Preamble

This is the first layout of how missionplanning could be done on „Harakiri“ using mavlink and missionplanner (MP). First step – protocol implementation, wp validation/saving/loading, autosense between protocols (mavlink/mwii). Mwii wp protocol is currently not in my focus.
Current code in repo https://github.com/Crashpilot1000?tab=repositories .

- With missionplanner/mavlink you can edit and store currently 146WP without external eeprom on naze. May size it down later if space becomes an issue.
- WP Saving/Loading via mavlink is only possible with disarmed copter, so playing around with the WP list when airborne is not possible - for safety.
- There are timeouts (ca. 1sec) between reading and writing to naze with MP.
So it's useless to go berserk on the read/write buttons. Naze will send an error-ack back to the gui or do no ack (MP does a timeout error then).
- The WP inputstream is watched by naze (like the expected WP number so no wp is lost, skipped or double saved. Out of memory etc), otherwise errors could occure with big wp lists.
If an error is detected (list too big, wp sequence wrong, copter armed etc), an error acknowledgement(ack) is send back to GUI.
 A WP list just consisting of an "homepoint" is considered as useless and is not accepted by naze ("error") - it has to have at least one valid WP (like a ROI alone will also not do it, since it's no WP). A faulty WP has an unknown command (see supported commands below) or has missing LAT/LON coordinates (if required) and is skipped during save.
 A WP "list" with only one invalid WP is pseudo - accepted but will actually delete your current wp list on naze (alternative cli: "wpflush").
- Every entered hight in a mission is relative to the ground level. If you set absolute hights (above sea level) with MP it will be ignored and treated as relative values.
- Don't reference to http://copter.ardupilot.com/wiki/planning-an-apmcopter-mission-with-waypoints-and-events/, since this is not Arducopter - some stuff handled differently or is not implemented etc.
-You can/should check what naze makes out of your mission by hitting reload on missionplaner so it will reload the WP like stored in naze. Some values may be altered upon saving, if needed (see below).
For whatever mavlink reason it stores the GPS coordinates not in the precise "int32" manner but in normal "float". That means you will see little rounding errors in the GPS coordinates (compare save/load) that is not a fault in the code, but the nature of the mavlink protocol.
- Instead of bothering with headings it is easier to define a ROI and let the copter „nose in“ on it, if needed. So for simplicity the copter will always face the target if not told to watch a ROI.
- These are the commands that are currently under developement and hopefully get working one by one:

Nomenclature:
The parameters will be called param1..param4 from left to right, so param4 will be yaw or some heading or something else.
Latitude will be called LAT and Longitude will be called LON. LAT/LON are only mentioned when they are ignored by the command or something special going on with them or I just want to mention them..
 Altitude is measured in meters and will be called ALT. ALT is only mentioned if there is something special about it.
Clockwise and counter clockwise will be called CW and CCW. Region of interest is called ROI.
Missionplanner will be called MP. "MP Requester with some errormessage" will be called ERR.

MAV_CMD_NAV_WAYPOINT:

Name in MP drop down menue: "WAYPOINT"
What does it do:
Trivial, it sets a waypoint, what did you think? BUT there is some non trivial stuff going on here:
param1 (aka "Delay") will decide upon the speed the copter does that wp.
param1 = 0 The copter will NOT brake before that wp and look for the next command (that may or may not cause him to brake)
param1 = 1 The copter will slow down when approaching the WP (that is defined in cli by "nav_approachdiv" and nav_speed_min/max of course)
Make sure that the corners are not too sharp, in doubt use param1 = 1. Higher values than 1 are taken as non zero and saved as "1".

param2 (aka "Hit Rad") Is the hitradius in meter. You can not define a smaller hitradius than preset in cli by "gps_wp_radius".
Note: gps_wp_radius is in cm and param2 is in meter, don't confuse that. Maybe I should redo gps_wp_radius in meter as well maybe later.
If you don't enter a value, naze will fill in gps_wp_radius.

param3: Not used.

param4 (aka "Yaw Ang") will not be accepted as a real yaw angle. Normally the copter will look directly (nose in) to the waypoint BUT:
param4 = 1 If a ROI is known it will nose in on the ROI point. To use that define a ROI BEFORE defining a WP. (see ROI)
param4 = 0 ROI is ignored, copter nose points to the next WP.
If param4 = 1 is used and no ROI defined, it will be saved as "0".

Valid LAT/LON needed, otherwise the command is skipped upon save.

ALT in meter will set a new targethight. That hight can be negative, if you want to fly down a canyon, loose GPS signal and crash.
Note: An ALT of 0 m is not tolerated and will be corrected to 2m upon save. To set a hight of 1 m just enter "1" but "0" is nogo, landing is a different command (see there).
Reaching the correct altitude (or hightchange between waypoints) is of low priority - that means a Waypoint is considered as done when the LAT/LON position (within the radius) is reached.
To ensure a correct hight (if needed) enter a loiterpoint somewhere with the wanted hight and a sufficient time (see there).

MAV_CMD_NAV_LOITER_UNLIM

Name in MP drop down menue: "LOITER_UNLIM"
What does it do:
Loiters (almost) for ever. It can only be the last command in a WP list, otherwise the WP list is not accepted (ERR).
It is treated as a waypoint that will be approached with braking. A hitradius must be defined or will be preset upon save.

param2 Is the hitradius in meter. You can not define a smaller hitradius than preset in cli by "gps_wp_radius".

param4 = 1 If a ROI is known it will nose in on the ROI point.
param4 = 0 No special yaw. Nose still points in the former flightpath.

Valid LAT/LON needed, otherwise the command is skipped upon save.

MAV_CMD_NAV_LOITER_TURNS

Name in MP drop down menue: "LOITER_TURNS"
What does it do:
Loiters and yaws/rotates copter.
It is treated as a waypoint that will be approached with braking. A hitradius must be defined or will be preset upon save.

param1 (aka "Turns") describes the number of turns done before proceeding. A positive number will cause CW turns, a negative number CCW turn.
Maximal values: -127..+127. param1 = 0 is not tolerated and is saved as "1" to ensure at least one CW turn.
I am still unsure how to implement the actual yaw speed, however it will be gps_yaw dependant.
I guess I will end up with param3 or so setting the time in seconds for one complete turn. Let's see when it comes to this.

param2 Is the hitradius in meter. You can not define a smaller hitradius than preset in cli by "gps_wp_radius".

Valid LAT/LON needed, otherwise the command is skipped upon save.

MAV_CMD_NAV_LOITER_TIME

Name in MP drop down menue: "LOITER_TIME"
What does it do:
Loiter some time.
It is treated as a waypoint that will be approached with braking. A hitradius must be defined or will be preset upon save.

param1 (aka "Time s") Defines the time in seconds. Max 255 sec.
"0" is not tolerated and is saved as "1" to ensure at least one second loiter.

param2 Is the hitradius in meter. You can not define a smaller hitradius than preset in cli by "gps_wp_radius".

param4 (aka "yaw per") Will not be accepted as a real yaw angle.
param4 = 1 If a ROI is known it will nose in on the ROI point. To use that define a ROI BEFORE. (see ROI)
param4 = 0 ROI is ignored, copter nose points to the next WP.

Valid LAT/LON needed, otherwise the command is skipped upon save.

MAV_CMD_NAV_RETURN_TO_LAUNCH

Name in MP drop down menue: "RETURN_TO_LAUNCH"
What does it do:
It will just (nose in return) to the physical launch place (not the MP "homepos"). Landing is a different command (see there).
LAT/LON are ignored (and saved as "0"), the in flight LAT/LON is taken.
Don't forget ALT value.
This command can not stand alone (ERR). That means at least one other WP has to be in the list.

MAV_CMD_NAV_LAND

Name in MP drop down menue: "LAND"
What does it do:
It will land at the current position. No other parameter accepted/used.
It can only be the last command in a WP list (ERR).
This command can not stand alone (ERR). That means at least one other WP has to be in the list.

MAV_CMD_NAV_TAKEOFF

Name in MP drop down menue: "TAKEOFF"
What does it do:
It does a takeoff at the current position, defined by cli parameters: as_lnchr, as_clmbr, as_stdev and esc_nfly.
ALT is the targethight of the launch, no matter what is defined in cli as_trgt.
ALT must be in the range: 2..15m, other values will be constrained to that range.
It can only be the first command (even a "ROI" has to come later) in a WP list (ERR).
This command can not stand alone (ERR). That means at least one other WP has to be in the list.
If the mission is engaged when the copter is already airborne the Takeoff command is skipped for obvious reason, but ALT is taken as minimal hight.
That means if the mission was engaged airborne but below the specified Takeoff ALT, a climb will be initiated.
If engaged above that hight it will approach the mission "homepos" at current hight and alter it according to further request.

MAV_CMD_NAV_ROI

Name in MP drop down menue: "ROI"
What does it do:
It defines a ROI for MAV_CMD_NAV_WAYPOINT, MAV_CMD_NAV_LOITER_UNLIM, MAV_CMD_NAV_LOITER_TIME.
A ROI is just defined by LAT/LON. ALT is not needed since we have no gimbal stuff going on.
ALT value will be set to 0.
In opposite to arducopter you don't have to define a ROI each time before calling an dependent function (like listed above).
Just make sure a ROI is defined before those functions are called (see there). If you set those functions to use a ROI but ROI isn't defined, they will be set to "don't use ROI" upon save.
You can define multiple ROIs in a WP list but only the previously defined is used for the current (and dependent) entry.
A WP list with no WP and just a ROI are not saved (ERR). ROI can not be the last command in the WP list (ERR).