



# Tổng hợp chương trình điều khiển T500

THỰC HIỆN: HOÀNG VĂN HÒA

## Tugger AGV – T500

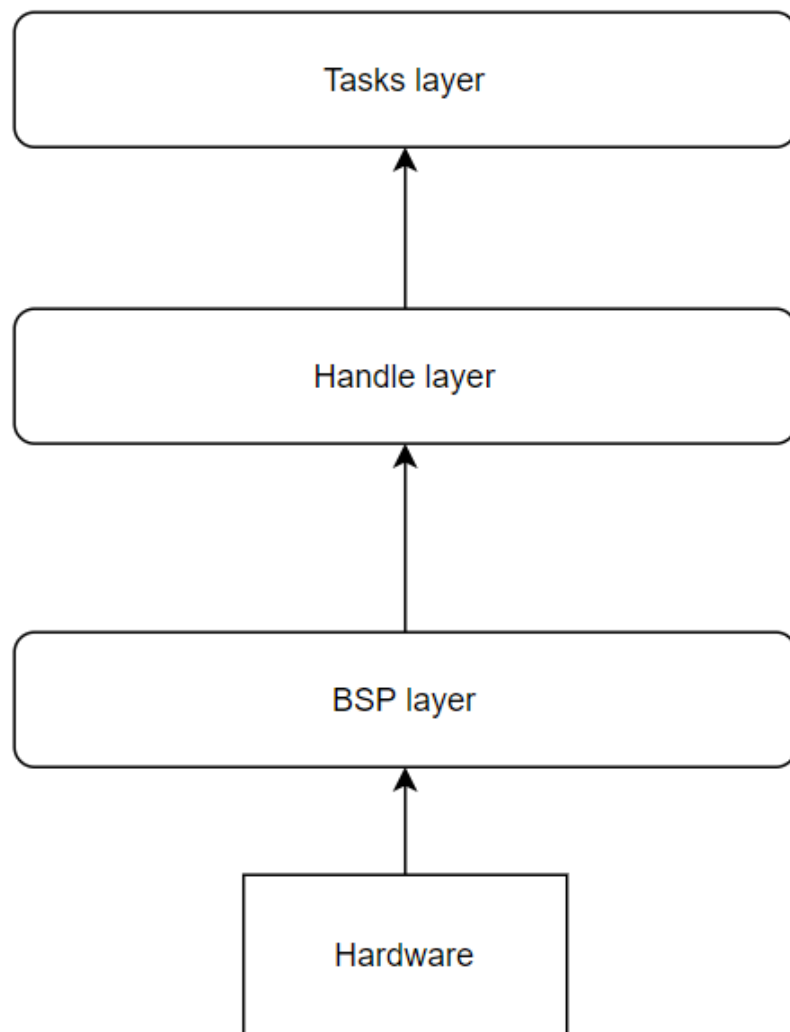
Xe AGV di chuyển bám theo đường từ, sử dụng kéo hàng hóa với tải trọng 500kg.

Một số chức năng như sau:

- Dẫn hướng theo đường từ
- Sử dụng cảm biến laser phát hiện vật cản
- Truyền thông không dây giúp đồng bộ giữa các AGV tránh sự xung đột
- Sử dụng thẻ RFID cài đặt hành động của AGV
- Lưu trữ dữ liệu trên thẻ nhớ SD
- Hỗ trợ truyền thông CAN và RS485



Hình 1. AGV - T500



## Cấu trúc chương trình

Chương trình điều khiển được chia làm 3 lớp chính gồm Tasks, Handle và BSP.

BSP là lớp chương trình làm việc trực tiếp với phần cứng, lập trình các giao tiếp ngoại vi của vi điều khiển như GPIO, UART, CAN,....

Handle là lớp trung gian trao đổi dữ liệu thô từ BSP và xử lý phù hợp với từng loại cảm biến được sử dụng

Task là lớp xử lý, sử dụng các chức năng từ Handle và đưa ra yêu cầu điều khiển. Lập trình các chức năng và hoạt động của AGV sẽ thực hiện trên lớp này.

Chỉnh sửa lớp BSP khi cần thay thế mạch main, phần cứng  
Chỉnh sửa lớp Handle khi thay thế cảm biến  
Chỉnh sửa lớp Tasks khi thay thế, lập trình chức năng AGV



Tập trung vào lớp Tasks – nơi thực hiện các chức năng của AGV

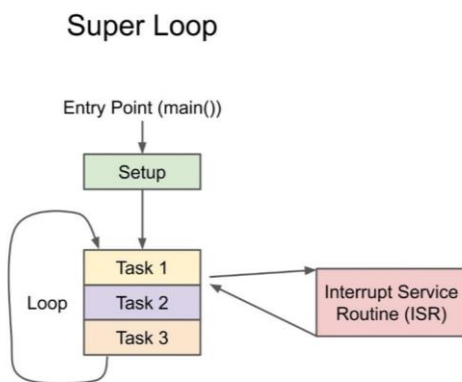


# Hệ điều hành của AGV

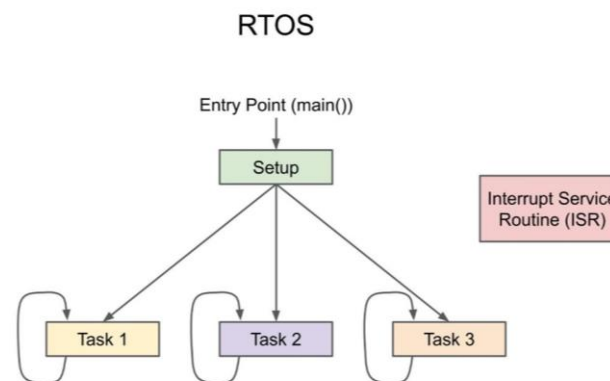
Hệ điều hành RTOS – Real Time Operator System hay được gọi là hệ điều hành thời gian thực mà cho phép ứng dụng của bạn chạy đa tác vụ và có thể đáp ứng được theo thời gian thực. Khác với Super loop thường được sử dụng trong các ứng dụng nhúng.

Tính năng của RTOS:

- Chia chương trình thành các Tasks chạy gần như đồng thời (Ưu tiên task có mức độ ưu tiên cao hơn thực hiện trước)
- Chia tasks giúp phân chia công việc tốt hơn, mỗi người/nhóm phụ trách phát triển 1 task riêng



Hình 2. Super loop

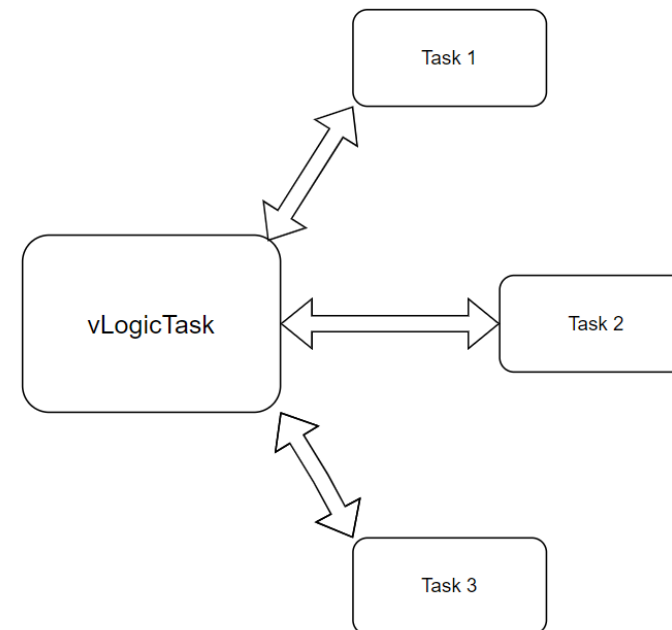


Hình 3. RTOS

## Hệ điều hành của AGV

Mỗi Task phụ trách xử lý một tính năng cụ thể của AGV.

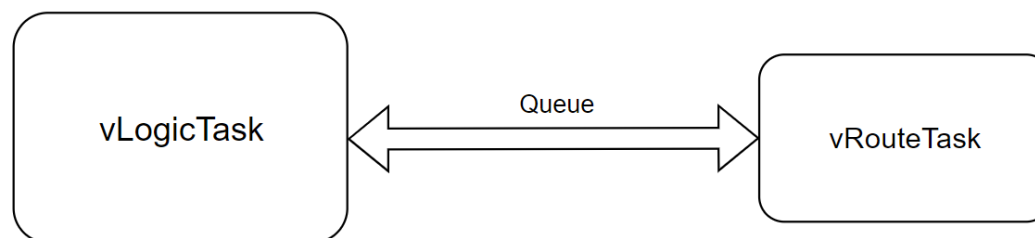
Sử dụng một Task trung tâm – vLogicTask để nhận dữ liệu từ các Tasks khác, các chức năng sẽ được lập trình tại đây sau đó phản hồi về các Tasks thành phần



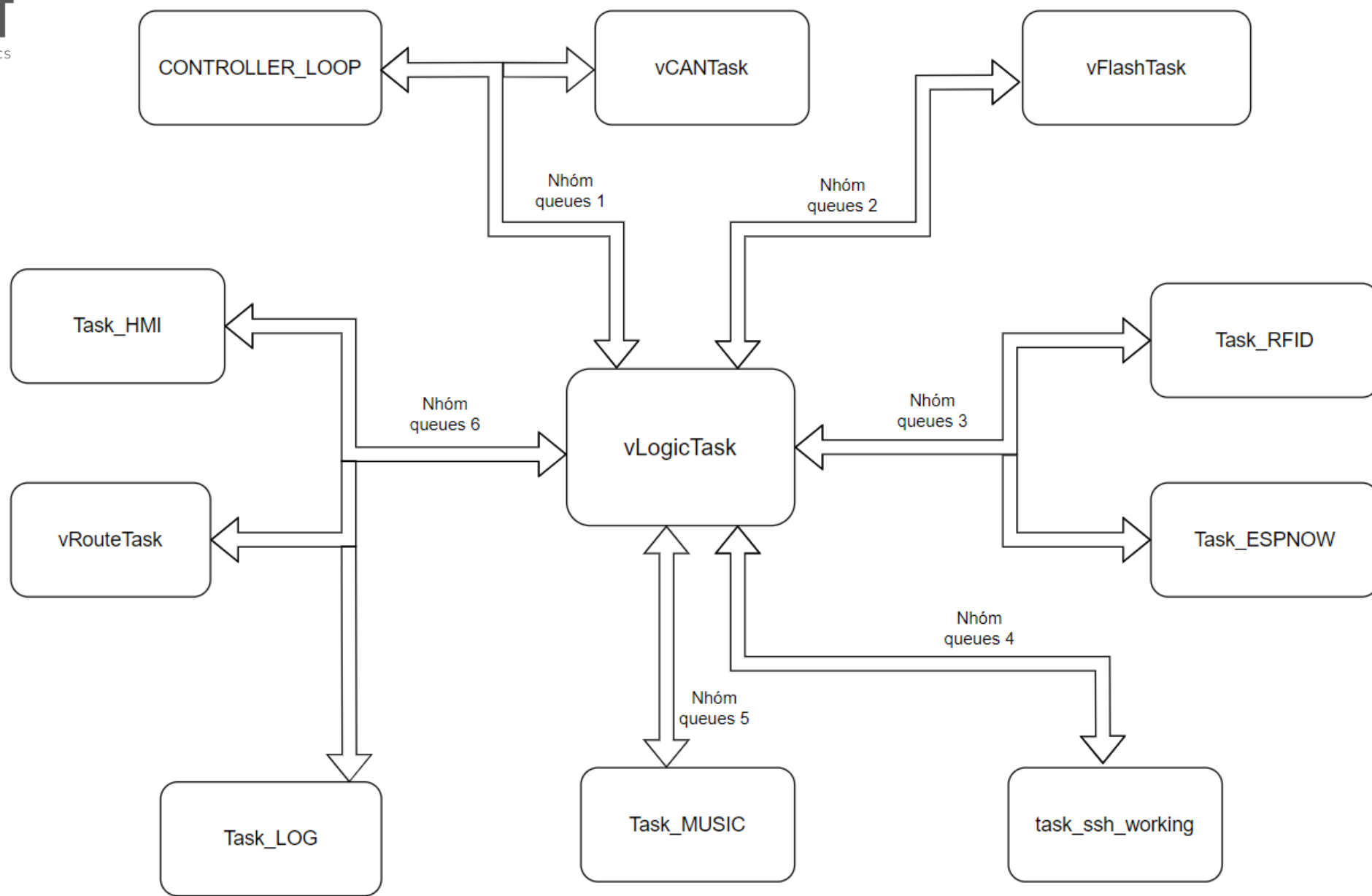
Hình 4. Cách tổ chức các Tasks

Hai task truyền dữ liệu với nhau bằng **Queue**

Mỗi Queue được quy định truyền một kiểu dữ liệu nhất định



Hình 5. Truyền dữ liệu giữa 2 Tasks



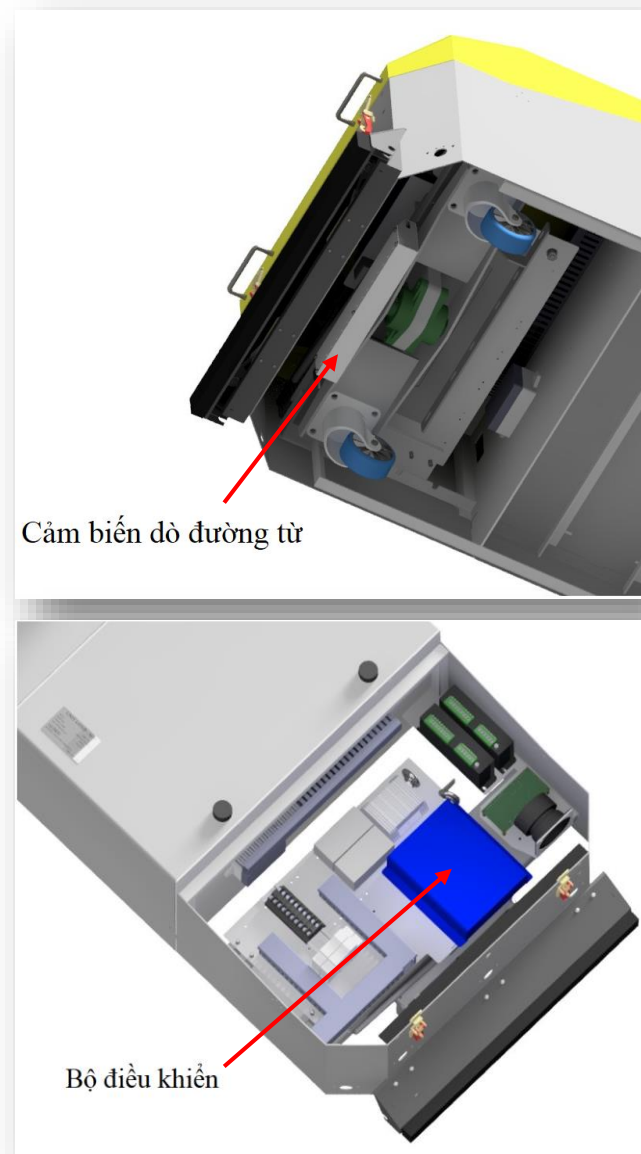
\*Note: Việc chia nhóm queue thuận tiện cho việc đọc sơ đồ, không liên quan đến cách lập trình

Hình 6. Sơ đồ trao đổi dữ liệu giữa các Tasks

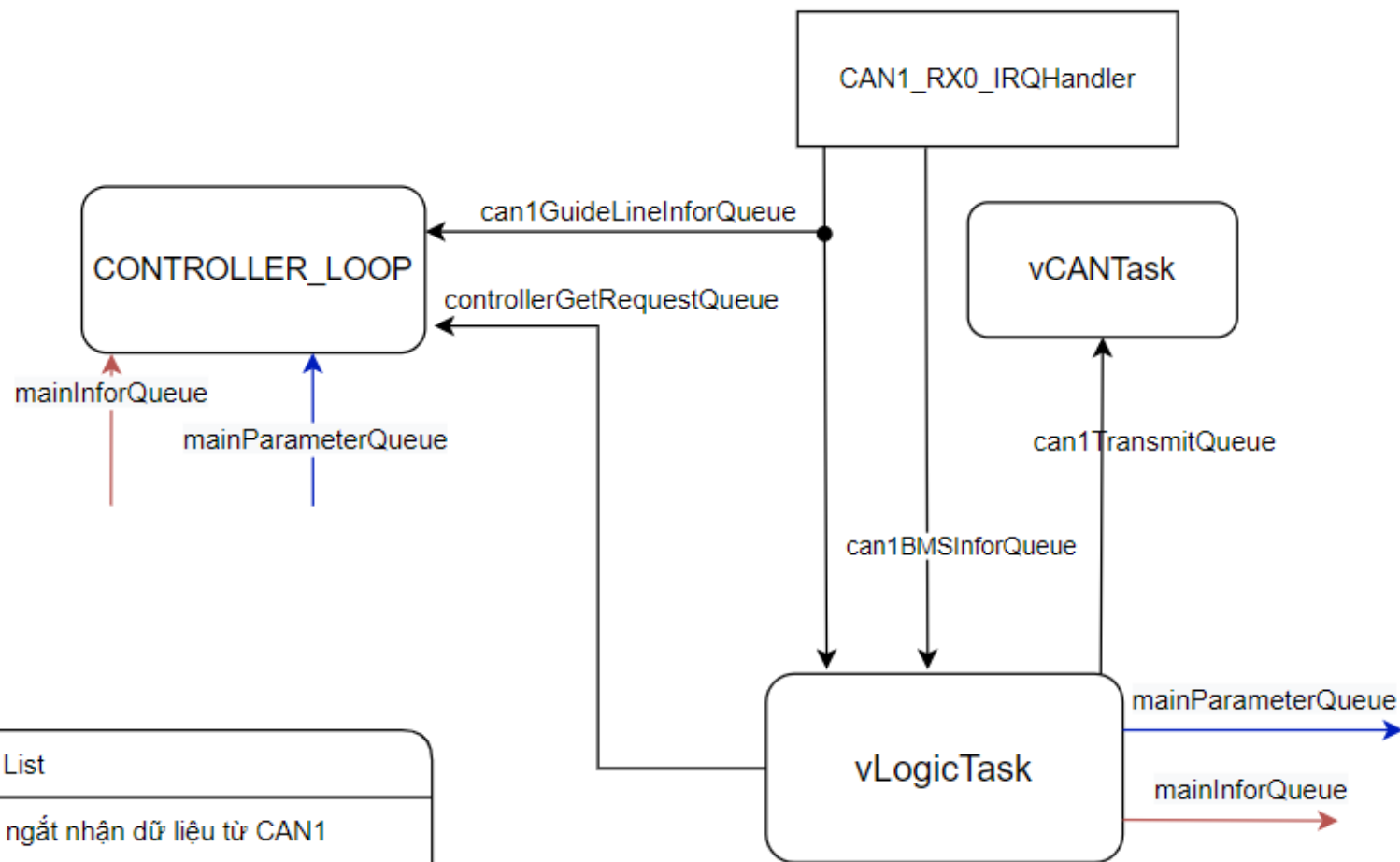
**CONTROLLER\_LOOP:** Bộ điều khiển di chuyển của AGV, đảm bảo di chuyển bám theo đường từ. Do yêu cầu chính xác về thời gian lấy mẫu từ cảm biến dò đường từ nên không được quản lý bởi FreeRTOS mà được gọi bởi hàm ngắt của hệ thống

**vCanTask:** Nhận dữ liệu từ Task\_Logic và gửi đến CAN1.

**CAN1\_RX0\_IRQHandler:** Hàm ngắt được gọi khi CAN1 nhận được dữ liệu. Chuyển dữ liệu nhận được tới Task\_Logic và CONTROLLER\_LOOP.



Hình 7. Hình minh họa

**Nhóm queues 1**

## List

CAN1\_RX0\_IRQHandler: Hàm ngắt nhận dữ liệu từ CAN1

mainInforQueue: Dữ liệu thông tin AGV gửi từ Task\_Logic

mainParameterQueue: Dữ liệu thông số cấu hình từ Task\_Logic

can1GuideLineInforQueue: Dữ liệu từ cảm biến dò line

controllerGetRequestQueue: Yêu cầu điều khiển từ Task\_Logic

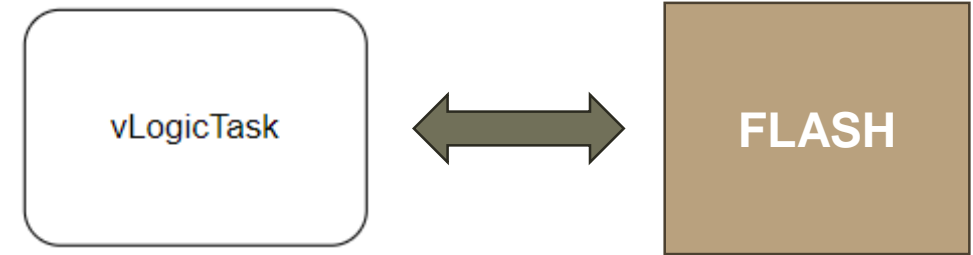
can1BMSInforQueue: Dữ liệu từ cảm biến mức pin

can1TransmitQueue: Dữ liệu Task\_Logic gửi đến CAN1

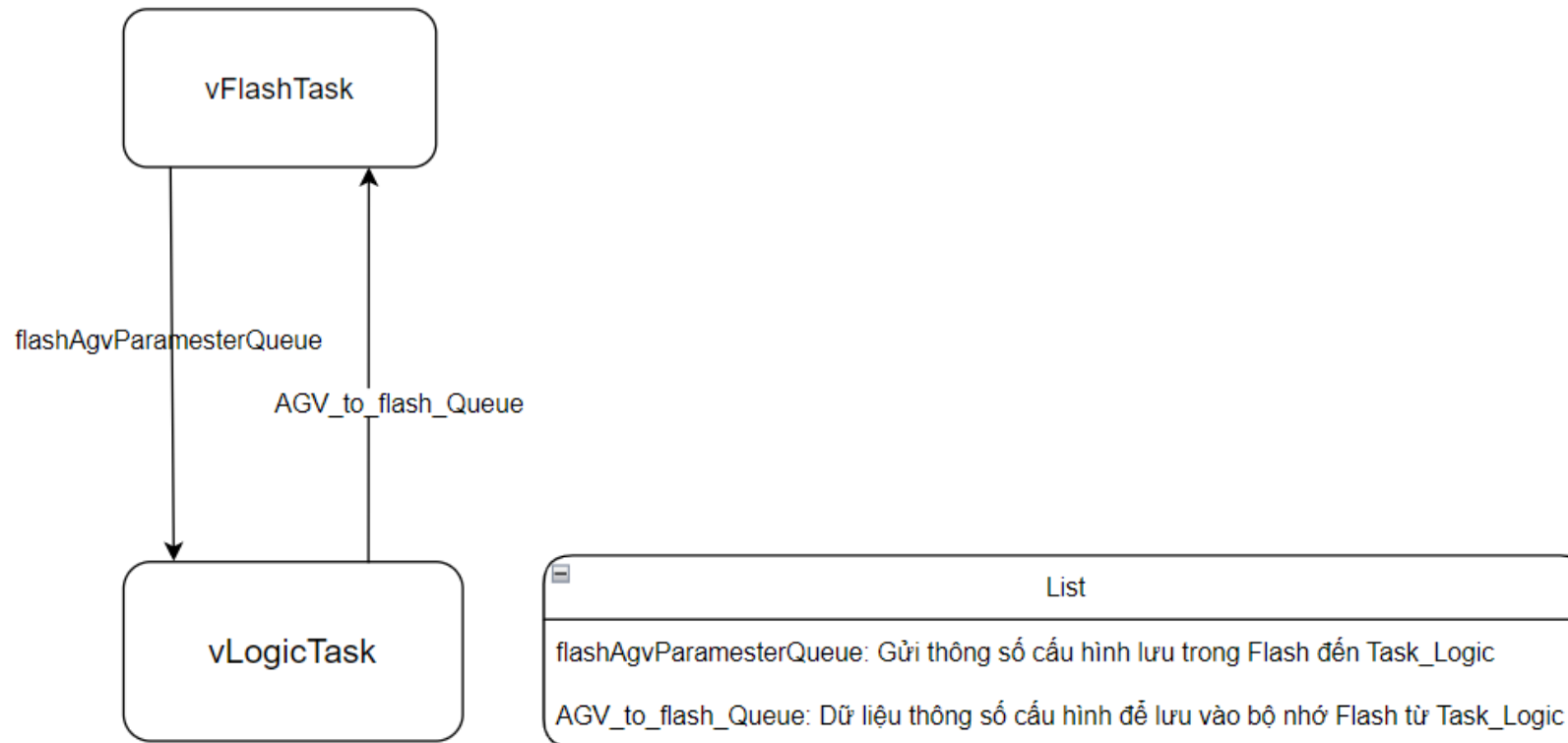


## Nhóm queue 2 Nhóm trao đổi dữ liệu giữa vFlashTask và vLogicTask.

**vFlashTask:** Thực hiện lưu thông số cài đặt đến Flash, nhưng hiện tại sẽ chủ yếu sử dụng thẻ nhớ SD ở Task\_LOG



Hình 8. Hình minh họa

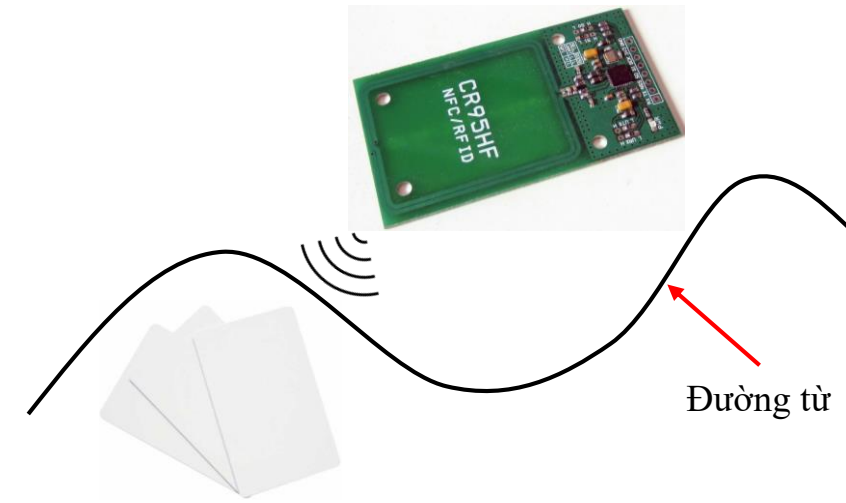


### Nhóm queue 3 Nhóm trao đổi dữ liệu giữa Task\_RFID, Task\_ESPNOW và vTaskLogic.

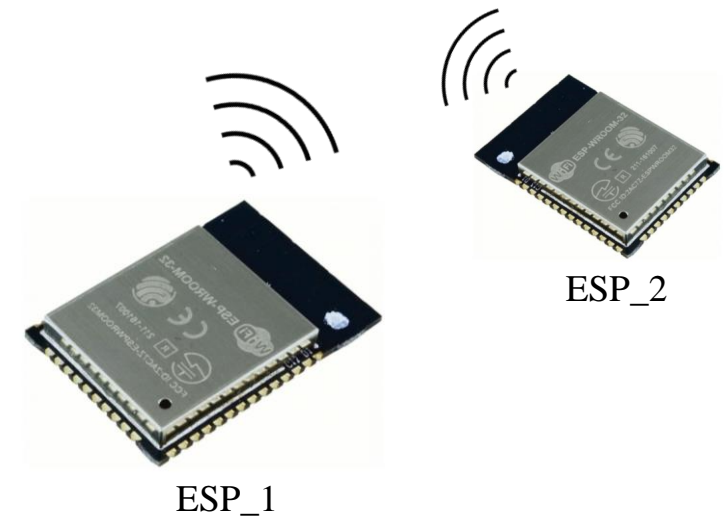
Thẻ RFID được gắn trên đường từ nhằm xác định vị trí AGV và cài đặt lệnh hoạt động

**Task\_RFID:** Đọc dữ liệu từ module RFID và gửi đến các Task xử lý như Task Logic và Task\_ESPNOW.

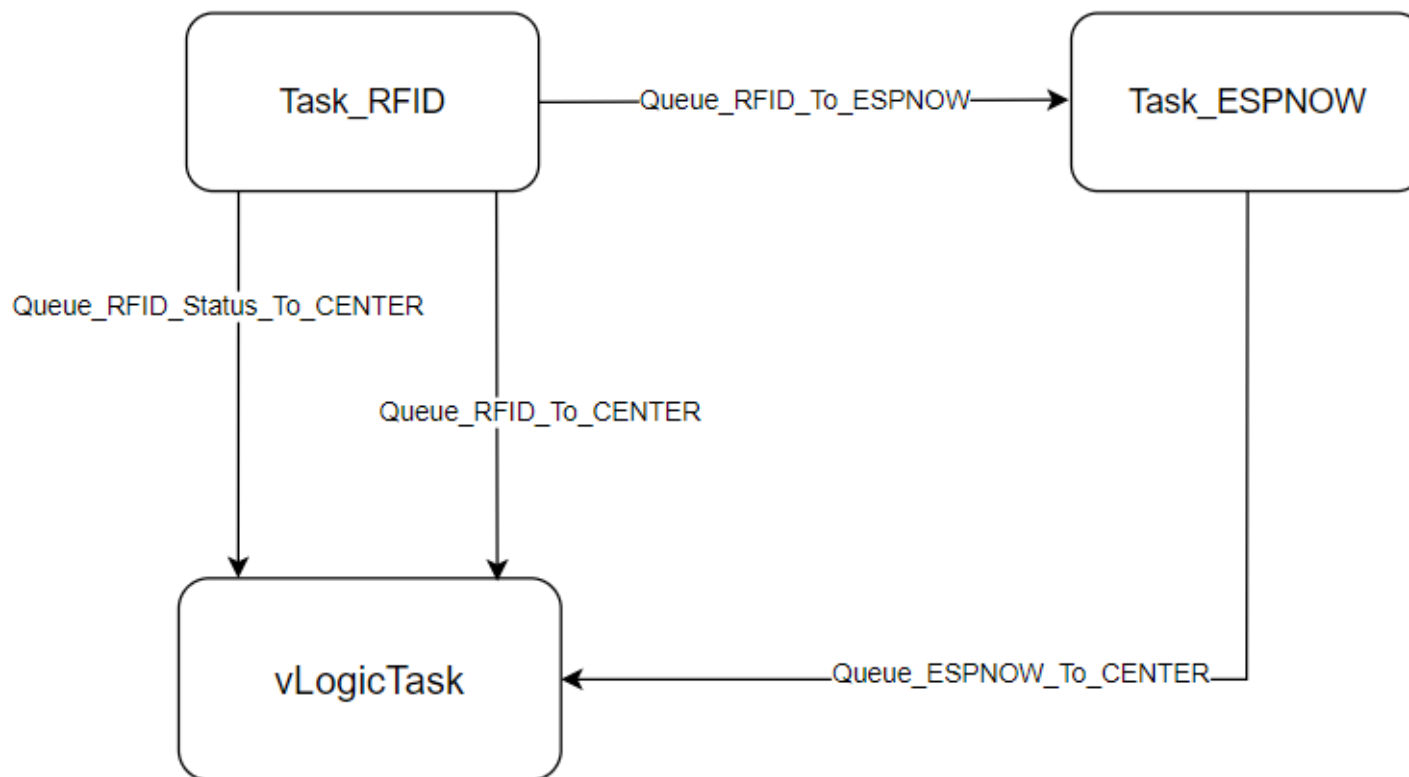
**Task\_ESPNOW:** AGV đọc thẻ RFID để xác nhận tới ngã giao, nếu có 2 xe cùng đi vào, ESPNOW sẽ giao tiếp giữa 2 AGV để quyết định thứ tự ưu tiên



Hình 9. Hình minh họa RFID



Hình 10. Hình minh họa ESPNOW

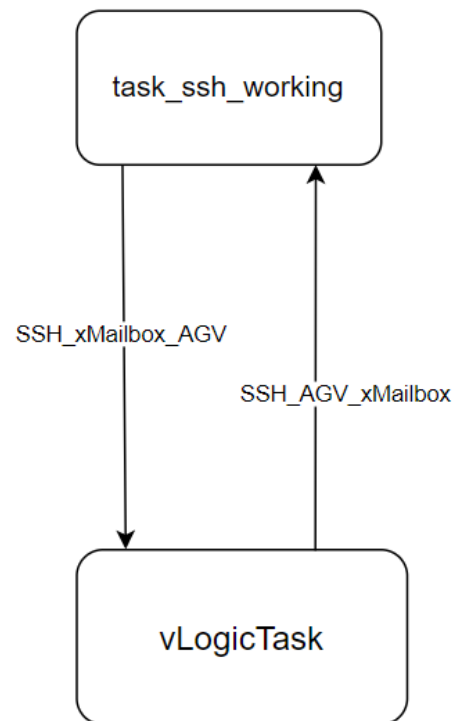
Nhóm queues 3

List
Queue_RFID_Status_To_CENTER: Gửi trạng thái module đọc thẻ đến Task_Logic
Queue_RFID_To_CENTER: Gửi dữ liệu đọc thẻ đến Task_Logic
Queue_RFID_To_ESPNOW: Gửi dữ liệu đọc thẻ đến Task_ESPNOW nhằm đồng bộ khi đến đường giao nhau
Queue_ESPNOW_To_CENTER: Task_ESPNOW giao tiếp với AGV khác và gửi dữ liệu về Task_Logic

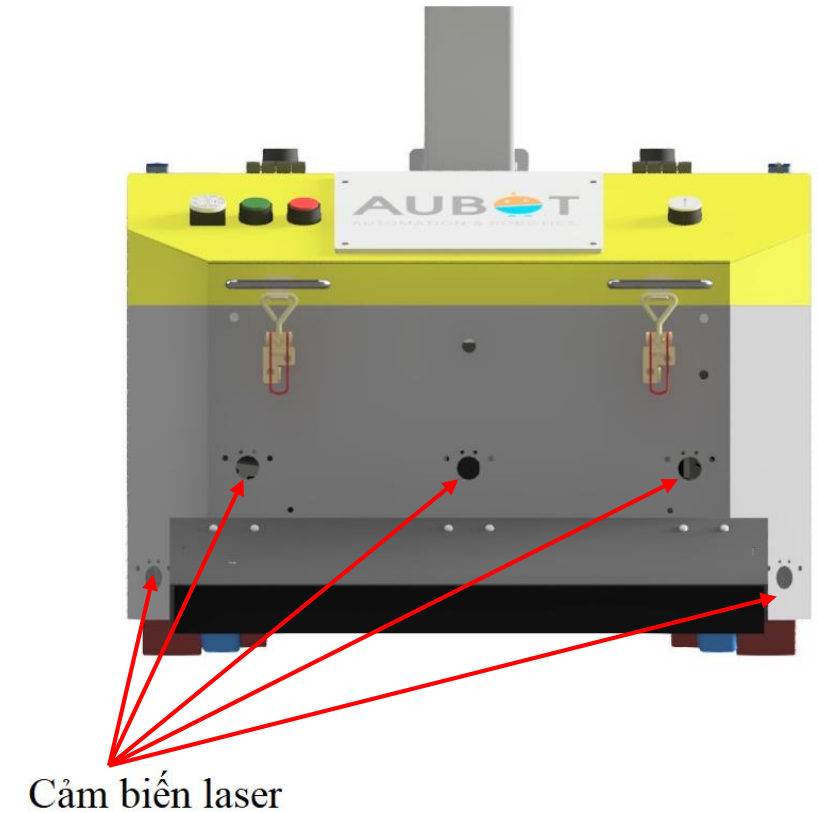
## Nhóm queue 4 Nhóm trao đổi dữ liệu giữa task\_ssh\_working và vTaskLogic.

**task\_ssh\_working:** Nhận dữ liệu SSH – dữ liệu cảm biến laser từ mạng CAN2, hàm chỉ xử lý dữ liệu để đọc ra trạng thái cảnh báo.

Lưu trạng thái và gửi trở lại để Task\_Logic thực hiện hành động xử lý.



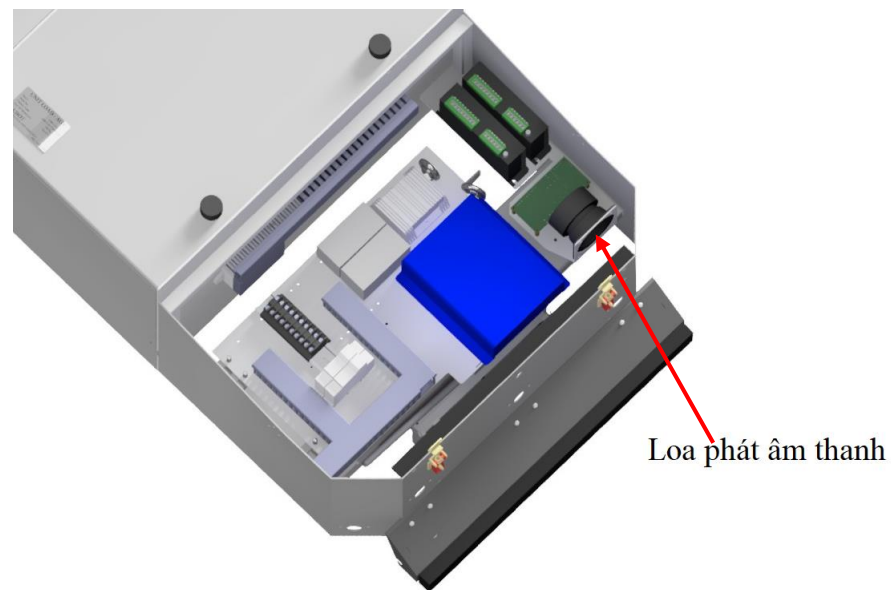
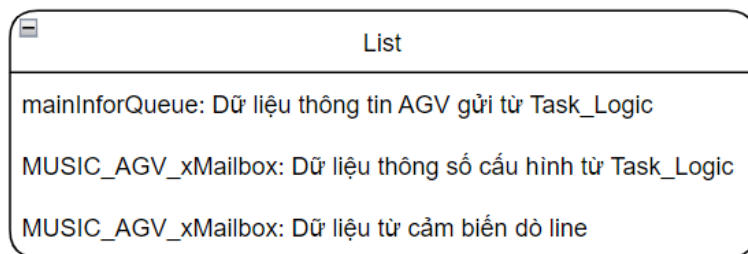
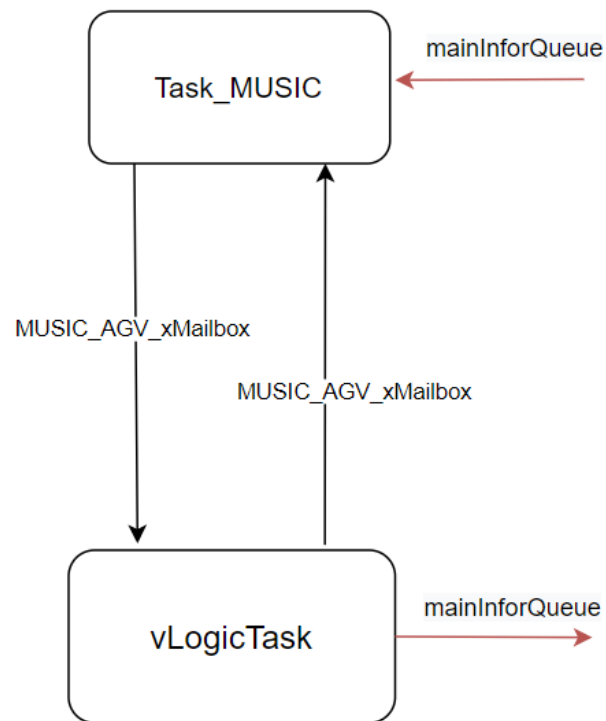
List
SSH_AGV_xMailbox: Gửi thông số cấu hình module cảm biến laser đến task_ssh_working
SSH_xMailbox_AGV: Gửi dữ liệu cảm biến laser đến Task_Logic



Hình 11. Hình minh họa Task SSH

## Nhóm queue 5 Nhóm trao đổi dữ liệu giữa Task\_MUSIC và vTaskLogic.

**Task\_MUSIC:** Phụ trách thực hiện các yêu cầu phát âm thanh từ Task\_Logic gửi đến và phản hồi lại trạng thái phát âm thanh.



Hình 12. Hình minh họa Task\_MUSIC

**Task\_HMI:** Nhận các dữ liệu thông tin vận hành và thông số cấu hình từ Task\_Logic để hiển thị lên trên màn hình HMI

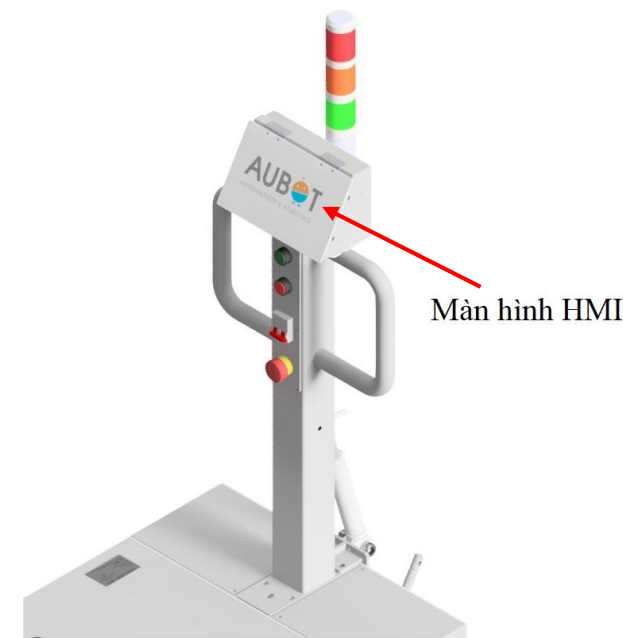
Gửi các yêu cầu được cài đặt trên màn hình xuống đến Task\_Logic

**vRouteTask:** Thực hiện tải lại bản đồ tuyến đường khi khởi động và lưu lại sau khi có yêu cầu

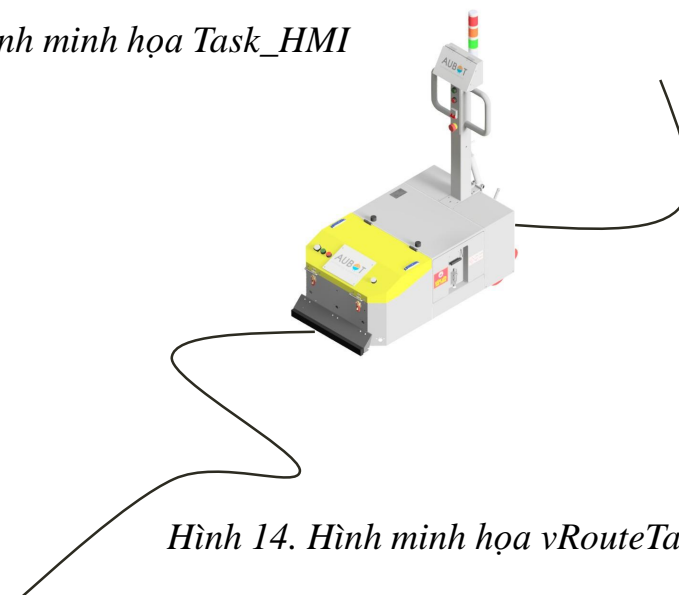
Ghi nhớ dữ liệu tuyến đường để dẫn hướng cho AGV khi đi đến các đoạn đường giao nhau

**Task\_LOG:** Thực hiện tải lại dữ liệu thông số cấu hình và bản đồ tuyến đường khi khởi động AGV

Khi dữ liệu cấu hình và tuyến đường được thay đổi sẽ thực hiện lưu những dữ liệu đó vào thẻ nhớ SD

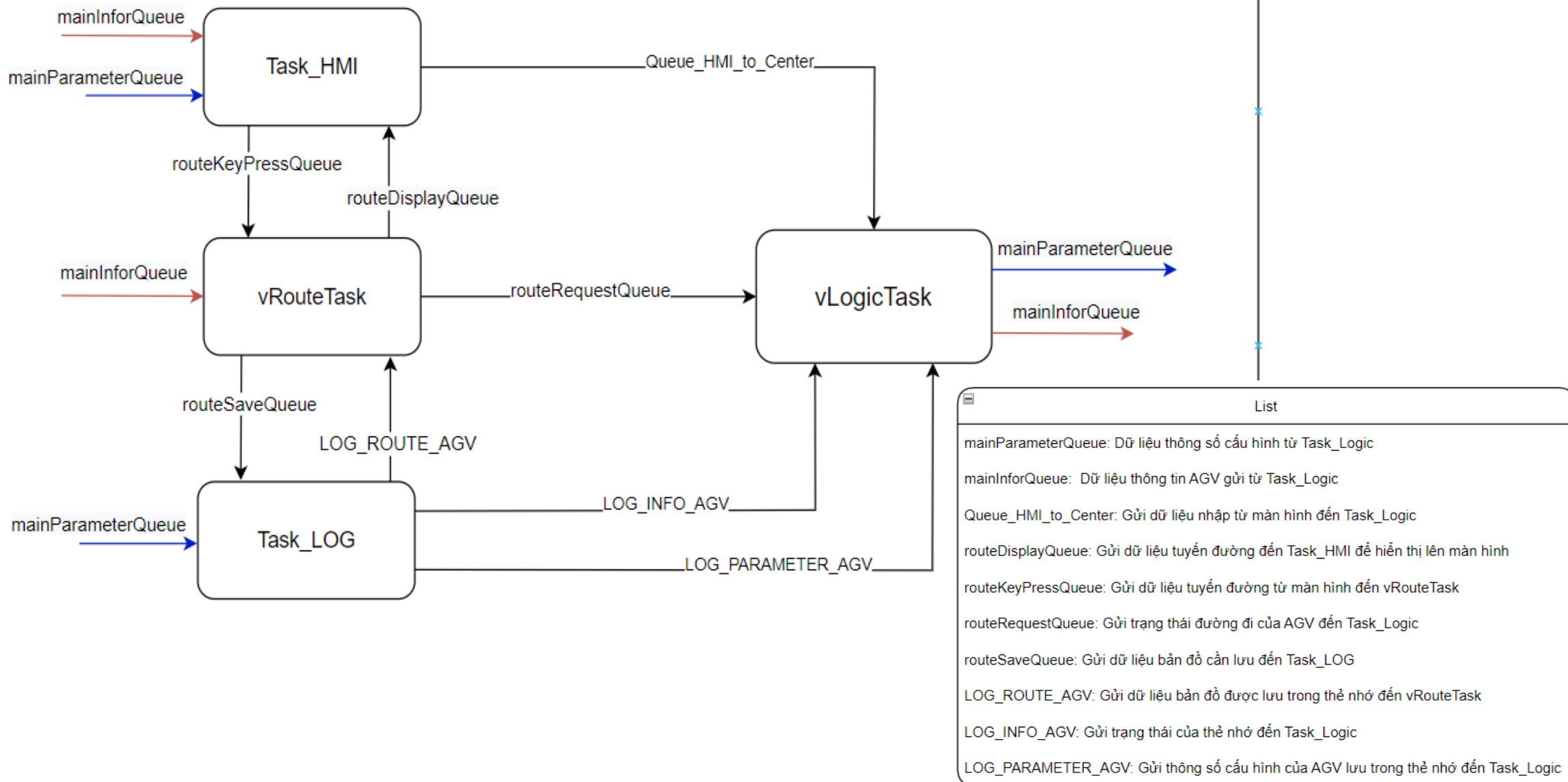


Hình 13. Hình minh họa Task\_HMI



Hình 14. Hình minh họa vRouteTask

## Nhóm queues 6



Thanks for watching