

CHỦ ĐỀ 1: KHÁM PHÁ SƠ BỘ CỦA UGOT

Trong tương lai xa, cộng đồng loài người trên Trái đất sẽ bước vào kỷ nguyên của những công nghệ tiên tiến cao. Trong thời đại này, con người đã chinh phục mọi góc ngách trên trái đất. Các thành phố lớn tràn ngập những tòa nhà chọc trời, giao thông tấp nập và cơ sở công nghệ tiên tiến. Tuy nhiên, Trái đất không còn có thể thỏa mãn ham muốn khám phá của con người.

Một ngày nọ, một nhóm các nhà khoa học và kỹ sư tập hợp lại và nghĩ ra một kế hoạch đầy tham vọng: Kế hoạch bay. Họ hình dung việc mở rộng hoạt động khám phá của con người đến rìa không gian để khám phá những bí ẩn của vũ trụ.

UGOT tham gia Kế hoạch bay với tư cách là trợ lý robot để khám phá không gian, hỗ trợ các nhà nghiên cứu khám phá Mặt trăng và Sao Hỏa, cũng như bảo trì căn cứ.

Siêu UGOT

Các kỹ sư kỹ thuật sẽ hợp tác chặt chẽ với UGOT, công ty có khả năng AI. Họ phải làm quen với các dạng UGOT khác nhau, các chức năng khác nhau, phương pháp hoạt động cơ bản và hiểu biết về công nghệ AI.



Hình 1.1 Xe biến hình UGOT

Để kiểm tra chức năng hiển thị của UGOT và đảm bảo rằng các nhà nghiên cứu có thể đọc thông tin do UGOT gửi một cách kịp thời, cần phải hoàn thành một số nhiệm vụ nhất định.

1. Tìm hiểu về các mô-đun chính của Xe biến hình UGOT và hoàn thành việc lắp ráp.
2. Kiểm tra chức năng hiển thị của UGOT bằng phần mềm lập trình uPython.

Học tập vui vẻ

Kiến thức nền tảng

Trí tuệ nhân tạo là gì?

Trí tuệ nhân tạo đề cập đến khả năng của một cỗ máy bắt chước trí thông minh của con người thông qua các phương tiện nhân tạo, cho phép nó nhận thức, suy nghĩ và đưa ra quyết định giống như con người.

UGOT có thể thực hiện các chức năng nhận dạng khuôn mặt, định vị âm thanh, trò chuyện bằng giọng nói, tránh chướng ngại vật bằng laser, tuần tra đường đơn và đường đôi, xử lý đối tượng, v.v. Nó có thể cung cấp phản hồi theo cách tương tự như trí thông minh của con người và do đó được phân loại là một robot thông minh.

Python là gì?

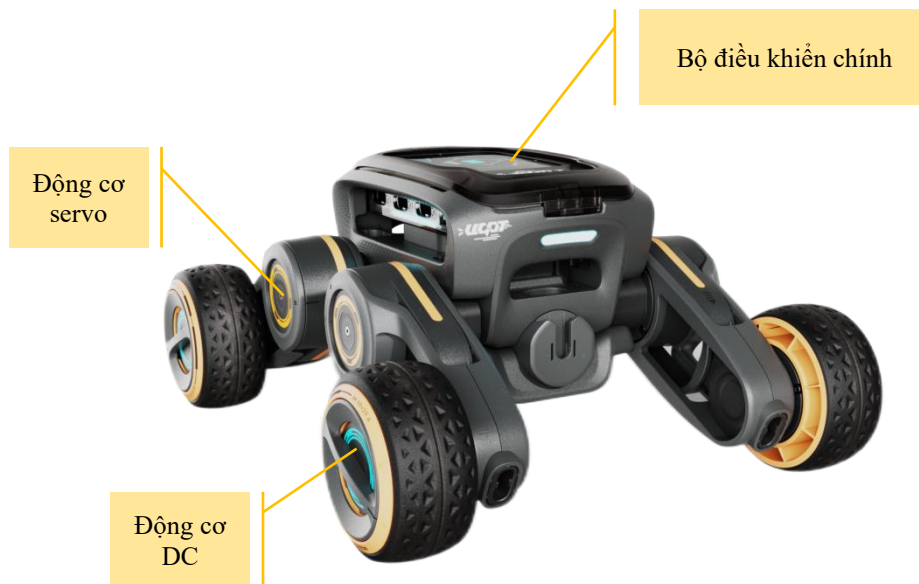
Python là một trong những ngôn ngữ lập trình AI phổ biến và được sử dụng rộng rãi nhất. Nó cung cấp cho các nhà phát triển một bộ công cụ và khung toàn diện để xây dựng và đào tạo các mô hình học máy, chẳng hạn như NumPy, Pandas và TensorFlow, với cú pháp dễ hiểu.

Nguồn gốc: Được tạo bởi Guido van Rossum ở Hà Lan vào năm 1989, Python 0.9.0 được phát hành vào năm 1991.

Phát triển: Python 1.0 được phát hành vào năm 1994, tiếp theo là một loạt phiên bản 2.x. Python ngày càng trở nên phổ biến đối với người dùng và nhà phát triển theo thời gian. Cộng đồng đã phát triển nhiều thư viện và công cụ hữu ích của bên thứ ba.

Hiện trạng: Python hiện được sử dụng trong nhiều lĩnh vực, bao gồm phát triển web, phân tích dữ liệu, trí tuệ nhân tạo, tính toán khoa học và tự động hóa. Python có một cộng đồng lớn và năng động. Các thành viên cộng đồng đã cam kết phát triển nhiều thư viện và công cụ của bên thứ ba nhằm mở rộng khả năng và ứng dụng của Python.

Các mô-đun chính của Xe biến hình UGOT



Hình 1.2 Các module chính của Xe biến hình UGOT

STT	Tên	Chức năng	Sự giống nhau
1	Bộ điều khiển chính	Lưu trữ và thực thi chương trình điều khiển để hoàn thành việc điều khiển UGOT	Bộ não con người
2	Động cơ DC	Dẫn động chuyển động quay của bánh xe	Tay người
3	Động cơ servo	Dẫn động cánh tay rotor đến một góc cụ thể	Khớp người

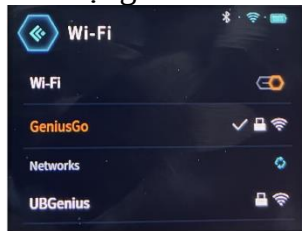
Chế tạo một chiếc xe biến hình UGOT

Để chế tạo Xe biến hình UGOT, hãy tham khảo video hoặc hướng dẫn xây dựng 3D.



Hình 1.3 Sơ đồ xe biến hình UGOT

Thiết lập kết nối mạng Xe biến hình UGOT

Bước 1: Nhấn và giữ công tắc nguồn ở phía dưới Xe biến hình UGOT để bật. 	Bước 2: Đợi màn hình hiển thị bộ điều khiển chính mở, chọn mẫu Xe biến hình UGOT và nhấn OK. 
Bước 3: Bấm vào Cài đặt và chọn mạng Wi-Fi để kết nối. 	Bước 4: Kết nối Wi-Fi thành công xác nhận rằng Xe biến hình UGOT đã được kết nối với mạng. 

Kiến thức lập trình

1. Câu lệnh **print**

Câu lệnh **print** chủ yếu được sử dụng để in ra các số, chuỗi, v.v. Văn bản in có thể không chỉ bao gồm các ký tự tiếng Anh mà còn cả các ký hiệu đặc biệt. Khi in văn bản cần phải đặt nội dung văn bản trong dấu ngoặc kép.

Dấu ngoặc kép có thể là dấu ngoặc đơn (' '), dấu ngoặc kép (") hoặc dấu ngoặc kép ba (""").

Trong Python, một chuỗi (str) dùng để chỉ văn bản được đặt trong một cặp dấu ngoặc kép. Chuỗi là kiểu dữ liệu được sử dụng phổ biến nhất trong Python.

Dữ liệu là cơ sở của tất cả các chương trình và được phân loại thành các loại dữ liệu bất biến và có thể thay đổi dựa trên việc loại dữ liệu đó có thể thay đổi hay không.

2. Mã để nhập hàm **ugot** từ mô-đun **UGOT**

Chúng ta có thể sử dụng **'from ugot import ugot'** để triển khai việc nhập các hàm **ugot** trong mô-đun **UGOT**.

3. Mã để nhập **UGOT**

Chúng ta có thể sử dụng **'got = ugot.UGOT()'** để nhập **UGOT** và gán nó cho **got**.

4. Mã để kết nối **UGOT**

Chúng ta có thể kết nối với **UGOT** bằng địa chỉ IP. Chúng ta sử dụng mã **'got.initialize(device_ip)'** trong đó tham số **device_ip** thuộc loại str. Điền địa chỉ IP của thiết bị dưới dạng chuỗi để kết nối với **UGOT**.

5. Mã để kích hoạt màn hình bộ điều khiển chính hiển thị màu nền

Chúng ta có thể sử dụng phương pháp **'screen_display_background(color)'** để cho phép màn hình bộ điều khiển chính hiển thị màu nền. Các chi tiết như sau:

Phương thức: `screen_display_background(color)`

Chức năng: Màn hình điều khiển chính hiển thị màu nền

Thông số:

color: màu sắc, kiểu int, phạm vi giá trị: [0-8]

0: Đen; 1: Trắng; 2: Tím; 3: Đỏ; 4: Cam; 5: Vàng; 6: Xanh lục; 7: Xanh lơ; 8: Xanh lam

6. Mã để cho phép màn hình bộ điều khiển chính in văn bản

Chúng ta có thể sử dụng phương thức '`screen_print_text(text, color)`' để cho phép màn hình bộ điều khiển chính in văn bản. Các chi tiết như sau:

Phương thức: `screen_print_text(text, color)`

Chức năng: Màn hình điều khiển chính in văn bản

Thông số:

text: kiểu str, nội dung văn bản cần in

color: màu sắc, kiểu int, phạm vi giá trị: [0-8]

0: Đen; 1: Trắng; 2: Tím; 3: Đỏ; 4: Cam; 5: Vàng; 6: Xanh lục; 7: Xanh lơ; 8: Xanh lam

7. Mã để cho phép màn hình bộ điều khiển chính in dòng văn bản và xuống dòng

Chúng ta có thể sử dụng phương thức '`screen_print_text_newline(text, color)`' để cho phép màn hình bộ điều khiển chính in dòng văn bản và xuống dòng. Các chi tiết như sau:

Phương thức: `screen_print_text_newline(text, color)`

Chức năng: Màn hình điều khiển chính in dòng văn bản và xuống dòng

Thông số:

text: kiểu str, nội dung văn bản cần in

color: màu sắc, kiểu int, phạm vi giá trị: [0-8]

0: Đen; 1: Trắng; 2: Tím; 3: Đỏ; 4: Cam; 5: Vàng; 6: Xanh lục; 7: Xanh lơ;

8: Xanh lam

8. Mã để cho phép xóa nội dung và màu nền của màn hình bộ điều khiển chính

'`got.screen_clear()`' xóa nội dung và màu nền của màn hình bộ điều khiển chính.

9. Mã để nhập mô-đun thời gian

'`import time`' nhập mô-đun thời gian.

import: dùng để nhập mô-đun. Định dạng: '`import module_name`', chẳng hạn như: `import tu` và `import random`.

10. Mã để giữ trạng thái hiện tại

'`time.sleep(t)`' giữ trạng thái hiện tại cho đến số giây nhất định.

t: thời lượng tính bằng giây, kiểu int/kiểu float

Công xưởng sáng tạo

Với những kiến thức mà chúng ta đã có được, hãy cùng khám phá thế giới UGOT và Python thông qua phần mềm lập trình uPython nhé!

Lập trình

1. Nhiệm vụ 1

Tổng quan nhiệm vụ: Tìm hiểu về phần mềm lập trình uPython và chương trình hiển thị văn bản trong khu vực xuất của uPython.

<pre>1 print(12345) 2 print(12+13) 3 print(99-23) 4 print(12*12) 5 print(12/3)</pre>	<pre>print('Hello, I am UGOT.') print('What can I do for you? ')</pre>
	<pre>print('''Hello, I am UGOT. What can I do for you? ''')</pre>

Hình 1.4 Chương trình demo cho Nhiệm vụ 1

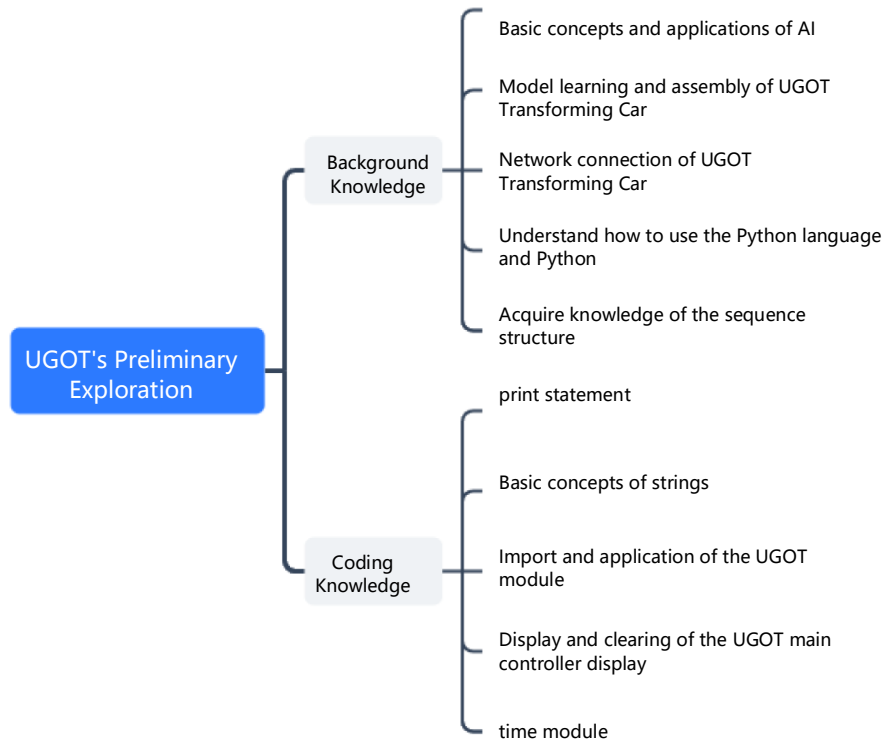
2. Nhiệm vụ 2

Tổng quan về nhiệm vụ: Gọi mô-đun UGOT để hiển thị văn bản trên màn hình bộ điều khiển chính.

```
1 from ugot import ugot          #Import the ugot function from the UGOT module
2 import time                    ## Import time
3 got=ugot.UGOT()                #Import UGOT
4 got.initialize("192.168.1.119") #Connect UGOT to the IP address
5
6 got.screen_display_background(0) #The main control display is black
7 got.screen_print_text_newline("Hello, I am UGOT.",1) #Print white text and line feed on the main control display
8 got.screen_print_text_newline("What can I do for you?",1)
9 time.sleep(1)                  #Keep the current state for 1 second
10 got.screen_clear()             #Clear the main control display
```

Hình 1.5 Chương trình demo cho Nhiệm vụ 2

Tóm tắt kiến thức



Hình 1.6 Tóm tắt kiến thức