskHook)(void); // The pointer to the task function to run
} TASK_COMPONENTS; // Task structure Definition
```



Task Remark

```
void TaskRemarks(void)
{
    uint8 i;
    for (i=0; i<TASKS_MAX; i++) // Traverse all the tasks
    {
        if (TaskComps[i].Timer) // TaskComps[i].Timer not equal to 0;
        {
            TaskComps[i].Timer--; //Timing: a unit of time has elapsed.
            if (TaskComps[i].Timer == 0) // It is the time for TaskComps[i] to run;
            {
                TaskComps[i].Timer = TaskComps[i].ItvTime; // Reload the timer
                TaskComps[i].Run = 1; //Task are ready to run
            }
        }
    }
}
```

The background of the slide features a repeating pattern of various educational and school-related icons in a light blue color. These icons include a globe, books, pencils, a calculator, musical notes, a brain, a test tube, a ruler, a magnifying glass, and various letters and numbers. The overall theme is "Back to School".

Set up a structure

```
void TaskProcess(void)
{
    uint8 i;
    for (i=0; i<TASKS_MAX; i++) // Traverse all the tasks
    {
        if (TaskComps[i].Run) // Is Task-I ready to run?
        {
            TaskComps[i].TaskHook(); // run Task I;
            TaskComps[i].Run = 0; // clear run flag;
        }
    }
}
```



03

How to build a task

How to program using timing slices;



Task in time slice

1. All tasks should finish before the next time slice;
2. Avoid *While(...)* or *delay()*;
3. Use Public variable to communicate between different tasks;

DEMO: L5 TimeSlicesDemo.ino

```
void getCMD()
{
    while (!Serial.available())
        inByte = Serial.read();
}
```



```
void getCMD()
{
    if (Serial.available())
        inByte = Serial.read();
}
```

Atlas 200dk



THANK YOU
FOR WATCHING