

Waypoints 1.0 Installation, Setup and Operation for MatrixNav and AileronAssist

Waypoint firmware is now available with MatrixNav version 1.8, and will be available for AileronAssist in a few weeks, with the following features:

- Waypoints are 3D. You specify location of the points relative to the initialization location of the board.
- You have the option of using either cross-track error navigation, or using navigation toward the target waypoint.
- Arrival at a waypoint is based on the change in sign of the projection of the vector from the location to the target onto the vector between the waypoints. This produces a reliable assessment of arrival, without any chance of loitering.
- The primary source of steering is the direction cosine matrix, so steering continues reliably and smoothly even when the GPS loses lock during banked turns.

This document covers only the waypoint function of MatrixNav and AileronAssist. For installation, setup, and operation of all other functions, refer to the other instructions that will accompany each release of the firmware.

The waypoint function replaces the commanded return-to-launch function in previous versions of MatrixNav and AileronAssist. The loss-of-signal return-to-launch function remains, but since most pilots use a radio with a fail-safe function, for all intents and purposes, waypoints completely replaces return-to-launch. However, since the autonomous command causes the waypoint function to immediately steer toward the first waypoint, you can use the waypoint function as RTL by careful selection of the first waypoint. In fact, if you select a single point waypoint with X and Y relative coordinates of 0, engaging the waypoint function will be exactly equivalent to the return to launch function.

The waypoint function will work in a 3D mode with altitude hold controlling height, if it is enabled. If you do not enable altitude hold, the waypoint function will work in a 2D mode, steering and stabilizing the plane, but leaving throttle and altitude control to the pilot.

Specifying waypoints

The waypoints are defined in the file waypoints.h that is a header file in the project source code for the firmware. There are two items you need to specify in the file:

- whether or not you want to use cross-tracking
- the coordinates of the points.

If you want to use cross-tracking, define the symbol CROSSTRACKING in the waypoints.h file:

```
#define CROSSTRACKING
```

If you select cross tracking, navigation will be based on the perpendicular distance from the aircraft to the line between the waypoints that define the course leg. The steering angle toward the course leg is proportional to the cross track distance, up to 45 degrees at a distance of 32 meters. Beyond 32 meters, the steering angle is 45 degrees. If you select cross tracking, arrival at waypoint will be declared when the plane crosses an imaginary finish line through the target waypoint, perpendicular to the course leg.

The definition of CROSSTRACKING can appear anywhere in the file waypoints.h. If you do not want to use cross-tracking either delete the line, or comment it out:

```
//#define CROSSTRACKING
```

If you do not define CROSSTRACKING, then steering will be toward the target waypoint of the course leg. Arrival at the waypoint will be declared if either the cross track arrival condition is satisfied, or if the aircraft approaches closer than 25 meters to the target waypoint.

The waypoints themselves are specified in the file waypoints.h using a C declaration of a constant array. Here is a simple example that I use for testing and debugging, with two points:

```
const struct relative3D waypoints[] = {
    { 0 , 0 , 100 } ,
    { 100 , 173 , 100 } ,
} ;
```

Each waypoint is specified as the three coordinates, measured in meters, relative to the initialization location of the board. The C syntax allows an optional comma after the last waypoint in the list. The valid range for each of the coordinates is plus or minus 32767 meters, though in the US you are supposed to maintain visual contact at all times.

The first coordinate (X) is measured toward the east from the initialization point. The second coordinate (Y) is measured toward the north. The third coordinate (Z) is the height above the initialization point.

There is no need for you to indicate how many waypoints you have specified, the compiler will count them for you.

In case you are interested, the waypoints are defined as an array named waypoints, of structures of type relative3D, defined elsewhere in the firmware. It is a constant array, so the compiler will store it in program memory. There is plenty of unused program space, so you can define on the order of 1000 waypoints.

You do not have to worry about the first and last line of the definition, it will never change. All you have to do is fill in the points. In this simple example,

there are only two points. The first point is located 100 meters above the initialization point. The second point is located 200 meters from the first point, on a bearing 30 degrees clockwise from the north.

Here is another example in which the waypoint function is used simply for an RTL function, with a target height of 50 meters:

```
const struct relative3D waypoints[] = {{ 0 , 0 , 50 }} ;
```

Activation of the waypoint function is achieved the same way as the return to launch function that it replaces. There are three control modes, manual, stabilized, and waypoint navigation, that are selected by a joystick or a three position switch.

The aircraft flies to each waypoint in the list in sequence. It uses whichever strategy you have selected to navigate between points, and uses the height of the waypoint as input to the altitude hold function. If you are not using altitude hold, you still need to specify all three coordinates of the waypoints anyway, but the height coordinates will be ignored. When waypoint processing finishes the list, it starts over.

When the throttle stick is placed in the full off position, the motor will turn off, no matter what control mode is selected. This useful when you want to pitch down to lose altitude quickly and you do not want altitude hold to turn the motor on. It is also useful if you want to use stabilization or return to launch in a sailplane while it is sailing.

When the waypoint function is activated, the aircraft flies to the first waypoint along a line between its position at the time that the waypoints function is activated, and the first waypoint. If you disengage the waypoint function at any time and then reengage it, it will start over with the first waypoint on the list. So, if you decide you want to return the plane to the first point at any time, simply disengage the waypoint function, wait a couple of seconds, and reengage it.

The coordinates of the waypoints must be constant C expressions. That means that you can specify them simply as numbers, or you can use expressions to simplify the specification of the waypoints. For example, the following is a valid specification of the “butterfly” course used in one of the diydrones T3 contests. CLEARANCE is used to raise the entire pattern a fixed amount, such as might be done to clear obstacles such as trees:

```
#define CROSSTRACKING
#define CORNER 100
#define CLEARANCE 25

const struct relative3D waypoints[] = {

    {    CORNER    ,    CORNER    , CLEARANCE + 100 } ,
    {    CORNER    , - CORNER    , CLEARANCE + 75 } ,
    { - CORNER    ,    CORNER    , CLEARANCE + 50 } ,
```

```

{ - CORNER , - CORNER , CLEARANCE + 25 } ,
{ CORNER , CORNER , CLEARANCE + 50 } ,
{ CORNER , - CORNER , CLEARANCE + 75 } ,
{ - CORNER , CORNER , CLEARANCE + 100 } ,
{ - CORNER , - CORNER , CLEARANCE + 75 } ,
{ CORNER , CORNER , CLEARANCE + 50 } ,
{ CORNER , - CORNER , CLEARANCE + 25 } ,
{ - CORNER , CORNER , CLEARANCE + 50 } ,
{ - CORNER , - CORNER , CLEARANCE + 75 } ,
} ;

```

Tuning the gains

There are no gains directly associated with the waypoint function that you need to tune. What you need to tune are all of the gains associated with yaw, pitch, roll, and altitude control, described in the instruction manuals for MatrixNav and AileronAssist. What you should do is adjust those gains for best performance, using a single waypoint for a return to launch function while you are tuning the gains. Then when you have the gains that you want, you enter a list of waypoints. Start with something simple, and work your way up.

Ground checkout

Before you actually use the waypoint function in flight, I suggest that you test it on the ground with a walk-around. Here are a few tips for doing that:

- Temporarily set any rate feedback gains to zero, so that you do not have to hold the plane steady.
- Keep the throttle stick off so the motor does not come on. Better yet, disconnect the motor, if that is possible.
- Use short course legs so that you do not have to walk too far.

Fail safe

Both MatrixNav and AileronAssist have return to launch with the motor off as the fail safe response to loss of signal. However, most radios have their own fail safe response. In that case, the behavior of MatrixNav and AileronAssist will depend on how you set up the radio. In order to make the best use of the capabilities of the waypoints function, it is suggested that you set up the radio fail safe as follows:

- Set the rudder, ailerons, and elevator for center position.
- Set the control mode selection channel to select the waypoints function.
- Set the throttle however you want. If you are not using altitude hold, the motor will respond accordingly. If you are using altitude hold, If you set the throttle for full off, the aircraft will follow the waypoints

with the throttle off. If you set the throttle for anything else, the aircraft will continue to follow the waypoints, including the height coordinates.

Operation

Operation is exactly the same as for previous versions of MatrixNav and AileronAssist. Refer to the operation instructions for MatrixNav or AileronAssist. The only difference is that the third control mode is a waypoint function instead of a return to launch function. Here are a few tips for waypoint operation:

- It is suggested (though not required) to set the X and Y coordinates of the first waypoint to zero, and set all of the other waypoints with respect to that point. That way you can line the pattern up with the initialization point and you can shift the pattern around a bit to move the waypoints upwind on a windy day.
- Before launch, check stabilized mode and waypoint mode. In both modes, the control surfaces should respond to pitch, yaw, and roll motions of the aircraft. If you are using altitude hold, check to see that the motor comes on when you engage the throttle stick.
- You can launch in stabilized mode if you want. You can land in stabilized mode, but remember to turn the motor off before you land.
- Do not launch in waypoint mode. The IMU needs the aircraft to be flying to achieve yaw lock. It takes a few minutes for that to happen, so wait at least 2 minutes after launch to engage the waypoint mode.