



TAY CẦM ĐIỀU KHIỂN DRONE NHỎ V1.1



Mục Lục

I. Phần cứng hệ thống	3
1. Ngoại vi	3
a) Màn hình oled hiển thị các giá trị điều khiển	3
b) Joystick	4
2. Nút nhấn, công tắc và LED	4
a) Nút nhấn	4
b) Công tắc gạt	5
c) LED	6
d) Công tắc nguồn	6
e) Các biến trở điều chỉnh cho Joystick	7
3. Các sơ đồ mạch điều khiển	8
II. Chương trình hệ thống	11
1. Dữ liệu điều khiển drone	11
a) Chế độ Manual	11
b) Chế độ Auto 1 điểm	12
c) Chế độ Auto đa điểm	13
2. Chương trình trên Arduino Mega	14
a) Hàm thiết lập setup()	14
b) Vòng lặp loop()	14
c) Các hàm đọc trạng thái	15
d) Gửi dữ liệu tới DRONE	16
3. Chương trình trên ESP32	17
a) Hàm thiết lập setup()	17
b) Vòng lặp xử lý hệ thống khi nhận dữ liệu	17

I. Phần cứng hệ thống

1. Ngoại vi

a) Màn hình oled hiển thị các giá trị điều khiển

- Ở chế độ Position:



- Ở chế độ Angle:



b) Joystick



Joystick bên phải: điều khiển roll \leftrightarrow và pitch \updownarrow .

Joystick bên trái: điều khiển yaw \leftrightarrow và z \updownarrow .

2. Nút nhấn, công tắc và LED



a) Nút nhấn

Gồm 3 nút nhấn:



Nút “**Start**”: Dùng để khởi động drone.

Nút “**Stop**”: Dùng để tắt drone.

Nút “**Menu**”: Dùng để chuyển màn hình oled sang các hiển thị giá trị khác.

b) Công tắc gạt

Gồm 3 công tắc gạt dùng để chuyển trạng thái điều khiển:



- Công tắc “**Auto/Manual**”:
 - + Gạt sang trái: chế độ Manual (bằng tay)
 - + Gạt sang phải: chế độ Auto (tự động)
- Công tắc “**Pos/Angle**”:
 - + Gạt sang trái: chế độ Pos (vị trí)
 - + Gạt sang phải: chế độ Angle (góc)

- Công tắc **“Emer”**: Chế độ Emergency (Dừng khẩn cấp)

+ Gạt lên: Tắt chế độ Emergency

+ Gạt xuống: Bật chế độ Emergency

c) LED

Gồm 4 Led biểu diễn các trạng thái làm việc của tay cầm:



- Led **“RF truyền”**: Đèn sáng khi tín hiệu truyền được tới drone.

- Led **“Control Mode”**: Đèn sáng khi ở chế độ Manual → Pos, tắt ở chế độ Manual → Angle, đèn chớp nháy ở chế độ Auto.

- Led **“Emer”**: Đèn sáng khi bấm nút “Start”, tắt khi bấm nút “Stop” và nhấp nháy khi bật công tắc “Emer”.

- Đèn led **“RF nhận”**: Đèn sáng khi ESP32 trong tay cầm nhận được tín hiệu RF.

d) Công tắc nguồn

Gồm 1 công tắc nguồn: Bật tắt nguồn tay cầm.



e) Các biến trở điều chỉnh cho Joystick

Gồm 4 biến trở:



Pitch_sc: điều chỉnh phạm vi giá trị pitch.

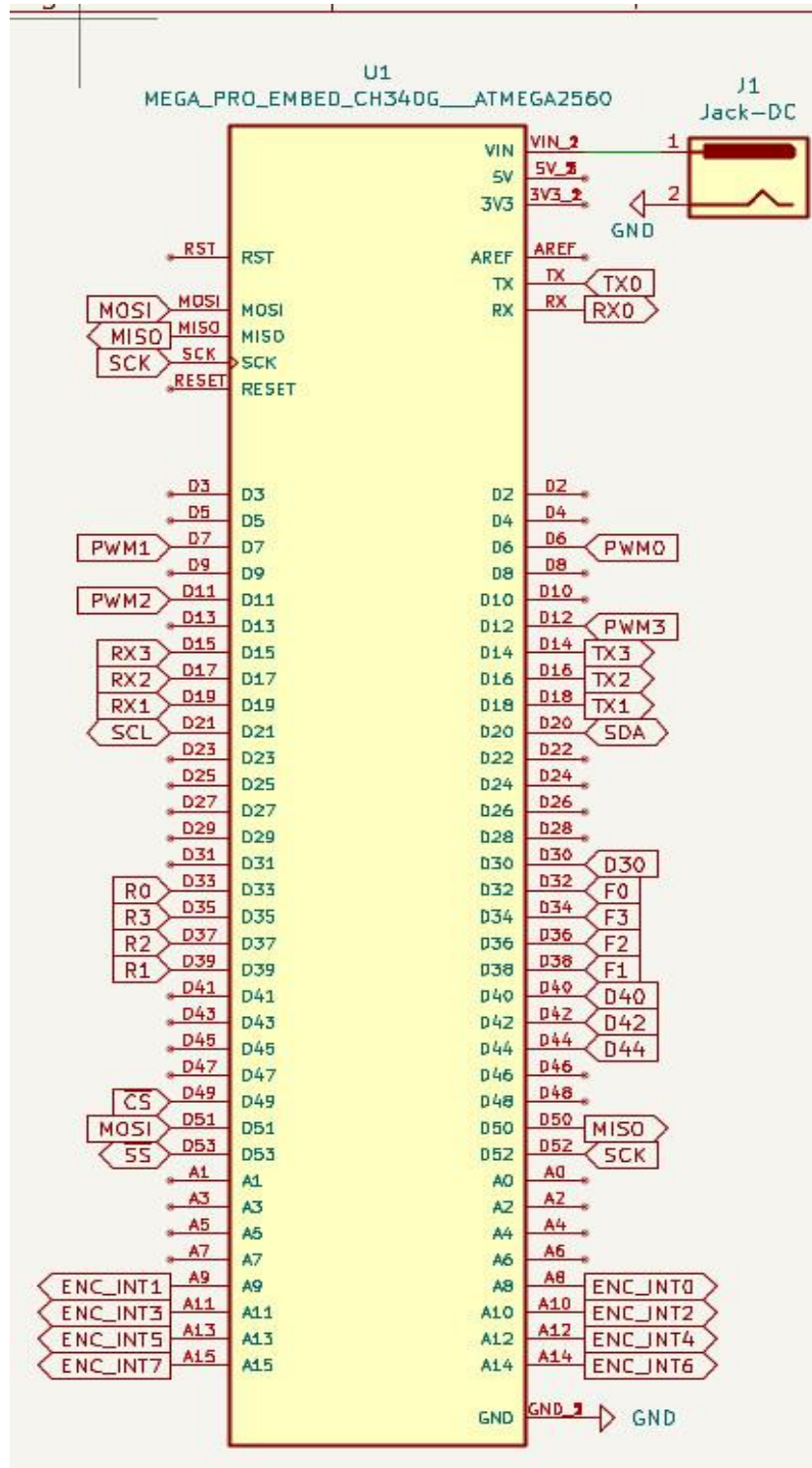
Roll_sc: điều chỉnh phạm vi giá trị roll.

Z_sc: điều chỉnh phạm vi giá trị z.

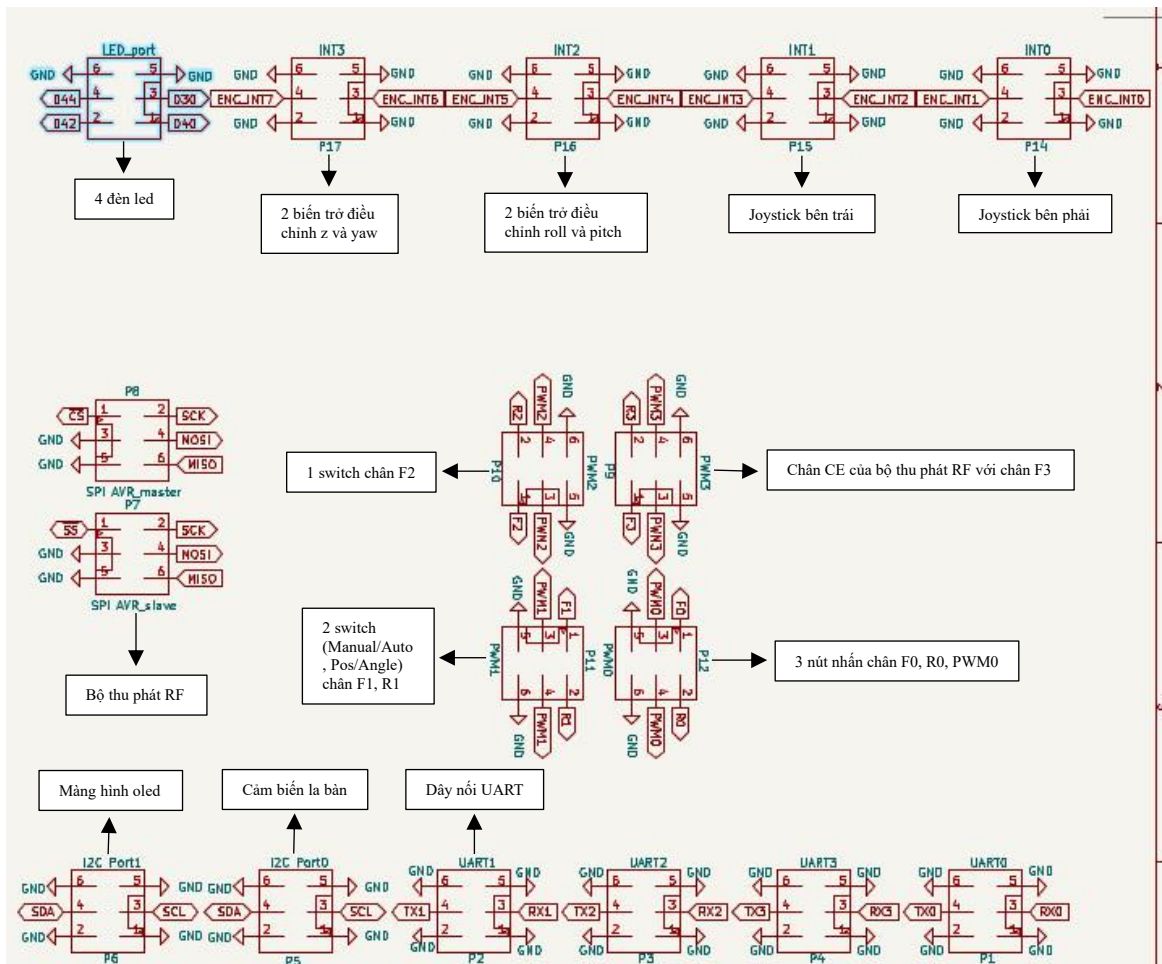
Yaw_sc: điều chỉnh phạm vi giá trị yaw.

3. Các sơ đồ mạch điều khiển

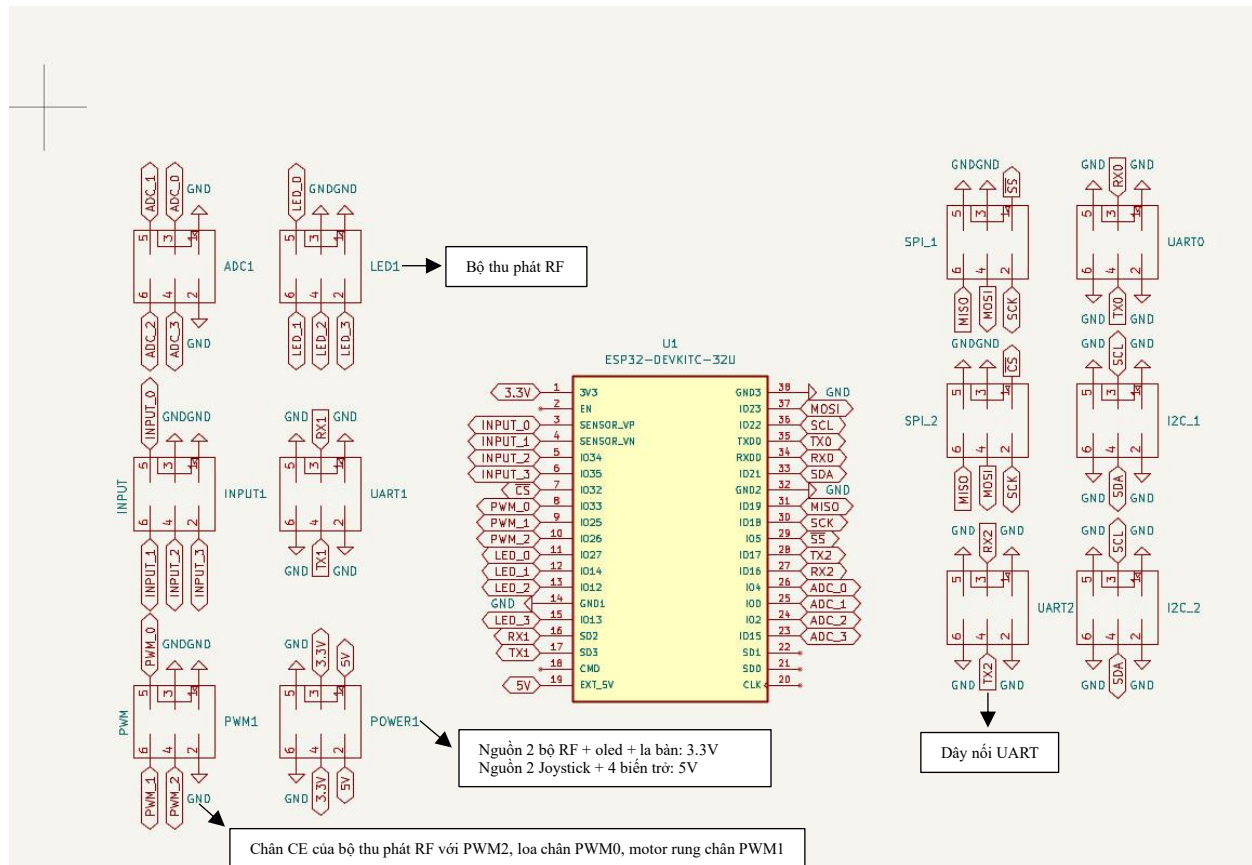
Sơ đồ chân Arduino Mega:



Sơ đồ ra chân và đi dây của module Arduino Mega:



Sơ đồ chân, sơ đồ ra chân và đi dây của ESP32:



II. Chương trình hệ thống

1. Dữ liệu điều khiển drone

Mỗi chuỗi đều gửi 7 long:

Chế độ Manual	[*, 0, state], [Dx], [-Dy], [yaw], [z], [0,0,flow,pump_state], [check_sum]
Chế độ Auto 1 điểm	[*, 0, state], [lat], [long], [z], [time], [0,0,flow,pump_state], [check_sum]
Chế độ Auto đa điểm	[*, id, state], [lat], [long], [z], [time], [len,stt,flow,pump_state], [check_sum]

a) Chế độ Manual

[*, 0, state], [Dx], [-Dy], [yaw], [z], [0, 0, flow, pump_state], [check_sum]
--

*****: byte_start

state = bt_mode*10 + bt_state

bt_mode = 1

bt_state: trạng thái nút nhấn (“Start” = 1, “Stop” = 2, “Emergency” = 8)

Dx: điều chỉnh roll (-138,138)

Dy: điều chỉnh pitch (-138,138)

yaw: điều chỉnh yaw.

z: điều chỉnh z.

flow : ở chế độ manual flow đặt cứng = 6

pump_state: trạng thái nút nhấn bơm (“Start_pump” = 1, “Stop_pump” = 0)

check_sum: kiểm tra tổng các byte của 6 long (24 byte) đầu.

b) Chế độ Auto 1 điểm

<code>[*, 0, state], [lat], [long], [z], [time], [0,0,flow,pump_state], [check_sum]</code>
--

*****: byte_start

state = bt_mode * 10 + bt_state

bt_mode = 2

bt_state: trạng thái nút nhấn (“Start” = 1, “Stop” = 2, “Emergency” = 8)

lat : gửi vị trí vĩ độ

long : gửi vị trí kinh độ

z : gửi độ cao đặt (cm)

time: gửi thời gian bay mong muốn (s)

flow và **pump_state**:

+ Nếu như chọn kèm theo chế độ bơm:

flow : lưu lượng bơm $0 < \text{flow} < 12$ (lit/phút)

pump_state: Khi nhấn nút gửi pump_state tự động lên 1, nút nhấn cứng có thể can thiệp vào pump state

+ Nếu không chọn chế độ bơm:

flow = 0

pump_state = 0

check_sum: kiểm tra tổng các byte của 6 long(24 byte) đầu

c) Chế độ Auto đa điểm

<code>[*, id, state], [lat], [long], [z], [time], [len, stt, flow, pump_state], [check_sum]</code>
--

*****: byte_start

id: id từng điểm lấy từ internet. Với id gồm 2 byte: byte thấp đứng trước.

Vd: 129 [129, 0], 512 [0, 2]

state = bt_mode * 10 + bt_state

bt_mode = 3

bt_state: trạng thái nút nhấn (“Start” = 1, “Stop” = 2, “Emergency” = 8)

lat : gửi danh sách vị trí vĩ độ các điểm

long : gửi danh sách vị trí kinh độ các điểm

z : gửi độ cao đặt (cm)

time: gửi thời gian bay mong muốn, đã chia thời gian theo khoảng cách (s)

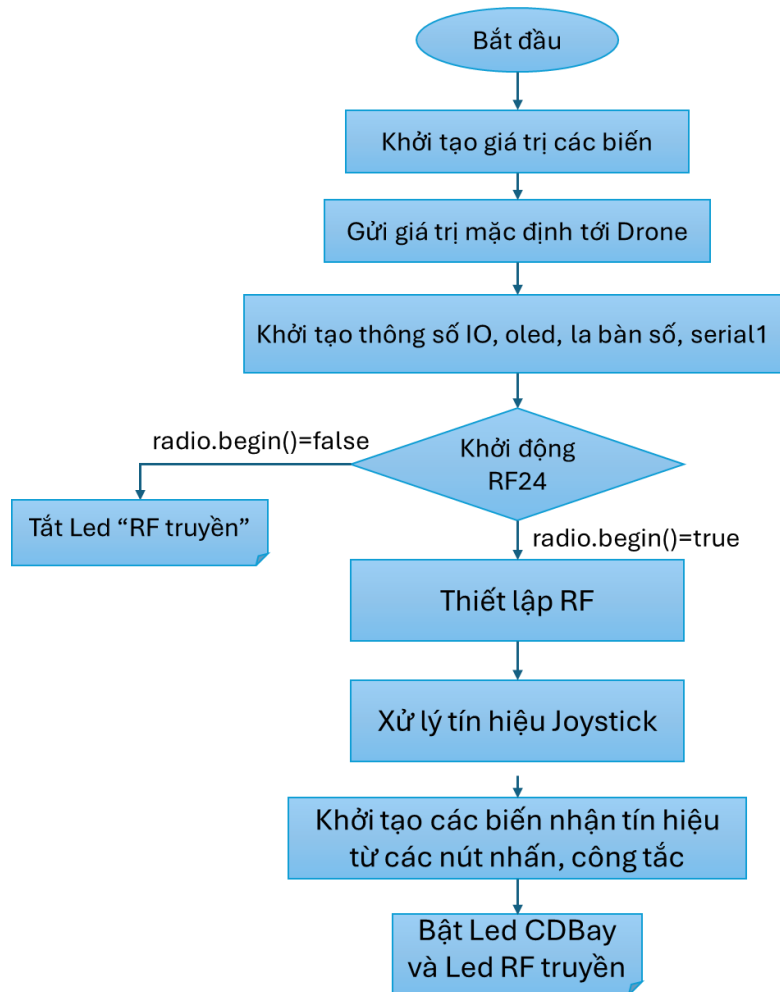
len: gửi số lượng điểm bay

stt: gửi số thứ tự các điểm (bắt đầu từ 0)

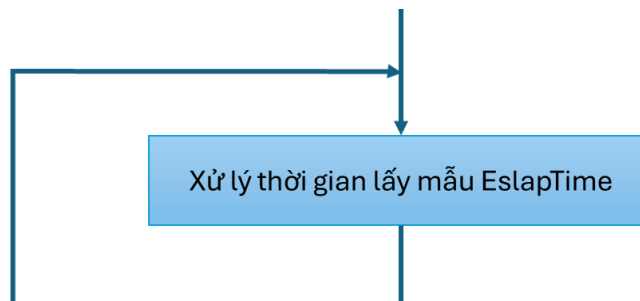
check_sum: kiểm tra tổng các byte của 6 long (24 byte) đầu

2. Chương trình trên Arduino Mega

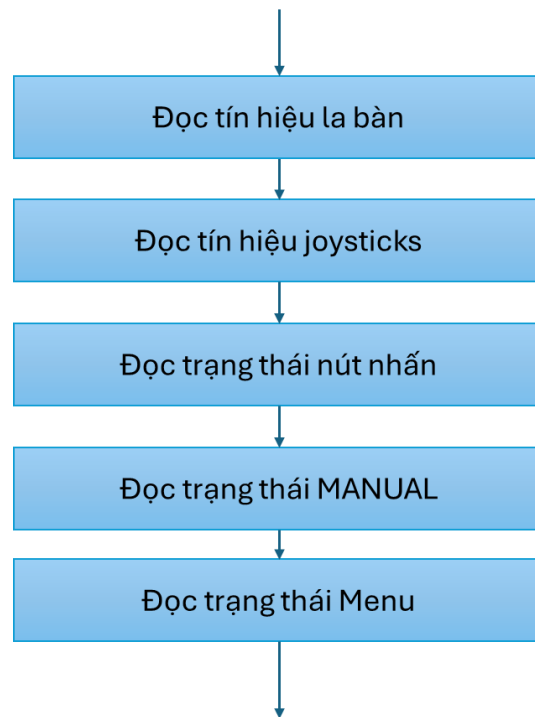
a) Hàm thiết lập setup()



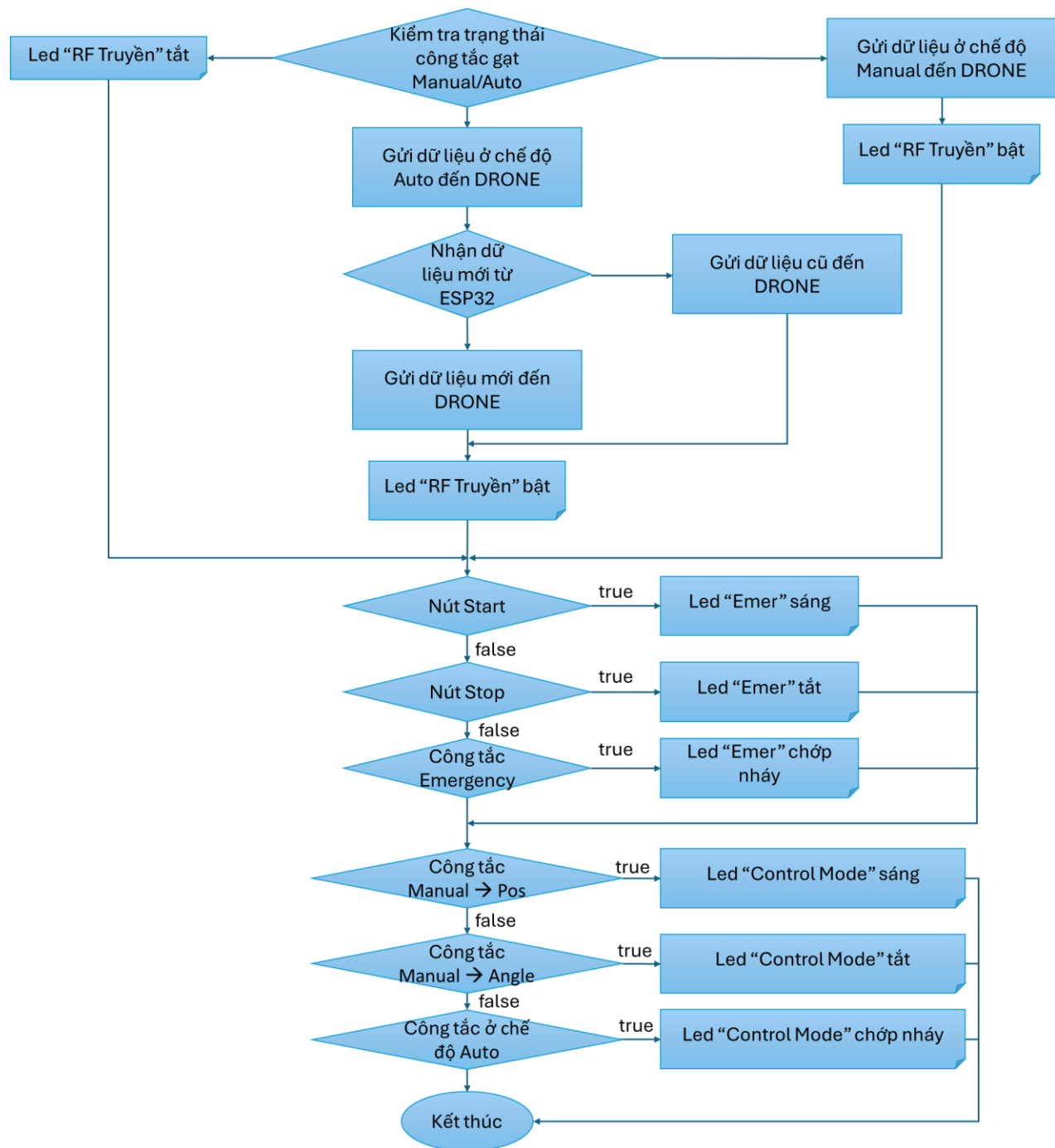
b) Vòng lặp loop()



c) Các hàm đọc trạng thái

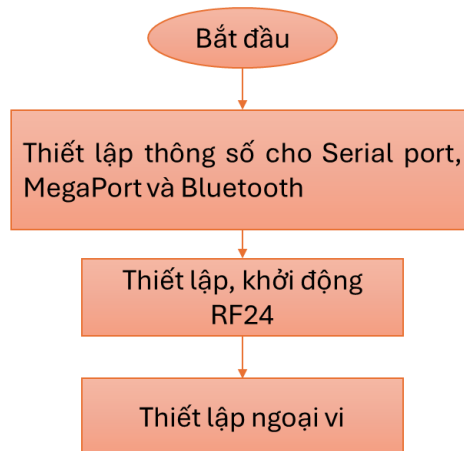


d) Gửi dữ liệu tới DRONE



3. Chương trình trên ESP32

a) Hàm thiết lập setup()



b) Vòng lặp xử lý hệ thống khi nhận dữ liệu

