

Tổ hợp bộ điều khiển đa nhiệm cho robot tự hành sử dụng máy tính nhúng

Building a multi-tasks controller based on embedded computer for autonomous mobile robot

Tăng Quốc Nam
Học viện Kỹ thuật Quân sự
e-Mail: tangquocnam@mta.edu.vn

Phạm Thế Hùng
Đại học Kỹ thuật Lê Quý Đôn
e-Mail: phamthehung83@yahoo.com

Nguyễn Bá Đại
Đại học Kỹ thuật Lê Quý Đôn
e-Mail: badainguyen@mta.edu.vn

Tóm tắt

Robot nói chung và robot tự hành nói riêng luôn phải thực hiện đồng thời nhiều nhiệm vụ trong quá trình hoạt động. Trên các robot sử dụng vi điều khiển truyền thống việc tổ chức thực hiện đồng thời nhiều nhiệm vụ thường khó khăn do khả năng của vi xử lý và/hoặc truyền thông. Với sự phát triển của công nghệ bán dẫn, các bộ vi xử lý đa nhân ngày nay đang được áp dụng rộng rãi trong các bài toán đa nhiệm, trong đó có robot tự hành. Trong nghiên cứu này, máy tính nhúng (MTN) đa nhân Raspberry Pi 2 được sử dụng để tổ hợp bộ điều khiển cho robot tự hành ER-400 AGV thực hiện đồng thời các nhiệm vụ như định vị, di chuyển theo quỹ đạo và tránh vật cản trên đường đi, đo vẽ bản đồ môi trường cục bộ, truyền dữ liệu về trung tâm. Các kết quả thực nghiệm cho thấy tần số hoạt động tăng lên đáng kể so với bộ điều khiển truyền thống.

Từ khóa: Robot tự hành, Thư viện Posix thread, Tổ chức vi xử lý đa nhân theo thời gian thực.

Abstract: Robot in general and particular mobile robot always has to perform multi-tasks at the same time during operation. On the mobile robot used microcontrollers, organizing perform multiple tasks simultaneously is often difficult due to the ability of microcontrollers and/or communications. Nowadays, with the development of semiconductor technology, multi-core processors are increasingly widely applied to handle multitasking problem, including mobile robots. In this study, multi-core embedded computer Raspberry Pi 2 is used to create the controller for mobile robot ER-400 AGV, so that robot can perform simultaneous tasks such as positioning, moving in orbit and avoid obstacles on paths, mapping local environment, transfer data to center. The experimental results show that the operating frequency increased significantly compared to traditional controllers.

Keywords: Mobile robot, Posix thread, Multi-core real-time scheduling

Chữ viết tắt

MTN Máy tính nhúng

POSIX	Portable Operating System Interface
GPIO	General-purpose input/output
I/O	Input/Output
USB	Universal Serial Bus

1. Mở đầu

Các nhiệm vụ cơ bản của một robot tự hành bao gồm: định vị, điều khiển chuyển động theo quỹ đạo, phát hiện và tránh vật cản trên đường đi, đo vẽ bản đồ môi trường cục bộ xung quanh robot trong quá trình vận động,... và các nhiệm vụ khác theo các yêu cầu công việc cụ thể. Trên các robot sử dụng vi điều khiển truyền thống việc tổ chức thực hiện đồng thời nhiều nhiệm vụ thường khó khăn do khả năng của vi xử lý và/hoặc truyền thông. Khi đó các nhiệm vụ của robot được thực hiện tuần tự, tuy nhiên với thời gian thực hiện các nhiệm vụ là nhỏ nên có thể coi như robot thực hiện các nhiệm vụ này một cách đồng thời.

Ngày nay, với sự phát triển của công nghệ bán dẫn, các bộ vi xử lý đa nhân đã trở nên phổ biến trong các ứng dụng kỹ thuật để giải quyết các bài toán đa nhiệm. Trong kỹ thuật robot di động, MTN đa nhân là một lựa chọn phù hợp để xây dựng bộ điều khiển với yêu cầu đa nhiệm, kích thước nhỏ gọn, nguồn năng lượng không cao, làm việc tốt trong các môi trường phức tạp. Đã có nhiều nghiên cứu phát triển các bộ điều khiển cho robot tự hành dựa trên các loại MTN khác nhau. Tuy nhiên, với mỗi cấu hình phần cứng của robot, các tác vụ mà robot cần thực hiện, cấu hình của MTN,... mà cần có các nghiên cứu cụ thể.

Trong nghiên cứu này, MTN đa nhân Raspberry Pi 2 được sử dụng để phát triển bộ điều khiển cho robot di động ER-400 AGV của Intelitek Inc., đã được phát triển thêm các tác vụ như định vị, di chuyển theo quỹ đạo và tránh vật cản, xây dựng bản đồ môi trường cục bộ trong quá trình vận động. Nội dung chính của báo cáo gồm 2 phần: phần 2 mô tả về robot di động ER-400 AGV của Intelitek Inc., các tác vụ được phát triển thêm và cấu trúc bộ điều khiển trên cơ sở máy tính nhúng Raspberry Pi 2 cho robot này; phần 3 giới thiệu về thư viện POSIX thread, các thuật toán phân bổ nhiệm vụ trên các nhân của MTN, phương án tổ chức phân bổ các nhiệm vụ của robot ER-400 AGV cho các nhân của MTN và các kết quả thực nghiệm; cuối cùng là các kết luận và kiến nghị.

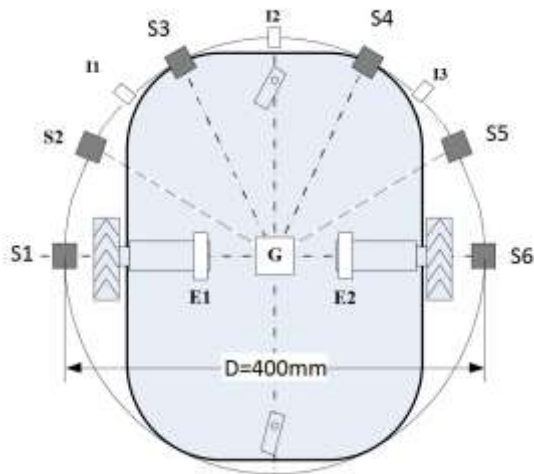
2. Tổ hợp phần cứng bộ điều khiển trên cơ sở MTN cho robot ER-400 AGV

2.1. Robot ER-400 AGV

ER-400 AGV (H. 1a) là một robot di động của hãng Intelitek Inc., được phát triển thêm các tác vụ như tránh vật cản [4][6], đo vẽ bản đồ môi trường [5], và định vị [7]. Đây là robot tự hành di chuyển bằng bánh lốp, với hai bánh chủ động dẫn động độc lập. Các encoder (E1, E2) được gắn trên trục động cơ dẫn động của từng bánh xe để đo quãng đường dịch chuyển tương ứng. Trên robot được lắp 6 cảm biến siêu âm (S1 – S6) loại SRF04 của Acroname Easier Robotics, có khoảng đo từ 30mm - 3000mm. Ngoài ra trên robot còn được bố trí thêm các cảm biến hồng ngoại (I1 – I3), cảm biến vận tốc và gia tốc (G). Sơ đồ bố trí cảm biến trên robot như trên hình H. 1b.



(a)



(b)

H. 1 Robot ER-400 AGV

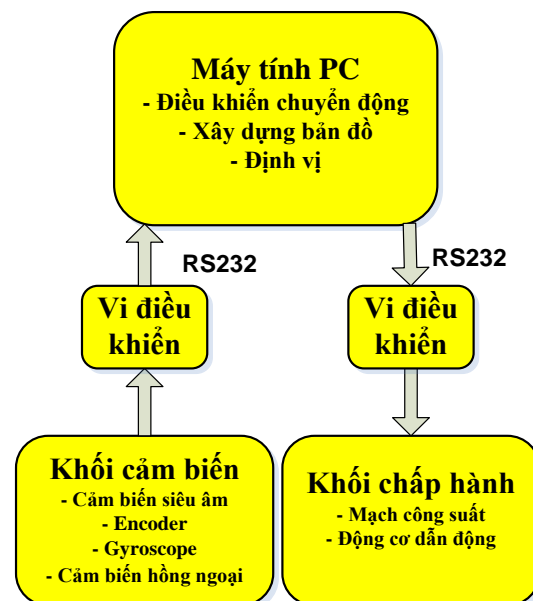
Các tác vụ đã được phát triển thêm trên robot tự hành ER-400 AGV gồm:

1. Điều khiển chuyển động [4][6]: Tùy theo nhiệm vụ cụ thể, robot tự hành thường phải di chuyển theo quỹ

đạo cứng (bám theo các quỹ đạo đã thiết lập trước, bám tường, bám vạch,...) hoặc di chuyển theo quỹ đạo thay đổi (bám mục tiêu di động, di chuyển tự do khám phá môi trường). Trong quá trình di chuyển, robot sẽ gặp phải các vật cản trên quỹ đạo. Vật cản xuất hiện trên đường đi của robot có thể là vật tĩnh hoặc động, có thể xuất hiện ngẫu nhiên, không được biết trước cả về vị trí, kích thước và hình dáng. Do đó trong quá trình vận động robot đồng thời phải phát hiện và tránh vật cản trên đường đi.

2. Định vị [7]: Dựa trên các dữ liệu nhận được từ 2 encoder và dữ liệu về vận tốc, gia tốc của robot từ cảm biến GY-86, bộ điều khiển của ER-400 AGV phải xác định được vị trí hiện tại của nó so với điểm xuất phát ban đầu hoặc vị trí của nó trên bản đồ số đã được biết trước.

3. Vẽ bản đồ [5]: Dữ liệu về vị trí hiện tại của robot được truyền lên máy tính PC thông qua Wifi để vẽ bản đồ môi trường hoạt động của nó.



H. 2 Bộ điều khiển nguyên thủy của robot ER-400 AGV

Cấu trúc hệ thống điều khiển nguyên thủy của robot ER-400 AGV như chỉ ra trên hình H. 2. Chương trình điều khiển chính và các tác vụ của robot được thực hiện trên máy tính. Khối cảm biến thu nhận tín hiệu từ các cảm biến siêu âm (để đo các khoảng cách đến vật cản tương ứng), từ encoder (để xác định vị trí hiện tại của robot) và truyền lên máy tính. Khối chấp hành nhận tín hiệu điều khiển chuyển động từ máy tính và điều khiển các động cơ dẫn động bánh xe. Truyền thông giữa các khối được thực hiện thông qua cổng RS232.

Khối cảm biến dùng để thu nhận dữ liệu về địa hình, quỹ đạo là mạch dùng vi điều khiển At89C51. Vi điều khiển này sẽ đọc lần lượt dữ liệu từ các cảm biến siêu âm, encoder, sau đó gửi lên máy tính PC thông qua cổng RS232 để tính toán. Sau khi bộ điều khiển tính toán xong, máy tính lại gửi dữ liệu xuống một vi điều khiển At89C51 khác trong khối điều khiển chuyển động để điều khiển robot di chuyển. Do đó, thời gian

của một chu kỳ đọc, gửi dữ liệu, xử lý, gửi dữ liệu, và điều khiển chuyển động của robot sẽ bằng tổng thời gian của các công việc trên và khoảng thời gian này không hề nhỏ (khoảng 0.12s - 0.2s). Qua thực nghiệm trong các nghiên cứu trước [4][5][6][7], tần số làm việc của robot vào khoảng 5 - 8Hz, có nghĩa là trong 1s robot có thể hoàn thành 5-8 lần toàn bộ các nhiệm vụ của nó như đã nói ở trên.

Như vậy để tăng tốc độ của một chu kỳ xử lý của robot, việc đọc các cảm biến cần được thực hiện song song, giảm tối đa việc dữ liệu phải truyền qua lại giữa máy tính và các vi điều khiển. Điều này có thể được thực hiện trên các MTN đa nhân, với việc đọc dữ liệu của các cảm biến được thực hiện song song nhau trên các nhân của bộ vi xử lý, không mất thời gian truyền dữ liệu giữa máy tính và vi điều khiển nếu như một nhân của MTN được dùng riêng để lấy, xử lý dữ liệu đọc được của các nhân khác.

2.2. Cấu trúc bộ điều khiển trên cơ sở máy tính nhúng Raspberry Pi 2

Để giải quyết vấn đề như đã phân tích ở phần trên, nghiên cứu này đưa ra một giải pháp mới giúp tăng tốc độ xử lý của robot bằng việc đồng thời đọc tín hiệu từ các cảm biến và thực hiện các nhiệm vụ khác nhờ MTN đa nhân. Các nhiệm vụ đọc cảm biến sẽ được thực hiện song song nhau trên 4 nhân của MTN. MTN được sử dụng ở đây là Raspberry Pi 2 - Model B với các thông số kỹ thuật chính như sau:

- Vi xử lý Broadcom BCM2836 Cortex A7 4 lõi
- Tốc độ xử lý 900MHz
- Bộ nhớ 1GB RAM
- 40 chân mở rộng GPIO
- 4 cổng USB
- Hệ điều hành: Linux, nhân thời gian thực 3.18.7-rt1-v7+ #1 SMP PREEMPT RT

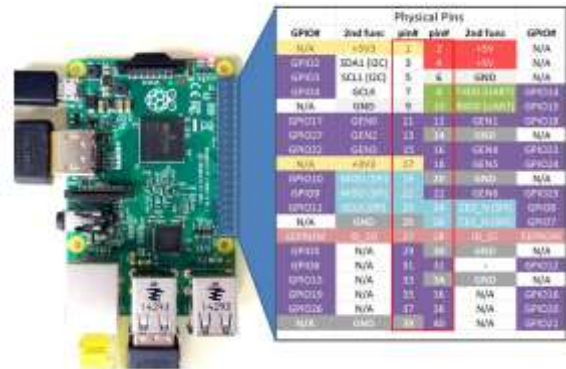
Các cảm biến được dùng trên robot gồm có: 6 cảm biến siêu âm HC-SR04, 2 Encoder tương đối được gắn với trục quay của động cơ dẫn động, 1 cảm biến gia tốc GY-86 được đặt ở trên khung của robot, 3 cảm biến hồng ngoại được bố trí xung quanh thân robot.

Bảng 1: Phân bố chân cắm của các cảm biến với MTN Raspberry Pi 2

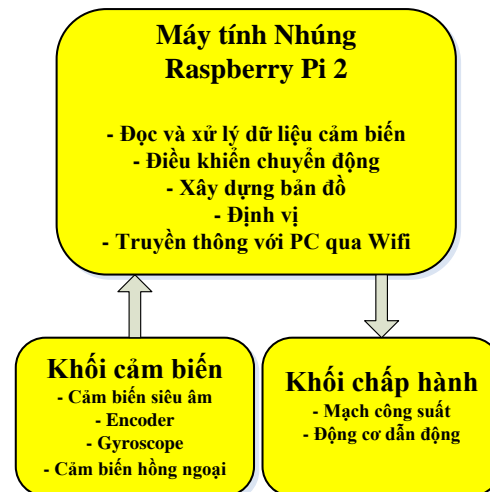
STT	Cảm biến	Số lượng	Tổng số chân	Chân trên Raspberry pi 2
1	SR04	6	12	I/O
2	Encoder	2	4	I/O
3	Gyroscope	1	2	SCL,SDA
4	Wifi	1	0	USB
5	Hồng ngoại	3	3	I/O
6	PWM	2	2	I/O

Như mô tả trong bảng 1, số chân cần thiết để kết nối đến Raspberry Pi từ các cảm biến là 21 chân. Ngoài ra, cần thiết phải sử dụng thêm 2 chân GPIO nữa của MTN để tạo ra xung điều khiển 2 động cơ dẫn động. Như vậy tổng số chân GPIO cần thiết là 23 chân.

MTN Raspberry Pi 2 có 40 chân GPIO (H. 3), trong đó có 8 chân GND, 2 chân nguồn 3.3V, 2 chân nguồn 5V, và 2 chân dùng cho EEPROM. Như vậy còn lại 26 chân GPIO của Raspberry Pi 2 có thể dùng cho việc đọc cảm biến - đủ dùng cho yêu cầu của bài toán với số lượng các cảm biến như trên.



H. 3 Sơ đồ bố trí các chân GPIO của MTN



H. 4 Cấu trúc hệ thống điều khiển trên cơ sở MTN

Cấu trúc hệ thống điều khiển trên cơ sở MTN của robot được chỉ ra trên H. 4. Bộ điều khiển và chương trình điều khiển chính được thực hiện trên MTN Raspberry Pi 2. Khối cảm biến gồm các cảm biến siêu âm (để đo các khoảng cách đến vật cản tương ứng), encoder (để xác định quãng đường di chuyển của mỗi bên tương ứng), cảm biến hồng ngoại để xác định các chướng ngại vật ở gần mà cảm biến siêu âm không phát hiện được, cảm biến vận tốc và gia tốc GY86 để hỗ trợ cho việc định vị trí hiện tại của robot. Khối chấp hành là hai mạch công suất, nhận tín hiệu điều khiển từ máy tính và điều khiển các động cơ dẫn động bánh xe. Truyền thông giữa các khối được thực hiện trực tiếp: mỗi cảm biến siêu âm, encoder được kết nối với Raspberry Pi thông qua 2 chân I/O bình thường, cảm biến tốc độ và gia tốc được kết nối với Raspberry Pi qua 2 chân I/O của cổng I2C, cảm biến hồng ngoại được kết nối qua 1 chân I/O bình thường. Các dữ liệu thu được từ cảm biến ngoài việc dùng làm đầu vào cho bộ điều khiển và được ghi lại vào file để sử dụng

về sau. Dữ liệu của các cảm biến được bộ điều khiển xử lý, kết quả của quá trình này được truyền lên máy tính PC thông qua Wifi để vẽ bản đồ.

Một điểm khác biệt cơ bản, và cũng là ưu điểm của việc sử dụng MTN như phương án trong hình 4 so với bộ điều khiển như phương án trong hình 2 đó là việc đọc dữ liệu các cảm biến được thực hiện trực tiếp giữa MTN và cảm biến, không mất thời gian truyền dữ liệu từ vi điều khiển lên PC và từ PC xuống vi điều khiển thông qua cổng RS232. Trong phương án này, dữ liệu cảm biến đọc được sẽ được MTN xử lý trực tiếp không qua truyền thông. Điều này giúp giảm đáng kể thời gian xử lý của robot.

3. Tổ chức phân bố nhiệm vụ trên các nhân của máy tính nhúng

3.1. Thư viện POSIX Thread và các thuật toán phân bố nhiệm vụ theo thời gian thực

Để thực hiện được việc phân chia nhiệm vụ cho các nhân, thư viện POSIX thread [2] được sử dụng trong quá trình lập trình cho MTN Raspberry Pi 2.

Thư viện POSIX luồng (thread) là một thư viện chuẩn, dựa trên giao thức luồng API cho C/C++. Nó cho phép tạo ra một luồng xử lý song song mới. Điều này rất hiệu quả, nhất là trên các hệ thống xử lý đa bộ xử lý hoặc hệ thống xử lý đa nhân mà các tiến trình (process) có thể được tổ chức để chạy trên nhiều bộ xử lý do đó tăng tốc độ xử lý thông qua quá trình xử lý song song (parallel processing) hay phân tán (distributed processing) [1].

Việc sử dụng luồng đòi hỏi ít phức tạp hơn so với sự phân chia (forking) hoặc tạo ra một tiến trình mới, vì hệ thống không khởi tạo một hệ thống không gian bộ nhớ ảo mới và môi trường cho tiến trình này.

Việc sử dụng luồng trong thư viện POSIX thread giúp cho các hệ thống đa nhân cũng như đơn nhân khai thác độ trễ (latency) của I/O và của hệ thống (Một luồng vẫn có thể được thực hiện trong khi các luồng khác đang chờ I/O hoặc độ trễ của hệ thống).

Các thuật toán phân bố nhiệm vụ theo thời gian thực đa nhân có thể được chia thành hai nhóm, cụ thể là thuật toán phân bố nhiệm vụ toàn cục (global scheduling algorithm) và thuật toán phân bố nhiệm vụ theo phân vùng (partitioned scheduling algorithm)[3].

Trong thuật toán phân bố toàn cục, tất cả các công việc được lưu trữ trong một hàng đợi ưu tiên duy nhất và lập kế hoạch phân bố các công việc đó thực hiện trên từng nhân theo giá trị ưu tiên của nó. Kết quả là, một công việc có thể được sắp xếp trên các lõi khác nhau vào những thời điểm khác nhau, do đó yêu cầu di chuyển các công việc trong lõi. Thuật toán phân bố toàn cục cho phép tối ưu việc phân bố nhiệm vụ, hiệu năng sử dụng thuật toán phân bố công việc cao và cân bằng tải cho các nhân rất tốt, tuy nhiên việc dự đoán nhiệm vụ nào được thực hiện trên nhân nào là một vấn đề rất khó khăn.






Đối với thuật toán phân bố nhiệm vụ theo phân vùng, các nhiệm vụ được phân cố định vào nhân và ở lại đây trong suốt quá trình tồn tại của nó. Việc phân bố các

nhiệm vụ cho mỗi nhân riêng biệt tương ứng với bài toán phân bố nhiệm vụ cho một nhân. Ưu điểm của phương pháp này là cải thiện khả năng dự đoán do loại bỏ được việc di chuyển các nhiệm vụ đến các nhân khác nhau trong quá trình thực hiện. Tuy nhiên, để tối ưu phân chia nhiệm vụ cho các nhân là một vấn đề khó khăn, hiệu quả thực hiện công việc của phương pháp phân vùng thường thấp hơn nhiều so với các thuật toán lập lịch toàn cục và sự mất cân bằng tải trên các nhân là điều không thể tránh khỏi do các nhiệm vụ có đặc tính khác nhau.

3.2. Tổ chức phân bố nhiệm vụ trên các nhân của máy tính nhúng

Trong quá trình thực hiện nhiệm vụ của mình, robot cần thực hiện một số tác vụ như: điều khiển chuyển động, đọc và xử lý tín hiệu cảm biến, định vị, xây dựng bản đồ, và truyền thông với PC (bảng 2). Trong các tác vụ trên, việc xử lý các tác vụ như xử lý dữ liệu cảm biến, điều khiển chuyển động, định vị xảy ra trong khoảng thời gian ngắn. Do đó có thể thực hiện chung các tác vụ này cùng trên một nhân của MTN.

Bảng 2. Các nhiệm vụ robot cần thực hiện

	Đọc dữ liệu từ các cảm biến siêu âm
	Đọc dữ liệu từ các cảm biến hồng ngoại
	Đọc dữ liệu từ các encoder
	Thu thập, xử lý dữ liệu, truyền dữ liệu lên PC, điều khiển chuyển động, định vị, xây dựng bản đồ
	Đọc dữ liệu từ cảm biến gia tốc

Tác vụ cần nhiều thời gian là đọc dữ liệu cảm biến, đặc biệt là đọc dữ liệu cảm biến siêu âm. Theo nguyên lý, thời gian để đọc tín hiệu cảm biến siêu âm bằng thời gian sóng âm di chuyển từ robot đến chướng ngại vật cộng với khoảng thời gian sóng siêu âm phản xạ từ chướng ngại vật trở lại robot. Do đó khoảng thời gian dành để thực hiện việc đọc cảm biến siêu âm sẽ được tính bằng thời gian đo khoảng cách xa nhất mà nó có thể đo được (3000mm đối với cảm biến siêu âm HR-04), vì nếu khoảng thời gian này được gán cố định bằng một số nhỏ hơn khoảng thời gian để cảm biến siêu âm đo khoảng cách xa nhất nó có thể, rất có thể tín hiệu cảm biến sẽ bị mất trong một số trường hợp khi sóng âm phản xạ chưa kịp về tới robot thì nhiệm vụ đọc đã kết thúc. Theo như tính toán, thời gian này vào khoảng 0.018s. Do đó, nếu việc đọc các cảm biến siêu âm được thực hiện tuần tự, sẽ dẫn đến việc thời gian một chu kỳ để robot có thể đọc tất cả các cảm biến và xử lý công việc của nó kéo dài. Do đó tần số hoạt động của robot không cao. Vì vậy các cảm biến siêu âm cần được chia đều thực hiện trên ba nhân còn lại của MTN. Thêm nữa, nếu đọc các cảm biến siêu âm cạnh nhau cùng lúc có thể dẫn đến tình

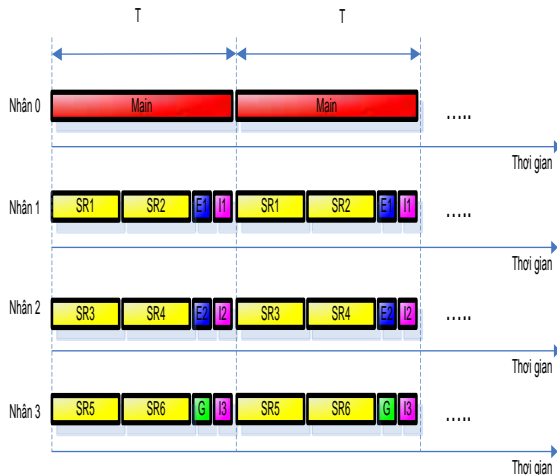
trạng hai cảm biến siêu âm cùng quét chung một vùng nhất định. Để tránh điều này, các cảm biến cạnh nhau không được đọc đồng thời. Có 6 cảm biến siêu âm được trang bị trên robot, trong đó chỉ có 3 nhân được dùng để đọc cảm biến, do đó trên mỗi nhân, 2 cảm biến siêu âm sẽ được đọc lần lượt.

Trên robot có 3 cảm biến hồng ngoại, do đó ba cảm biến này sẽ được thực hiện đồng thời trên 3 nhân khác nhau.

Bộ điều khiển của robot cần phải đọc thêm 2 encoder và 1 gyroscope, do đó 3 nhiệm vụ này cũng sẽ được thực hiện song song nhau trên 3 nhân độc lập.

Từ các phân tích ở trên, phương án tổ chức các nhiệm vụ trên 4 lõi của Raspberry Pi cần có các đặc điểm như sau (H. 5):

- Nhiệm vụ Main được thực hiện riêng biệt trên lõi số 0
- Ba lõi 1,2,3 được dùng để đọc tín hiệu của các cảm biến siêu âm, encoder, GyroScope, hồng ngoại.
- Các cảm biến siêu âm số 1, 3, 5 được đọc đồng thời để tránh hiện tượng đọc chéo giữa hai cảm biến cạnh nhau.
- Các cảm biến siêu âm số 2, 4, 6 được đọc đồng thời sau các cảm biến siêu âm số 1, 2, 3
- 2 encoder và GyroScope được đọc đồng thời. 2 Encoder được đọc đồng thời để đảm bảo thời gian đọc dữ liệu của 2 encoder là như nhau.
- 3 cảm biến hồng ngoại được đọc đồng thời.
- Các nhiệm vụ trên không được ngắt quãng (not preempted) khi đang thực hiện để thực hiện các nhiệm vụ khác, nếu không dữ liệu đọc được từ các cảm biến có thể bị mất.
- Thời gian để thực hiện nhiệm vụ đọc các cảm biến giống nhau được không chế bằng nhau.



H. 5 Tổ chức nhiệm vụ trên các lõi của Raspberry Pi 2

Thời gian để thực hiện toàn bộ các nhiệm vụ của robot được tính theo công thức:

$$T_{//} = \max\{C_{main}; C_{SR1} + C_{SR2} + C_{E1} + I_1; C_{SR2} + C_{SR4} + C_{E2} + C_{I2}; C_{SR5} + C_{SR6} + C_G + C_{I3}\} \quad (1)$$

Trong đó C_X : là thời gian lâu nhất để thực hiện nhiệm vụ X.

Đối với việc bộ điều khiển sử dụng các loại vi điều khiển thông thường hoặc MTN đơn nhân, thời gian $T_{//}$ để thực hiện toàn bộ các nhiệm vụ của robot sẽ bằng tổng tất cả thời gian thực hiện từng nhiệm vụ một do các nhiệm vụ này phải được thực hiện một cách tuần tự.

Do đó:

$$T_{//} = \sum_{i=1}^6 C_{SRi} + \sum_{i=1}^3 C_{Ii} + C_G + C_{E1} + C_{E2} + C_{Main} \quad (2)$$

Dễ dàng nhận thấy $T_{//}$ nhỏ hơn nhiều lần so với $T_{//}$, có nghĩa bộ điều khiển của robot sử dụng việc phân bố các nhiệm vụ trên các nhân của MTN như trên có thể giúp robot xử lý nhanh hơn nhiều so với khi bộ điều khiển của robot sử dụng vi điều khiển thông thường hoặc MTN đơn nhân.

Theo như phân tích về hai phương pháp phân bổ nhiệm vụ cho các nhân của MTN, trong nội dung của bài báo này, phương pháp phân bổ nhiệm vụ theo phân vùng thích hợp hơn so với phương pháp phân bổ nhiệm vụ toàn cục do yêu cầu cần thiết phải thực hiện việc đọc tín hiệu của nhiều cảm biến một cách song song trên các nhân khác nhau để tăng tốc độ xử lý của robot. Nếu sử dụng phương pháp phân bổ nhiệm vụ toàn cục, các nhiệm vụ sẽ được thực hiện trên các nhân khác nhau, dẫn đến việc không kiểm soát được các nhiệm vụ nào sẽ được thực hiện một cách song song, nhiệm vụ nào sẽ được thực hiện trước nhiệm vụ nào. Đối với phương pháp phân bổ nhiệm vụ theo phân vùng, các nhiệm vụ cố định được thực hiện trên một nhân cố định theo ý đồ của người lập trình. Do đó, việc bắt các nhiệm vụ đọc cảm biến siêu âm được thực hiện đồng thời trước các nhiệm vụ khác sẽ dễ dàng thực hiện hơn so với phương pháp phân bổ nhiệm vụ toàn cục.

Các nhiệm vụ của robot đều là các nhiệm vụ tuần hoàn (periodic), có nghĩa là cứ sau mỗi khoảng thời gian nhất định thì nó lại được thực hiện. Trong mỗi khoảng thời gian T nhất định đó, mỗi nhiệm vụ chỉ được phép thực hiện một lần. Trong chu kỳ T tiếp theo, nhiệm vụ Main sẽ sử dụng dữ liệu của các cảm biến đọc được ở trong chu kỳ ngay trước đó để sử dụng cho các tính toán của nhiệm vụ này. Do đó T phải đủ lớn để tất cả các nhiệm vụ trên mỗi nhân đủ thời gian để thực hiện.

3.3. Các kết quả thực nghiệm

Sau quá trình thử nghiệm, thời gian lâu nhất để thực hiện các nhiệm vụ đo được được cho trong bảng 3:

Bảng 3: Thời gian thực hiện các nhiệm vụ

Nhiệm vụ	Thời gian (s)
Main	0.01
SR1	0.018
E1	0.0013
I2	0.0017
G	0.0021

Thời gian lâu nhất thực hiện các nhiệm vụ trên các nhân được cho trong bảng 4:

Bảng 4: Thời gian thực hiện nhiệm vụ của các nhân

Nhiệm vụ	Thời gian(s)
Nhân 0	0.0103
Nhân 1	0.039043
Nhân 2	0.039116
Nhân 3	0.039127

Theo như công thức (1), kết hợp với quá trình thực nghiệm trên robot, chu kỳ được chọn để thực hiện các nhiệm vụ trên các nhân $T = 0,05s$. Vì thế tần số thực hiện của robot là 20Hz (có thể lớn hơn nếu chọn $0.039127s < T < 0.05s$) so với 5 - 8Hz như khi bộ điều khiển nguyên thủy có sử dụng vi điều khiển At89C51.

4. Kết luận

Trong nghiên cứu này một bộ điều khiển sử dụng MTN đa nhân Raspberry Pi 2 đã được phát triển cho robot tự hành ER-400 AGV. Việc tổ chức các nhiệm vụ trên các nhân của MTN đã cho phép tăng tốc độ xử lý của robot lên 3-4 lần so với bộ điều khiển nguyên thủy của ER-400 AGV. Điều này cho phép các tác vụ của robot được thực hiện với tần số cao hơn dẫn đến chất lượng tốt hơn, không bị một số lỗi như mất dữ liệu, bộ điều khiển bị treo sau một thời gian hoạt động.

Kết quả nghiên cứu tổ chức các nhiệm vụ trên các nhân của MTN không chỉ ứng dụng cho robot tự hành ER-400 AVG mà còn có thể ứng dụng cho các robot di động trong nhà khác, cũng như các ứng dụng đa nhiệm khác.

Lời cảm ơn

Các kết quả nghiên cứu trong bài báo này được thực hiện trong đề tài NCKH cấp Học viện KTQS năm 2014-2015. Các tác giả chân thành cảm ơn cơ quan quản lý các cấp và các đồng nghiệp đã có nhiều giúp đỡ quý báu trong quá trình thực hiện.

Tài liệu tham khảo

[1] Abusayeed Saifullah, Jing Li, Kunal Agrawal, Chenyang Lu, Christopher Gill: *Multi-core real-time scheduling for generalized parallel task models*, Article Real-Time Systems, Volume 49, Issue 4, pp 404-435, 2013.

[2] David R. Butenhof., *Programming with POSIX Threads*, Addison Wesley, 2008.

[3] James H. Anderson, John M. Calandrino, and UmaMaheswari C. Devi.: *Real-Time Scheduling on Multicore Platforms*, Proceedings of the Twelfth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06), 2006.

[4] Tăng Quốc Nam, Đào Văn Hiệp, Vũ Minh Đức: *Điều khiển robot tự hành bám quỹ đạo và tránh vật cản*. Tuyển tập công trình khoa học Hội nghị Cơ học toàn quốc lần thứ 8, Tập 1. Động lực học và điều khiển, tr 314-323, Hà Nội, 2007.

[5] Tăng Quốc Nam, Đào Văn Hiệp: *Xây dựng bản đồ môi trường cục bộ cho robot tự hành trong quá trình vận động từ dữ liệu cảm biến siêu âm*. Tuyển tập Hội nghị toàn quốc lần thứ 4 về Cơ điện tử, tr 266-272, 2008.

[6] Tăng Quốc Nam, Đào Văn Hiệp: *Ứng dụng điều khiển mờ trong bài toán tránh vật cản của robot tự hành dùng cảm biến siêu âm*. Tạp chí khoa học Giáo dục Kỹ thuật, Đại học Sư phạm Kỹ thuật thành phố Hồ Chí Minh số 7, tr 82-88, 2008.

[7] Tăng Quốc Nam, Đào Văn Hiệp: *Về một phương pháp định vị dựa trên dữ liệu tham chiếu dạng bản đồ lưới cho robot tự hành dùng cảm biến siêu âm*. Tạp chí Khoa học và Kỹ thuật, Học viện Kỹ thuật Quân sự, số 127, tr. 78-88, 2009.



TS. Tăng Quốc Nam tốt nghiệp đại học ngành *Cơ khí động lực* năm 1995 tại Học viện Kỹ thuật Quân sự, nhận bằng thạc sĩ về *Cơ điện tử* tại Viện Công nghệ châu Á (AIT) - Thái Lan năm 2002 và bằng tiến sĩ về *Cơ học kỹ thuật (Cơ điện tử)* tại Học viện Kỹ thuật Quân sự năm 2011. Từ năm 1995 đến nay là giảng viên bộ môn Robot đặc biệt và Cơ điện tử, khoa Hàng không vũ trụ, Học viện Kỹ thuật Quân sự. Các hướng nghiên cứu chính bao gồm: Mô hình hóa và mô phỏng các hệ động lực, Động lực học và điều khiển robot, Định vị và xây dựng bản đồ cho robot di động, Robot đặc biệt trong quân sự và thám hiểm, Thiết bị cơ điện tử trong quân sự.



Ths. Phạm Thế Hùng, tốt nghiệp đại học ngành Các hệ thống Robot kỹ thuật của vũ khí Hàng không năm 2009 tại trường Đại học Hàng không Matxcova (MAI), Liên bang Nga, nhận bằng thạc sĩ về chuyên ngành Hệ thống nhúng – ngành Các hệ thống Hàng không và Vũ trụ năm 2013 tại Học viện Hàng không và Vũ trụ (ISAE), Toulouse, Cộng hòa Pháp. Từ năm 2010 đến nay, là giảng viên tại bộ môn Robot đặc biệt và Cơ điện tử, khoa Hàng không vũ trụ, Học viện Kỹ thuật Quân sự. Các hướng nghiên cứu chính bao gồm: mobile robot, Embedded systems, Machine learning.

Ths. Nguyễn Bá Đại, tốt nghiệp đại học ngành Robot và hệ thống robot kỹ thuật năm 2011 tại Đại học Kỹ



thuật Tổng hợp Moskva mang tên Bauman- LB Nga. Từ năm 2012 đến nay là giảng viên tại bộ môn Robot đặc biệt và Cơ điện tử, khoa Hàng không Vũ trụ, Học viện Kỹ thuật Quân sự. Các hướng nghiên cứu chính bao gồm Cơ điện tử, Robot di động, Robot đặc biệt trong quân sự và và thị giác máy.

