

10 种模式

```
mkbldctrl.cpp  stm32_pwm.c  stm32_pwm.h  errno.h  navigator_ma...  »27
Firmware/src/modules/navigator/navigator_main.cpp
157  _param_acceptance_radius(this, "ACC_RAD"),
158  _param_dataLinkLoss_act(this, "DLL_ACT"),
159  _param_rcLoss_act(this, "RCL_ACT"),
160  _param_cruising_speed_hover(this, "MPC_XY_CRUISE", false),
161  _param_cruising_speed_plane(this, "FW_AIRSPD_TRIM", false),
162  _mission_cruising_speed(-1.0f)
163 {
164  /* Create a list of our possible navigation types */
165  _navigation_mode_array[0] = &_mission;
166  _navigation_mode_array[1] = &_loiter;
167  _navigation_mode_array[2] = &_rtl;
168  _navigation_mode_array[3] = &_dataLinkLoss;
169  _navigation_mode_array[4] = &_engineFailure;
170  _navigation_mode_array[5] = &_gpsFailure;
171  _navigation_mode_array[6] = &_rcLoss;
172  _navigation_mode_array[7] = &_takeoff;
173  _navigation_mode_array[8] = &_land;
174  _navigation_mode_array[9] = &_follow_target;
175
176  updateParams();
```

```
uORB.h  navigator_ma...  systemlib.h  stm32_tim.h  mkbldctrl.cpp
Firmware/src/modules/navigator/navigator_main.cpp
380  /* vehicle control mode updated */
381  orb_check(_control_mode_sub, &updated);
382  if (updated) {
383      vehicle_control_mode_update();
384  }
385
386  /* vehicle status updated */
387  orb_check(_vstatus_sub, &updated);
388  if (updated) {
389      vehicle_status_update();
390  }
391
```

```
uORB.h  navigator_ma...  systemlib.h  stm32_tim.h  mkbldctrl.cpp  stm32_pwm.c  stm32_pwm.h  errno.h  »24
Firmware/src/modules/navigator/navigator_main.cpp
232
233 void
234 Navigator::fw_pos_ctrl_status_update()
235 {
236     orb_copy(ORB_ID(fw_pos_ctrl_status), _fw_pos_ctrl_status_sub, &fw_pos_ctrl_status);
237 }
238
239 void
240 Navigator::vehicle_status_update()
241 {
242     if (orb_copy(ORB_ID(vehicle_status), _vstatus_sub, &vstatus) != OK) {
243         /* in case the commander is not be running */
244         _vstatus.arming_state = vehicle_status_s::ARMING_STATE_STANDBY;
245     }
246 }
247
248 void
```

```
sensors.cpp  sensor_combi...  commander.cpp  uORB.h  navigator_ma...  systemlib.h  stm32_tim.h  Firmware/src/modules/navigator/navigator_main.cpp
515
516 /* Do stuff according to navigation state set by commander */
517 switch (_vstatus.nav_state) {
518     case vehicle_status_s::NAVIGATION_STATE_MANUAL:
519     case vehicle_status_s::NAVIGATION_STATE_ACRO:
520     case vehicle_status_s::NAVIGATION_STATE_ALTCTL:
521     case vehicle_status_s::NAVIGATION_STATE_POSCTL:
522     case vehicle_status_s::NAVIGATION_STATE_TERMINATION:
523     case vehicle_status_s::NAVIGATION_STATE_OFFBOARD:
524         _navigation_mode = nullptr;
525         _can_loiter_at_sp = false;
526         break;
527     case vehicle_status_s::NAVIGATION_STATE_AUTO_MISSION:
528         if (_fw_pos_ctrl_status.abort_landing) {
529             // pos controller aborted landing, requests loiter
530             // above landing waypoint
531             _navigation_mode = &loiter;
532             _pos_sp_triplet_published_invalid_once = false;
533         } else {
534             _pos_sp_triplet_published_invalid_once = false;
535             _navigation_mode = &mission;
536         }
537         break;
538     case vehicle_status_s::NAVIGATION_STATE_AUTO_LOITER:
539         _pos_sp_triplet_published_invalid_once = false;
540         _navigation_mode = &loiter;
541         break;
542     case vehicle_status_s::NAVIGATION_STATE_AUTO_RCRECOVER:
543         _pos_sp_triplet_published_invalid_once = false;
544         if (_param_rcloss_act.get() == 1) {
545             _navigation_mode = &loiter;
546         }
547         break;
548 }
549
550 _navigation_mode = nullptr;
551 _can_loiter_at_sp = false;
552 break;
553 }
554
555 /* iterate through navigation modes and set active/inactive for each */
556 for (unsigned int i = 0; i < NAVIGATOR_MODE_ARRAY_SIZE; i++) {
557     _navigation_mode_array[i]->run(_navigation_mode == _navigation_mode_array[i]);
558 }
559 }
```

选择模式，进入对应的 cpp

- ▼ navigator
 - > .c datalinkloss_params.c
 - > .c datalinkloss.cpp
 - > .h datalinkloss.h
 - > .c enginefailure.cpp
 - > .h enginefailure.h
 - > .c follow_target_params.c
 - > .c follow_target.cpp
 - > .h follow_target.h
 - > .c geofence_params.c
 - > .c geofence.cpp
 - > .h geofence.h
 - > .c gpsfailure_params.c
 - > .c gpsfailure.cpp
 - > .h gpsfailure.h
 - > .c land.cpp
 - > .h land.h
 - > .c loiter.cpp
 - > .h loiter.h
 - > .c mission_block.cpp
 - > .h mission_block.h
 - > .c mission_feasibility_checker.cpp
 - > .h mission_feasibility_checker.h
 - > .c mission_params.c
 - > .c mission.cpp
 - > .h mission.h
 - > .h navigation.h
 - > .c navigator_main.cpp
 - > .c navigator_mode.cpp
 - > .h navigator_mode.h
 - > .c navigator_params.c
 - > .h navigator.h
 - > .c rclose_params.c
 - > .c rclose.cpp
 - > .h rclose.h
 - > .c rtl_params.c
 - > .c rtl.cpp
 - > .h rtl.h
 - > .c takeoff.cpp
 - > .h takeoff.h
 - CMakeLists.txt

```

70 }
71
72 void
73 Land::on_activation()
74 {
75     /* set current mission item to Land */
76     set_land_item(&mission_item, true);
77     _navigator->get_mission_result()->reached = false;
78     _navigator->get_mission_result()->finished = false;
79     _navigator->set_mission_result_updated();
80     reset_mission_item_reached();
81
82     /* convert mission item to current setpoint */
83     struct position_setpoint_triplet_s *pos_sp_triplet = _navigator->get_position_setpoint_triplet();
84     pos_sp_triplet->previous.valid = false;
85     mission_item_to_position_setpoint(&mission_item, &pos_sp_triplet->current);
86     pos_sp_triplet->next.valid = false;
87
88     _navigator->set_can_loiter_at_sp(false);
89
90     _navigator->set_position_setpoint_triplet_updated();
91 }
92
93 void
94 Land::on_active()
95 {
96     if (is_mission_item_reached() && !_navigator->get_mission_result()->finished) {
97         _navigator->get_mission_result()->finished = true;
98         _navigator->set_mission_result_updated();
99         set_idle_item(&mission_item);
100
101         struct position_setpoint_triplet_s *pos_sp_triplet = _navigator->get_position_setpoint_triplet();
102         mission_item_to_position_setpoint(&mission_item, &pos_sp_triplet->current);
103         _navigator->set_position_setpoint_triplet_updated();
104     }
105 }
106

```

每个 cpp 都有这些东西，经过各自的操作产生

```

574 item->time_inside = 0.0f;
575 item->pitch_min = 0.0f;
576 item->autocontinue = true;
577 item->origin = ORIGIN_ONBOARD;
578 }
579
580 void
581 MissionBlock::set_idle_item(struct mission_item_s *item)
582 {
583     item->nav_cmd = NAV_CMD_IDLE;
584     item->lat = _navigator->get_home_position()->lat;
585     item->lon = _navigator->get_home_position()->lon;
586     item->altitude_is_relative = false;
587     item->altitude = _navigator->get_home_position()->alt;
588     item->yaw = NAN;
589     item->loiter_radius = _navigator->get_loiter_radius();
590     item->loiter_direction = 1;
591     item->acceptance_radius = _navigator->get_acceptance_radius();
592     item->time_inside = 0.0f;
593     item->pitch_min = 0.0f;
594     item->autocontinue = true;
595     item->origin = ORIGIN_ONBOARD;
596 }

```

```

>void
MissionBlock::mission_item_to_position_setpoint(const struct mission_item_s *item, struct position_setpoint_s *sp)
{
    /* set the correct setpoint for vtol transition */

    if(item->nav_cmd == NAV_CMD_DO_VTOL_TRANSITION && PX4_ISFINITE(item->yaw)
        && item->params[0] >= vtol_vehicle_status_s::VEHICLE_VTOL_STATE_FW - 0.5f) {
        sp->type = position_setpoint_s::SETPOINT_TYPE_POSITION;
        waypoint_from_heading_and_distance(_navigator->get_global_position()->lat,
            _navigator->get_global_position()->lon,
            item->yaw,
            1000000.0f,
            &sp->lat,
            &sp->lon);
        sp->alt = _navigator->get_global_position()->alt;
    }

    /* don't change the setpoint for non-position items */
    if (!item_contains_position(item)) {
        return;
    }

    sp->valid = true;
    sp->lat = item->lat;
    sp->lon = item->lon;
    sp->alt = item->altitude_is_relative ? item->altitude + _navigator->get_home_position()->alt : item->altitude;
    sp->yaw = item->yaw;
    sp->loiter_radius = (item->loiter_radius > NAV_EPSILON_POSITION) ? item->loiter_radius :
        _navigator->get_loiter_radius();
    sp->loiter_direction = item->loiter_direction;
    sp->pitch_min = item->pitch_min;
    sp->acceptance_radius = item->acceptance_radius;
    sp->disable_mc_yaw_control = false;
    sp->cruising_speed = _navigator->get_cruising_speed();

    switch (item->nav_cmd) {

```

最后是发布

```

uORB.h  navigator_ma...  land.cpp  systemlib.h  mission_bloc...  stm32_tim.h  mkbldctrl
605      }
606      Firmware/src/modules/navigator/navigator_main.cpp
607      /* if nothing is running, set position setpoint triplet invalid once */
608      if (_navigation_mode == nullptr && !_pos_sp_triplet_published_invalid_once) {
609          _pos_sp_triplet_published_invalid_once = true;
610          _pos_sp_triplet.previous.valid = false;
611          _pos_sp_triplet.current.valid = false;
612          _pos_sp_triplet.next.valid = false;
613          _pos_sp_triplet_updated = true;
614      }
615
616      if (_pos_sp_triplet_updated) {
617          publish_position_setpoint_triplet();
618          _pos_sp_triplet_updated = false;
619      }
620
621      if (_mission_result_updated) {
622          publish_mission_result();
623          _mission_result_updated = false;
624      }
625
626      perf_end(_loop_perf);
627  }

```