G-NAV

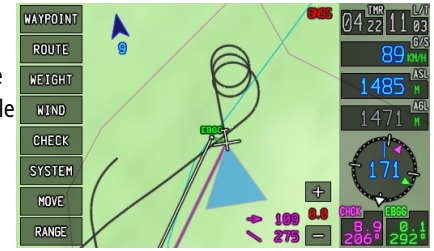
github.com/GuillermoHazebrouck/gnav

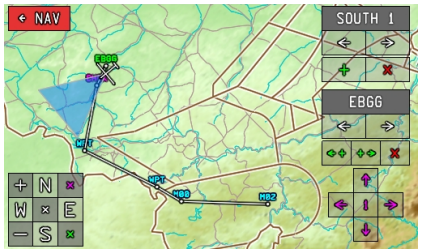
The free gliding computer

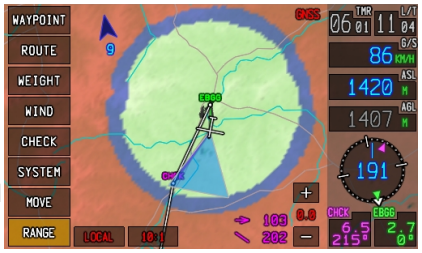
G-NAV is an open source gliding computer designed to help you plan your flight, avoid incursion inside unintended airspace and fly at best performance. All of this using only GNSS data, free software and free data.

Design philosophy

There are many gliding computers on the market. What makes G-NAV special is the usage of a strong-typed programming language (used and tested in professional ATM and avionics applications) and its simplicity of use and design. G-NAV is designed under the premise of providing maximum navigation aid with as little input from the pilot as possible, so that you can collect useful flight data in the blink of an eye and keep looking outside.







Navigation features

Moving map. A moving map follows the glider and provides navigation information. The map is able to display the topography, geographical references, route, vectors. The north is always kept above (no rotation possible) and a rotating quadrant is provided for guidance to help you leave the thermal in the right direction.
Route. The software has a route edition page to create or adapt many flying routes. Changing or adapting the route is relatively easy and can be done directly from the touchscreen.
Vectoring. The navigation screen continuously display vectors to the active way point and the home location.
Range computation. The directional range is represented by direct adaptation of the terrain colors, so that the vertical situation can be evaluated in the blink of an eye. There are three range modes: *straight line*, *local at optimum speed* and *local within a 10:1 cone*. In all cases the wind, expected sink (Mac Cready setup) and aircraft mass and performance are taken into account.
Performance navigation. Beside the range cone, the optimal speed an expected descent for the active way point are always visible in the navigation screen.
Wind. The software counts with an automatic wind computation algorithm, but the input can be switched to manual at own discretion.
Data quality and status awareness. The software will not display data that is no longer relevant or valid, so that you are not mislead and can directly fall back and focus in a different navigation method in case of failure.
Static data. The software provides static data like QNH and radio frequency tables.

Data sources

Topography. G-NAV can load a topographic chart of an extended region bound within two parallels and two meridians. This information is used for the navigation when computing the AGL and range. The terrain data is loaded usig an ESRI file in ASCII format. This data is available for free in the world wide web.
Geographic data. G-NAV can load geographic data like rivers, roads, railways and country borders from standard *shape* files and display it on the navigation screen.
Airspace data. G-NAV can load airspace sectors from ASCII files. These are displayed on the navigation screen. The program does not generate alerts based on this data.
Glider data. G-NAV can load glider performance data from ASCII files. This is then used to compute the range, best gliding speed and expected descent on the route legs.
Streaming. The data can flow from a serial port (UART), UDP or a file (in replay mode). Two data formats are allowed: NMEA and G-NAV. Note that the UDP streaming allows the software to be driven by flight simulators.

Software architecture and environment

G-NAV works best on UNIX-like systems. It is developed in Linux and intended to run on Raspberry Pi 4 (ARM computer) under its standard OS. Other hardware can also be used.
G-NAV is entirely written in Ada and it is compiled using Gnat GPL. Because all widgets and fonts are part of the project, it only links to the GLFW and GL-ES libraries, making it a very compact and stable.

How to try it

The best way to know if this is something for you is to try it in a simulation environment. Flight Gear has been tested and although it does not offer the best flying experience, it is largely sufficient to evaluate the software. Information about how to setup the simulation can be found on the github repository.