Waypoint Live Recording

By Angel Fernandez
(aka “afernan”)
Sep´13

WLR specification

The goal of this "mod" is to enable AQ code with the possibility of recording waypoints (wp´s) during flight.
By using the transmitter "gear" switch (ch5), each time that is set, a new waypoint is added to the list. Then you can run it with “auto” mode, as usual, or download to GCS to edit.
This feature lets you fly auto missions without the need of any gcs. Convenient sound signals are set to drive you in the process.

This mod is done in AQ6.6 **r220**

Main features:

- It can record wp´s in “***manual***” or “***position hold***” mode
- It can record wp´s while in ARMED or DISARMED
- 3D position recorded. Absolute altitude recorded (wp´s have 3D info)
- To delete WP´s simply set ROLL stick to the top left while in DISARMED

See how performs in this video

Code changes explanation

(Annex 1 shows complete list of code lines changed)

There are only two file affected: ***nav.c*** and ***nav.h***

The basic idea is to introduce in the "main loop" inside "nav.c"

```
void navNavigate(void){}
```

an "if" statement that controls when switch "gear" is activated, and then launch the recording of a new wp thru an specific function.

Changes in "nav.c"

When the Tx gear switch is set…

```
if (RADIO_GEAR > 250) {
```

We want the "main loop" runs very quickly. Since our switching by hand will take some time, we don´t want to record many times same WP. This is known as « switch debouncing » problem. To avoid that effect we set a flag:

```
navData.set_flag = 1;   // to avoid debouncing
```

Then, we need to wait a certain time until set the waypoint (let say 0.5 sec). To do that we read a reference time first:

```
navData.navSetWPtimer = timerMicros();
```

then we wait 0.5s before recording the WP. Once fulfilled we launch the actual function to record a new waypoint:

```
navSetWpCurrent(navData.wp_index);
```

we make a very short beep (to minimize the impact in the main loop)

```
signalingBeep(2000, 50);
```

we increase the WP index (starts in "0") and control that is lower than the maximum number of wp´s:

```
navData.wp_index++;   // increase waypoint index
```

we reset the flag (= …"*Hey I´m ready to record another WP*")

```
navData.set_flag = 0;
```

we reset the timer and send a message

```
navData.navSetWPtimer = 0;
AQ_NOTICE ("waypoint set");
```

The function **`navSetWpCurrent`** assign the actual GPS information to new WP.

Deleting all WP´s

To delete all WP´s we need to launch the function:
`navClearWaypoints();`

This is done ONLY when, rec switch (ch5) is “off” and AQ “DISARMED”. It sounds 3 beeps and a message acknowledge.

Changes in “nav.h”

“nav.h” simply defines the new variables and functions used:

```
      uint8_t wp_index;             //afd WLR index
      uint8_t set_flag;             //afd WLR flag
      uint32_t navSetWPtimer;       //afd WLR timer

extern void navSetWpCurrent(int k);  //afd
```

Annex 1: full code changes over r220

Nav.c, line 82 add:

```
// afd
void navSetWpCurrent(int k) {
      navData.missionLegs[k].type = NAV_LEG_GOTO;
      navData.missionLegs[k].relativeAlt = 0;
      navData.missionLegs[k].targetAlt = UKF_ALTITUDE +
      navUkfData.presAltOffset;
      navData.missionLegs[k].targetRadius = 1.0f;
      navData.missionLegs[k].targetLat = gpsData.lat;
      navData.missionLegs[k].targetLon = gpsData.lon;
      navData.missionLegs[k].maxHorizSpeed = 2.0f; //  m/s
      navData.missionLegs[k].poiHeading = -0.0f; // relative
      AQ_NOTICE("point set\n");

}
```

Nav.c, line 390 add:

```
// afd  WLR = WPs Live Recording using Ch5, RADIO_GEAR
// waypoint recording : gear switch down 1s = record waypoint
if (RADIO_GEAR > 250) {
navData.set_flag = 1;         // set waypoint on switch low to high(debouncing)
navData.navSetWPtimer = timerMicros();
}
// only record when switch is held for 1 second.
if ( ( navData.set_flag) && (( timerMicros() - navData.navSetWPtimer) > (5e5) ) ){
   navSetWpCurrent(navData.wp_index);
   signalingBeep(2000, 50);
   navData.wp_index++;   // increase waypoint index
   navData.wp_index = constrainInt(navData.wp_index,0, NAV_MAX_MISSION_LEGS);
   navData.set_flag = 0;
   navData.navSetWPtimer = 0;
   AQ_NOTICE ("waypoint set");
}

// If waypoint "rec switch is low" AND "AQ not armed" AND "roll is held left" then erase waypoints...
if ( (!navData.set_flag) &&  (RADIO_ROLL < -550) && (supervisorData.state == STATE_DISARMED) ){
navClearWaypoints();
signalingBeep(2000, 50);
delay(200);
signalingBeep(2000, 50);
delay(200);
signalingBeep(2000, 50);
AQ_NOTICE ("waypoints deleted");
      navData.set_flag = 0;
      navData.wp_index = 0;
      }
```

Nav.h, line 127

```
      uint8_t wp_index;            //afd WLR index
      uint8_t set_flag;            //afd WLR flag
      uint32_t navSetWPtimer;      //afd WLR timer
```

Nav.h, line 149

```
extern void navSetWpCurrent(int k);  //afd
```

Annex 2. Other concepts

Switch Debouncing

“Switch debouncing” is one of those things you generally have to live with when playing with switches and digital circuits. If you want to input a manual switch signal into a digital circuit you'll need to debounce the signal so a single press doesn't appear like multiple presses.