# AUTO模式任务调度相关程序结构分析

## 1.ArduCopter.cpp中 scheduler_tasks[] 的50hz任务：

*run_nav_updates()* [navigation.cpp]; 其调用 *run_autopilot()*;

## 2.*run_autopilot()* 判断是否AUTO模式，是则调用 *mission.update()*;

## 3.*mision.update()* 进行任务调度：

3.1. *mission.update()* 调度nav_command任务：

- I. *update()* 首先根据 *_flags.nav_cmd_loaded* 来判断当前nav任务状态，此标志为false表示无正在执行的任务，则调用 *advance_current_nav_cmd()* 载入下一个任务; *advance_current_nav_cmd()* 循环调用 *get_next_cmd()* 直到载入一个nav任务；
- II. *_flags.nav_cmd_loaded* 为true表示当前任务仍在执行，*update()* 调用 *_cmd_verify_fn(_nav_cmd)* 来调用具体的任务函数并检查返回值。返回true表示当前任务执行完成，将 *_flags.nav_cmd_loaded* 置为false，以便下次执行 *update()* 时载入新的任务;

3.2. *mission.update()* 调度do_command任务：

- 类似3.1，根据 *_flags.do_cmd_loaded* 判断当前do任务状态，然后调用 *_cmd_verify_fn(_do_cmd)* 或 *advance_current_do_cmd()* 来继续执行当前do任务或载入新的任务；

## 4. 函数指针封装

4.1. *_cmd_verify_fn* 数据类型为 *mission_cmd_fn_t*;
此数据类型的定义来自：

```
FUNCTOR_TYPEDEF(mission_cmd_fn_t,bool,const Mission_command&);
```

*FUNCTOR_TYPEDEF* 为宏定义(*AP_HAL/utility/functor.h*):

```
#define FUNCTOR_TYPEDEF(name, rettype, ...) \
    typedef Functor<rettype, ## __VA_ARGS__> name
```

Functor<>为一个模板类，重载了()运算符;
故 *mission_cmd_fn_t* 为输入参数为 *Mission_command* 类型，返回 *bool* 类型的函数指针类型;

4.2. *_cmd_start_fn* 的数据类型也为*mission_cmd_fn_t*，其被 *resume()*, *set_current_cmd()*, *advance_current_nav_cmd()* 和 *advance_current_do_cmd()* 调用，用于开始执行一个任务;

4.3. 同理还有另一变量*_mission_complete_fn*，数据类型来自

```
FUNCTOR_TYPEDEF(mission_complete_fn_t,void)
```

即无输入无输出的函数指针，指向任务结束时调用的函数，在 *AP_Mission::complete()* 中调用；

# 5. *Copter* 类和 *AP_Mission*类 的接口

5.1. *_cmd_start_fn*, *_cmd_verify_fn* 和 *_mission_complete_fn*都在 *AP_Mission* 类的构造函数中初始化：

```
AP_Mission(AP_AHRS &ahrs, mission_cmd_fn_t cmd_start_fn, mission_cmd_fn_t cmd_verify_fn,
mission_complete_fn_t mission_complete_fn) :
    _ahrs(ahrs),
    _cmd_start_fn(cmd_start_fn),
    _cmd_verify_fn(cmd_verify_fn),
    _mission_complete_fn(mission_complete_fn),
    _prev_nav_cmd_id(AP_MISSION_CMD_ID_NONE),
    _prev_nav_cmd_index(AP_MISSION_CMD_INDEX_NONE),
    _prev_nav_cmd_wp_index(AP_MISSION_CMD_INDEX_NONE),
    _last_change_time_ms(0)
{......};
```

5.2. Copter.cpp 中，在 Copter 的构造函数中，AP_Mission 类型变量 mission 的构造函数为：

```
mission(ahrs,
        FUNCTOR_BIND_MEMBER(&Copter::start_command, bool, const AP_Mission::Mission_Comma
nd &),
        FUNCTOR_BIND_MEMBER(&Copter::verify_command_callback, bool, const AP_Mission::Mis
sion_Command &),
        FUNCTOR_BIND_MEMBER(&Copter::exit_mission, void)),
```

*FUNCTOR_BIND_MEMBER* 同 *FUNCTOR_TYPEDEF*，也是 *AP_HAL/utility/functor.h* 中的宏定义：

```
#define FUNCTOR_BIND_MEMBER(func, rettype, ...) \
    Functor<rettype, ## __VA_ARGS__>::bind<std::remove_reference<decltype(*this)>::type,
func>(this)
```

从而，AP_Mission 类中的三个函数指针 **_cmd_start_fn**， **_cmd_verify_fn**， **_mission_complete_fn** 分别对应到了 Copter类的三个成员函数(*commands_logic.cpp*)：
**start_command()**, **verify_command_callback()**, **exit_mission()**.

---

FY
2016.10.05