SYSTEM REQUIREMENT DOCUMENT

# GIỚI THIỆU VÀ MỤC ĐÍCH

Tài liệu này mô tả chức năng và nhiệm vụ cho bộ điều khiển bao gồm nhiệm vụ cho 2 chip STM32F103 và chip STM32H743.

Phạm vi nghiên cứu thiết kế hệ thống điều khiển UAV có độ ổn định cao, khả năng bay theo chế độ điều khiển bằng tay, khả năng bay theo chế độ tự động và giao tiếp giữa trạm mặt đất với UAV.

# Yêu cầu chức năng

## Vi điều khiển STM32F103

R1.1: Đọc dữ liệu gyro and accelerometer thô từ cảm biến MPU6050 và scale về giá trị chuẩn

R1.2: Đọc dữ liệu magnetometer thô từ cảm biến BMM350

R1.3: Đo khoảng cách từ cảm biến HCSR04 (Ultrasonic Range Finde)

R1.4: Đọc dữ liệu GPS từ cảm biến NEO-6

R1.5: Đọc dữ liệu từ cảm biến đo nguồn điện

R1.6: Điều chế xung cho đông cơ.

R1.7: Đóng gói và giao tiếp bản tin với vđk STM32H743.

## Vi điều khiển STM32H743 (chip chính AP)

R1.8: Giao tiếp với chip STM32F1

R1.9: Truyền và nhận dữ liệu với trạm mặt đất

R1.10: Nhận tín hiệu từ tay điều khiển

R1.11: Tích hợp thuật toán lọc nhiễu (EKF)

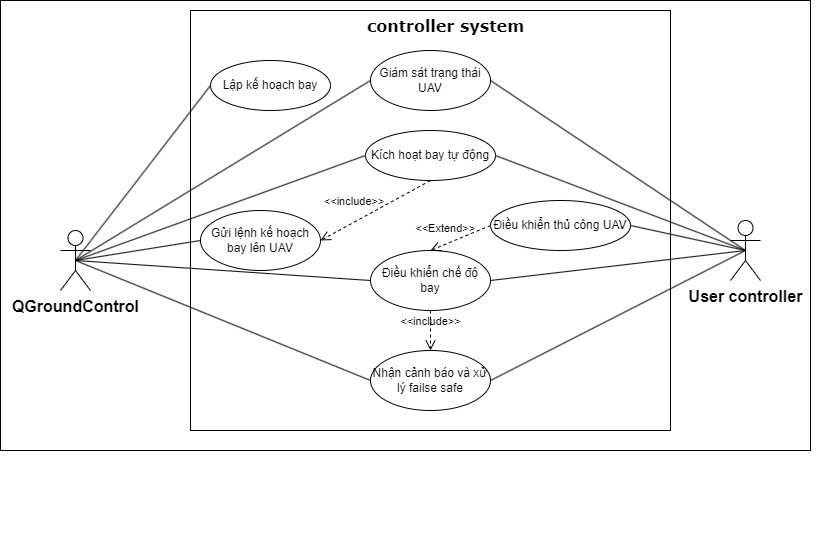
R1.12: Định vị vị trí UAV (Đưa ra được trạng thái của UAV, thiết kế AHRS)

R1.13: Thuật toán tự cân bằng cho UAV

R1.14: Thuật toán điều khiển động cơ

R1.15: Thuật toán bay UAV theo chế độ thủ công hoặc tự động

R1.16: Thuật toán xử lý false safe.



Mô tả hoạt động của hệ thống

# Yêu cầu phi chức năng

## Hiệu suất

R2.1: Tần số gửi bản tin giám sát về trạm mặt đất tối thiểu 25hz

R2.2: xử lý dữ liệu theo thời gian thực

R2.3: Tốc độ lấy mẫu cảm biến 100khZ cho IMU, magnetometer, Barometer

## Độ tin cậy và an toàn

R2.4: Hệ thống ổn định trong các điều kiện bay khác nhau

R2.5: Cơ chế reset khi treo phần mềm (một vòng điều khiển hoặc 1 task FreeRTos bị kẹt)

R2.6: Cơ chế reset khi cảm biến MPU, BMM350 không trả dữ liệu trong 0.012ms

R2.7: Tắt động cơ và vào chế độ safe sau khi reset

R2.8: Cơ chế giữ vị trí khi mất tín hiệu điều khiển

R2.9: Có chế tự quay về khi mất tín hiệu điều khiển quá 30s (RTL)

R2.10: Cơ chế tự quay về khi pin yếu trong chế độ bay tự động (RTL)

## Tính bảo trì

R2.7: Mã nguồn dễ đọc, dễ mở rộng và sửa lỗi

R2.8: Cấu trúc module rõ ràng (mỗi module có file cấu thực thi riêng)

R2.9: Log dữ liệu sau mỗi chuyến bay

## Tính thời gian thực

R2.10: Các nhiệm vụ thời gian thực (PID control, xử lý dữ liệu cảm biến, nhận và gửi dữ liệu) đúng hạn

R2.11: Xây dựng hệ điều hành RTOS với bộ lập lịch thích hợp

# Yêu cầu về giao diện

R3.1: Sử dụng QGroundcontrol

R3.2: Hiển thị các thông tin trên QG: tốc độ, GPS, attitude, điện áp, chế độ bay, vị trí…

R3.3: Dữ liệu telemetry truyền về với tốc độ thấp

R3.4: Hiện thị lỗi từ hệ thống: mất GPS, mất RC, pin yếu, vị trí lỗi

R3.5: Truyền hình ảnh từ camera