Project limitations

Due to the constraints of the test environment, I took the following steps:

1. I wrote a small flight control system (FCS) simulator in Go-lang. It can simulate interaction with external devices (sensors) and simply respond to various HTTP requests. This was done solely for demonstration purposes. In a real-world scenario, an actual type of connection would be used, and separate modules for interfacing and communication would be created.
2. Testing. I had to slightly "litter" the project structure to demonstrate my proficiency in working with both Robot Framework and Pytest. In a real project, a cleaner and more understandable structure would be used. Specifically, a dedicated directory for tests would be created (instead of two, as is currently the case, see README.md), along with separate directories for artifacts, test data files, constants, and variables.
3. I would have preferred to use a Docker container to run the FCS binary, but since this is just a demonstration, I decided against it.
4. For ease of execution, two scripts were created that will do everything for the user: build the binary, create a virtual environment, install dependencies, run the tests, and deactivate the virtual environment. The files can be found in the project root.
5. This project was tested on CI/CD: GitHub Actions is set up and ready to run on the Pipeline.