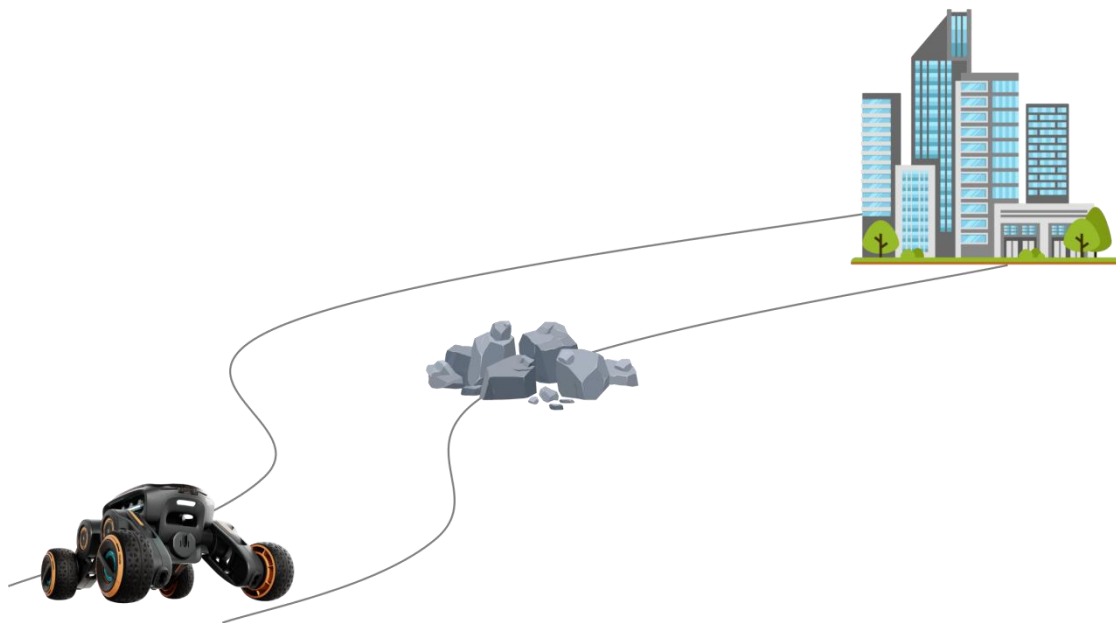


CHỦ ĐỀ 2: CHUYỂN ĐỘNG UGOT

Kế hoạch bay đã đạt được bước đột phá đáng kể nhờ sự chăm chỉ và nghiên cứu của nhiều nhà khoa học. Họ đã phát triển một siêu chip mạnh mẽ mà UGOT được chỉ định cung cấp để đảm bảo an ninh thông tin.



Hình 2.1 Xe biến hình UGOT cung cấp siêu vi mạch

I. Siêu UGOT

Để đảm bảo vận chuyển siêu chip thành công và chính xác đến đích, chúng ta điều khiển từ xa Xe biến hình UGOT bằng bộ điều khiển Bluetooth.



Hình 2.2 Bộ điều khiển Bluetooth

Để cho phép phân phối nhanh chóng và chính xác thông qua bộ điều khiển Bluetooth, chúng ta cần giải quyết các vấn đề sau:

1. Làm cách nào để lập trình điều khiển chuyển động của Xe biến hình UGOT?
2. Làm cách nào chúng ta có thể điều khiển từ xa các chuyển động của Xe biến hình UGOT thông qua bộ điều khiển Bluetooth?

II. Học tập vui vẻ

Kiến thức nền tảng

1. Đặc điểm của xe biến hình UGOT

Xe biến hình UGOT là phương tiện dẫn động bốn bánh có thể điều chỉnh độ cao của thân xe một cách độc lập để di chuyển qua các loại địa hình khác nhau. Nó có thể đi qua những lối đi có giới hạn độ cao, trèo qua rào chắn, leo bậc thang hoặc băng qua những địa hình bất thường có chênh lệch độ cao. Nó có khả năng off-road trên những con đường phức tạp.

	Trợ lực: 1. Hạ hoặc nâng thân Xe Biến hình UGOT. 2. Nâng hoặc hạ tay bánh xe của Xe biến hình UGOT.
---	--

	Động cơ: 1. Cho phép Xe biến hình UGOT di chuyển tiến, lùi, rẽ trái và rẽ phải ở các tốc độ khác nhau. 2. Xoay một hoặc nhiều bánh xe của Xe biến hình UGOT với tốc độ cụ thể.
---	---

2. Bluetooth và bộ điều khiển Bluetooth

Bluetooth là công nghệ giao tiếp không dây tầm ngắn cho phép kết nối và truyền dữ liệu giữa các thiết bị như điện thoại di động, máy tính và dàn âm thanh trong phòng khách. Công nghệ Bluetooth được sử dụng rộng rãi trong tai nghe, hệ thống tích hợp, nhà thông minh, v.v.

Ưu điểm của công nghệ Bluetooth:

Sự tiêu thụ ít điện năng: Chế độ năng lượng thấp của công nghệ Bluetooth giúp kéo dài thời gian sử dụng thiết bị một cách hiệu quả.

Dễ sử dụng: Kết nối Bluetooth có thể được hoàn thành bằng cách ghép nối đơn giản mà không yêu cầu quy trình thiết lập phức tạp.

Ứng dụng rộng rãi: Công nghệ Bluetooth được sử dụng rộng rãi trong điện thoại di động, máy tính, ô tô, TV thông minh và các thiết bị khác.

An toàn và độ tin cậy: Công nghệ Bluetooth có mức độ bảo mật nhất định, có thể mã hóa và bảo vệ dữ liệu được truyền đi để tránh rò rỉ dữ liệu.

Nhược điểm của công nghệ Bluetooth:

Khoảng cách truyền giới hạn: Vì công nghệ Bluetooth sử dụng giao tiếp tầm ngắn nên khoảng cách truyền tương đối hạn chế, thường trong phạm vi 10 mét.

Tốc độ truyền chậm hơn: So với các công nghệ không dây khác, công nghệ Bluetooth có

tốc độ truyền chậm hơn, điều này có thể dẫn đến thời gian truyền lâu đối với các tệp lớn. Bộ điều khiển Bluetooth là bộ điều khiển sử dụng công nghệ Bluetooth để kết nối không dây. Khi bộ điều khiển Bluetooth và Xe biển hình UGOT được kết nối, bộ điều khiển hoạt động như một bộ phát để gửi tín hiệu điều khiển và UGOT hoạt động như một bộ thu để nhận tín hiệu.



Hình 2.3 Giới thiệu bộ điều khiển Bluetooth

Các bước kết nối bộ điều khiển Bluetooth với UGOT

Bước 1: Bật chức năng Bluetooth của UGOT trong phần cài đặt. 	Bước 2: Nhấn và giữ nút bật/tắt của bộ điều khiển Bluetooth cho đến khi đèn báo sáng lên, cho biết đã bật thành công. 
Bước 3: Nhấn và giữ nút kết nối Bluetooth; đèn báo nhấp nháy nhanh cho biết quá trình ghép nối đang được tiến hành. 	Bước 4: Đưa bộ điều khiển Bluetooth lại gần bộ điều khiển chính của UGOT cho đến khi đèn báo vẫn sáng, cho biết bộ điều khiển đã được ghép nối thành công. 

Kiến thức lập trình

1. Xe biến hình tiến/lùi

Chúng ta có thể sử dụng phương thức '**transform_move_speed(direction, speed)**' để điều khiển Xe biến hình UGOT di chuyển tiến hoặc lùi ở một tốc độ cụ thể. Các chi tiết như sau:

Phương thức: **Transform_move_speed(direction, speed)**

Chức năng: điều khiển Xe biến hình UGOT di chuyển tiến hoặc lùi với tốc độ xác định

Thông số:

direction: hướng; số 0 biểu thị tiến và số 1 biểu thị lùi

speed: tốc độ chuyển động; phạm vi: 5–80 cm/s

2. Xe biến hình rẽ trái/phải

Chúng ta có thể sử dụng phương thức '**transform_turn_speed(turn, speed)**' để điều khiển Xe biến hình UGOT rẽ trái hoặc phải ở một tốc độ cụ thể. Các chi tiết như sau:

Phương thức: **Transform_turn_speed(xoay, tốc độ)**

Chức năng: điều khiển Xe biến hình UGOT rẽ trái hoặc phải với tốc độ xác định

Thông số:

turn: hướng; số 2 là rẽ trái, số 3 là rẽ phải

speed: tốc độ quay; phạm vi: 5–280 độ/s

3. Chuyển đổi điểm dừng xe

Chúng ta có thể sử dụng phương thức '**transform_stop()**' để điều khiển Xe biến hình UGOT ngừng chuyển động. Các chi tiết như sau:

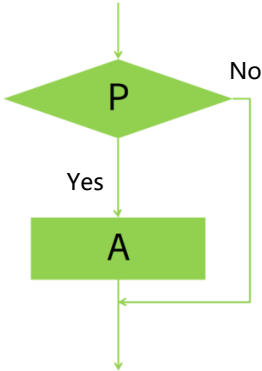
Phương thức: Transform_stop()

Chức năng: điều khiển Xe biến hình UGOT ngừng chuyển động

4. Cấu trúc nhánh

Trong ngôn ngữ lập trình Python, cấu trúc nhánh kiểm soát luồng chương trình. Các câu lệnh phán đoán có điều kiện có thể được sử dụng để thực thi các khối mã khác nhau dựa trên các điều kiện cụ thể.

Dưới đây là ba cấu trúc nhánh phổ biến:

Cấu trúc nhánh	Dạng viết	Sơ đồ	Giới thiệu
Phán đoán có điều kiện một nhánh	biểu thức 'nếu' P: Khối mã A		Nếu điều kiện P được đáp ứng, thực thi khối mã A, nếu không thì bỏ qua khối mã A và đi xuống.

Phán đoán có điều kiện nhánh đôi	biểu thức 'nếu' P: Khối mã A khác: Khối mã B	<pre> graph TD Start(()) --> P{P} P -- Yes --> A[A] P -- No --> B[B] A --> Exit(()) B --> Exit </pre>	Nếu điều kiện P được đáp ứng, khối mã A được thực thi và nếu không, khối mã B được thực thi.
Phán đoán có điều kiện nhiều nhánh	biểu thức 'nếu' P1: Khối mã A biểu thức 'elif' P2: Khối mã B khác: Khối mã C	<pre> graph TD Start(()) --> P1{P1} P1 -- Yes --> A[A] P1 -- No --> P2{P2} P2 -- Yes --> B[B] P2 -- No --> C[C] A --> Exit(()) B --> Exit C --> Exit </pre>	Sự phán xét được thực hiện từ trên xuống dưới; khi một trong các điều kiện được thực thi, các điều kiện khác sẽ không được thực thi.

Dưới đây là một số lưu ý phổ biến khi sử dụng cấu trúc nhánh trong Python:

(1) Thụt lề: Trong cấu trúc nhánh, cần đảm bảo rằng các khối mã bên dưới câu lệnh điều kiện được thụt lề chính xác. Việc thụt lề thường được thực hiện bằng cách sử dụng bốn dấu cách hoặc dấu cách Tab. Việc thụt lề không chính xác có thể dẫn đến lỗi cú pháp hoặc lỗi logic mã.

(2) Biểu thức điều kiện: Trong cấu trúc nhánh, cần đảm bảo rằng kết quả của biểu thức điều kiện là giá trị Boolean (Đúng hoặc Sai).

(3) Dễ đọc và đơn giản: Mã tốt phải có tính năng dễ đọc và đơn giản. Khi viết cấu trúc nhánh, hãy cố gắng sử dụng các điều kiện và khối mã rõ ràng và ngắn gọn, đồng thời tránh logic quá phức tạp và mã dài dòng.

5. Nhận thông tin nút nhấn

Chúng ta có thể sử dụng phương thức '`get_joypad_pressing_buttons()`' để lấy thông tin về nút được nhấn trên bộ điều khiển Bluetooth. Các chi tiết như sau:

Phương thức: `get_joypad_pressing_buttons()`
Chức năng: lấy thông tin về nút được nhấn trên bộ điều khiển Bluetooth. Giá trị trả về là một **danh sách**, ví dụ: `[]`, `['X']`

Khái niệm về danh sách: Trong Python, danh sách là một tập hợp các phần tử được sắp xếp theo thứ tự, được phân tách bằng dấu phẩy và được xác định bằng `[]`.

Ví dụ về các danh sách phổ biến: danh sách trống `[]`, danh sách số nguyên `[1, 2, 3]` và danh sách chuỗi `['a', 'b']`

6. Toán tử so sánh

Toán tử so sánh trong Python được sử dụng để so sánh hai giá trị về đẳng thức hoặc kích thước. Khi so sánh hai giá trị, kết quả của toán tử so sánh là giá trị Boolean (Đúng hoặc Sai). Các toán tử so sánh phổ biến được sử dụng trong Python được liệt kê bên dưới:

Biểu tượng	Tên	Giới thiệu
<code>==</code>	Bình đẳng	Kiểm tra xem hai giá trị có bằng nhau không
<code>!=</code>	Không công bằng	Kiểm tra nếu hai giá trị không bằng nhau
<code>></code>	Lớn hơn	Kiểm tra xem giá trị bên trái có lớn hơn giá trị bên phải không
<code>>=</code>	Lớn hơn hoặc bằng	Kiểm tra xem giá trị bên trái có lớn hơn hoặc bằng giá trị bên phải không
<code><</code>	Ít hơn	Kiểm tra xem giá trị bên trái có nhỏ hơn giá trị bên phải không

<=	Ít hơn hoặc bằng	Kiểm tra xem giá trị bên trái có nhỏ hơn hoặc bằng giá trị bên phải không
----	------------------	---

'==' trong bài được dùng để kiểm tra xem hai giá trị có bằng nhau hay không. Ví dụ: a==b biểu thị việc đánh giá xem giá trị của a có bằng giá trị của b hay không, trong khi a=b biểu thị việc gán giá trị của b cho a. Trong Python, '=' là toán tử gán.

7. Vòng lặp 'while'

Cấu trúc vòng lặp	Dạng viết	Sơ đồ
while	biểu thức 'while' P: Khối mã A	<pre> graph TD Entry(()) --> P{P} P -- Yes --> A[A] A --> P P -- No --> Exit(()) </pre>

Vòng lặp 'while' lặp lại một đoạn của khối mã khi đáp ứng một số điều kiện nhất định. Lấy sơ đồ làm ví dụ. Trong quá trình thực hiện, một phán đoán được đưa ra xem liệu điều kiện P có đúng hay không. Nếu điều kiện đúng, khối mã A được thực thi và sau đó đánh giá lại điều kiện P cho đến khi nó sai, kết thúc vòng lặp.

Để ngăn vòng lặp vô hạn, cần đảm bảo rằng điều kiện vòng lặp có thể được thay đổi thành Sai tại một số điểm hoặc có thể chấm dứt vòng lặp theo cách thủ công bằng câu lệnh break bên trong vòng lặp.

III. Công xưởng sáng tạo

Với những kiến thức đã học, chúng ta hãy lập trình điều khiển từ xa các chuyển động của Xe biển hình UGOT bằng bộ điều khiển Bluetooth.

1. Nhiệm vụ 1

Tổng quan nhiệm vụ: Chương trình điều khiển chuyển động của Xe biến hình UGOT.

```
1 #Initialisation
2 from ugot import ugot
3 import time
4 got=ugot.UGOT()
5 got.initialize("192.168.1.115")
6
7 got.transform_move_speed(0,20) #The UGOT Transforming Car moves forwards at a speed of 20 cm/s
8 time.sleep(5)                 #Wait 5 seconds
9 got.transform_move_speed(1,20) #The UGOT Transforming Car moves backwards at a speed of 20 cm/s
10 time.sleep(5)
11 got.transform_turn_speed(2,40) #The UGOT Transforming Car turns left at a speed of 40 degrees/s
12 time.sleep(5)
13 got.transform_turn_speed(3,40) #The UGOT Transforming Car turns right at a speed of 40 degrees/s
14 time.sleep(5)
15 got.transform_stop()          #The UGOT Transforming Car stops
```

Hình 2.4 Chương trình demo của Nhiệm vụ 1

2. Nhiệm vụ 2

Tổng quan nhiệm vụ: Chương trình thực hiện điều khiển chuyển động Xe biến hình UGOT bằng bộ điều khiển Bluetooth.

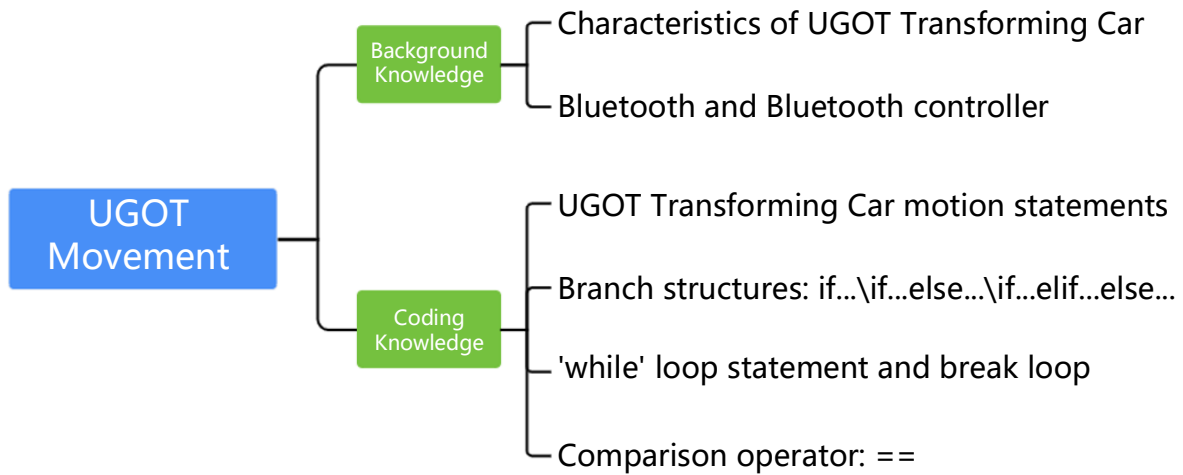
```

1  #Initialisation
2  from ugot import ugot
3  got=ugot.UGOT()
4  got.initialize("192.168.1.115")
5
6  while True:  #Infinite loop
7      a=got.get_joypad_pressing_buttons()
8
9
10     if a==['X']:
11         got.transform_move_speed(0,20)
12     if a==['B']:
13         got.transform_move_speed(1,20)
14     if a==['Y']:
15         got.transform_turn_speed(2,40)
16     if a==['A']:
17         got.transform_turn_speed(3,40)
18     if a==['R2']:
19         got.transform_stop()
20         break  #Break loop

```

Hình 2.5 Chương trình demo cho Nhiệm vụ 2

IV. Tóm tắt kiến thức



Hình 2.6 Tóm tắt kiến thức