

CHỦ ĐỀ 3: CHUẨN BỊ SẴN SÀNG

Việc chuyển giao thành công các siêu chip đã tạo nền tảng vững chắc cho việc thực hiện Kế hoạch bay. Hiện tại, UGOT đang chuẩn bị dần thân vào sứ mệnh không gian nhằm khám phá bí ẩn của Mặt Trăng và Sao Hỏa.



Hình 3.1 Vụ phóng tên lửa

I. Siêu UGOT

Để đảm bảo sự thành công của các sứ mệnh thám hiểm Mặt trăng và Sao Hỏa của UGOT, các cuộc thử nghiệm cuối cùng phải được tiến hành trước khi phóng tên lửa.

Các thử nghiệm này bao gồm kiểm tra màn hình, điều khiển chuyển động và chức năng cũng như kiểm tra mô phỏng trên bốn hệ thống chức năng quan trọng của UGOT: đèn điều khiển chính, giọng nói, động cơ và servo. Mục đích của những cuộc thử nghiệm này là để đảm bảo Kế hoạch bay được khởi động suôn sẻ.

Để thực hiện kiểm tra chức năng mô phỏng của đèn điều khiển chính, giọng nói, động cơ và servo của UGOT, cần giải quyết các vấn đề sau:

1. Làm cách nào để chúng ta chọn các chức năng cần thử nghiệm?

2. Làm cách nào chúng ta có thể mô phỏng thử nghiệm và dừng thử nghiệm dựa trên lựa chọn?

II. Học tập vui vẻ

Kiến thức nền tảng

1. Kiểm tra chức năng UGOT



Hình 3.2 Xe biến hình UGOT

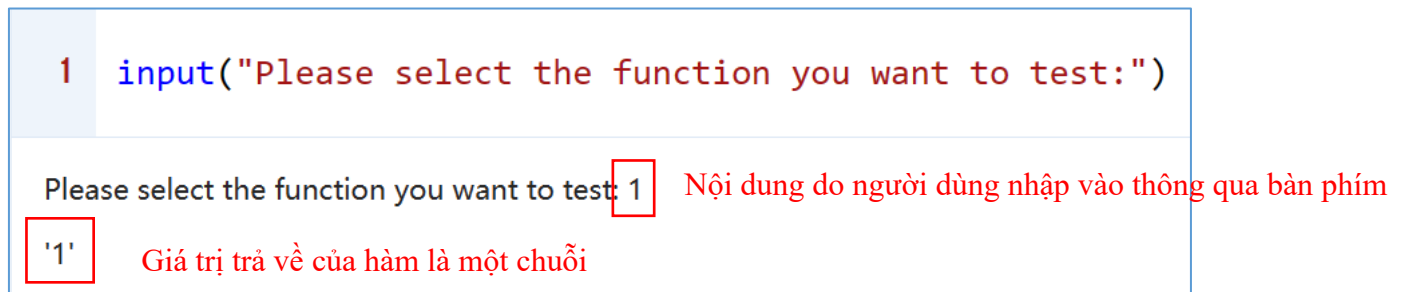
1. Màn hình hiển thị: Màn hình điều khiển chính hỗ trợ hiển thị văn bản và màu nền.
2. Hiệu ứng thanh đèn của bộ điều khiển chính: điều khiển các thanh đèn của bộ điều khiển chính sáng lên theo màu được chỉ định hoặc tắt.
3. TTS speech: phát nội dung văn bản nhập thông qua các bài phát biểu.
4. Động cơ: điều khiển một hoặc nhiều động cơ quay với tốc độ xác định.
5. Servo: điều khiển một hoặc nhiều servo quay theo một góc xác định.

Kiến thức lập trình

1. Hàm input()

input() là một hàm dựng sẵn trong Python để nhận dữ liệu nhập của người dùng. Đầu vào sẽ được sử dụng làm giá trị trả về của hàm.

Chương trình minh họa:

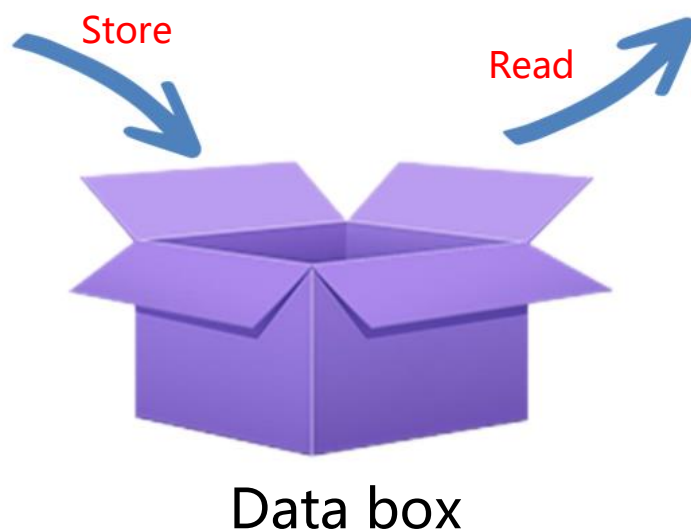


Hình 3.3 Kiểm tra hàm input()

Lưu ý rằng dữ liệu mà hàm input() thu được luôn có kiểu chuỗi. Nếu muốn chuyển đổi dữ liệu sang loại khác (chẳng hạn như số nguyên, số dấu phẩy động, v.v.), hãy sử dụng hàm thích hợp.

2. Biến

Trong Python, một biến có thể được coi là một hộp chứa dữ liệu. Dữ liệu được lấy ra để so sánh mỗi lần sử dụng và dữ liệu mới sẽ được lưu trữ khi thích hợp ở cuối.



Hình 3.4 Giới thiệu các biến

Nếu chúng ta muốn sử dụng biến được lưu trong máy tính thì có thể dùng tên biến để lấy. Các quy tắc sau đây áp dụng cho việc đặt tên biến:

- (1) Tên biến phải bắt đầu bằng một chữ cái hoặc dấu gạch dưới, không được bằng số.

Ví dụ: 'student1' là hợp lệ nhưng '1student' thì không.

- (2) **Tên biến chỉ được chứa chữ cái, số, ký tự và dấu gạch dưới.** Không được phép sử dụng các ký tự đặc biệt (@, ¥, #, v.v.).
- (3) **Tên biến có phân biệt chữ hoa chữ thường.** Ví dụ: 'Student' và 'student' là hai biến khác nhau.
- (4) **Không thể sử dụng từ khóa Python làm tên biến.** Ví dụ: tránh sử dụng các từ như 'for', 'if' và 'else'.
- (5) **Tên biến nên mô tả cách sử dụng hoặc nội dung của biến càng nhiều càng tốt.** Ví dụ: để tạo một biến lưu trữ tuổi của người dùng, bạn có thể sử dụng tuổi làm tên biến.
- (6) Khi tên biến bao gồm hai từ trở lên, có thể sử dụng dấu gạch dưới để phân tách chúng. Ví dụ: my_age.

Chương trình minh họa:

```
1 age=10
2 print(age)
```

Tạo biến 'tuổi' và gán cho nó một giá trị

Gọi biến 'tuổi'

10

Xuất kết quả

Hình 3.5 Chương trình minh họa biến

3. Màn hình màu thanh đèn điều khiển chính

Chúng ta có thể sử dụng phương pháp '**show_light_rgb(lights, red, green, blue)**' để chiếu sáng một màu nhất định của đèn điều khiển chính. Các chi tiết như sau:

Phương thức: `show_light_rgb(lights, red, green, blue)`

Chức năng: điều khiển ánh sáng của một màu nhất định trên thanh ánh sáng của đèn điều khiển chính

Thông số:

`lights`: danh sách thanh đèn [0-3]; [0] đại diện cho thanh ánh sáng phía trên, [1] đại diện cho thanh ánh sáng bên trái, [2] đại diện cho thanh ánh sáng bên phải, [3] đại diện cho thanh ánh sáng phía dưới và [0, 1] đại diện cho thanh ánh sáng phía trên và bên trái.

`red, green, blue`: ba màu cơ bản của ánh sáng (đỏ, lục và lam); giá trị của mỗi màu nằm trong khoảng từ 0 đến 255; các tỷ lệ khác nhau của các giá trị đỏ, lục và lam có thể tạo ra bất kỳ màu nào.

4. Tắt đèn điều khiển chính

Chúng ta có thể sử dụng phương thức '**turn_off_lights()**' để tắt tất cả các đèn của bộ điều khiển chính. Các chi tiết như sau:

Phương thức: `turn_off_lights()`

Chức năng: tắt tất cả các đèn điều khiển chính

5. Phát bài phát biểu

Chúng ta có thể phát các bài phát biểu TTS bằng phương thức '**play_audio_tts(data,**

voice_type, wait)'. Các chi tiết như sau:

Phương thức: **play_audio_tts(data, voice_type, wait)**

Chức năng: phát bài phát biểu TTS

Thông số:

data: kiểu chuỗi; bạn có thể viết nội dung để phát

voice_type: âm sắc của bài phát biểu sẽ được phát, với 0 đại diện cho giọng nữ và 1 đại diện cho giọng nam

wait: kiểu boolean; nó đánh giá xem có bị chặn hay không; các giá trị bao gồm Đúng và Sai;

giá trị mặc định là Sai, biểu thị trạng thái không chặn.

6. Quay động cơ

Chúng ta có thể sử dụng phương thức **'transform_motor_control(lf, rf, lb, rb)'** để điều khiển chuyển động quay của động cơ của Xe biến hình UGOT. Các chi tiết như sau:

Phương thức: `Transform_motor_control(lf, rf, lb, rb)`

Chức năng: điều khiển chuyển động quay của động cơ Xe biển hình UGOT

Thông số:

lf,rf,lb,rb: lf biểu thị tốc độ của bánh trước bên trái, rf biểu thị tốc độ của bánh trước bên phải, lb biểu thị tốc độ của bánh sau bên trái và rb biểu thị tốc độ của bánh sau bên phải. Phạm vi tốc độ là -360 đến 360 vòng/phút.

7. Xoay servo

Chúng ta có thể sử dụng phương thức '**transform_arm_control(joint, location, time)**' để điều khiển chuyển động quay của các động cơ servo của Xe biển hình UGOT. Các chi tiết như sau:

Phương thức: Transform_arm_control(joint, location, time)

Chức năng: điều khiển chuyển động quay của các servo của Xe biến hình UGOT

Thông số:

joint: tay bánh xe; kiểu int; số 1 tượng trưng cho cánh tay trước bên trái, số 2 tượng trưng cho cánh tay sau bên trái, số 3 tượng trưng cho cánh tay sau bên phải và số 4 tượng trưng cho cánh tay trước bên phải.

location: góc quay servo; phạm vi góc: -180 đến 180 độ

time: thời gian quay của servo, tính bằng ms

III. Công xưởng sáng tạo

Với những kiến thức đã có được, chúng ta hãy cùng nhau kiểm tra các chức năng của UGOT nhé!

1. Nhiệm vụ 1

Tổng quan nhiệm vụ: Chương trình thực hiện việc lựa chọn các chức năng cần kiểm thử.

```

1 #Initialisation
2 from ugot import ugot
3 u=ugot.UGOT()
4 u.initialize("192.168.11.77")
5 u.transform_restory()
6 u.screen_display_background(0) #Set the background of the main control display to black
7 u.transform_set_chassis_height(3) #The UGOT Transforming Car is 3 cm high
8
9 while True: #Loop
10     x=input("Select a function you want to detect: 1/Light, 2/Speech, 3/Motor, 4/Servo, Others/Exit:")
11     #Use the input() function to obtain what the user inputs through the keyboard
12     #The X variable stores the unknown content input by the user
13
14     if x=="1": #Compare the input content with the target
15         u.screen_print_text_newline("Detection of main controller light",1) #Show the text in white on the main control display
16     elif x=="2":
17         u.screen_print_text_newline("Speech detection",1)
18     elif x=="3":
19         u.screen_print_text_newline("Motor detection",1)
20     elif x=="4":
21         u.screen_print_text_newline("Servo detection",1)
22     else:
23         u.screen_print_text_newline("Detection completed!",1)
24         u.screen_clear() #Clear the main control display
25         break

```

Hình 3.6 Chương trình demo của Nhiệm vụ 1

2. Nhiệm vụ 2

Tổng quan nhiệm vụ: Chương trình thực hiện các thử nghiệm mô phỏng khác nhau của các chức năng tương ứng theo lựa chọn.

```

1 #Initialisation
2 from ugot import ugot
3 u=ugot.UGOT()
4 u.initialize("192.168.11.77")
5 u.transform_restory()
6 u.screen_display_background(0) #Set the background of the main control display to black
7 u.transform_set_chassis_height(3) #The UGOT Transforming Car is 3 cm high
8
9 while True:
10     x=input("Select a function you want to detect: 1/Light, 2/Speech, 3/Motor, 4/Servo, Others/Exit:")
11     if x=="1":
12         u.show_light_rgb([0,1,2,3], 255, 0, 0) #Four light bars light up in red
13         u.screen_print_text_newline("The main controller lights function normally!",1)
14     elif x=="2":
15         u.play_audio_tts("Hello!",1,True) #Say 'Hello' with a male voice
16         u.screen_print_text_newline("Speech function is normal!",1)
17     elif x=="3":
18         u.transform_motor_control(80, 80, 80, 80) #Control the four motors of the Transforming Car to rotate
19         u.screen_print_text_newline("Motors function normally!",1)
20     elif x=="4":
21         u.transform_arm_control(1, 30, 10) #Control the four servos of the Transforming Car to rotate
22         u.screen_print_text_newline("Servos function normally!",1)
23     else:
24         u.screen_print_text_newline("Detection completed!",1)
25         u.screen_clear() #Clear the main control display
26         u.turn_off_lights() #Turn off all lights
27         u.transform_stop() #The Transforming Car stops
28         break

```

Hình 3.7 Chương trình demo cho Task 2

IV. Tóm tắt kiến thức

