Topic 2 UGOT Movement

The Flying Plan has achieved a significant breakthrough thanks to the hard work and research of many scientists. They have developed a powerful superchip, which UGOT has been designated to deliver to ensure information security.

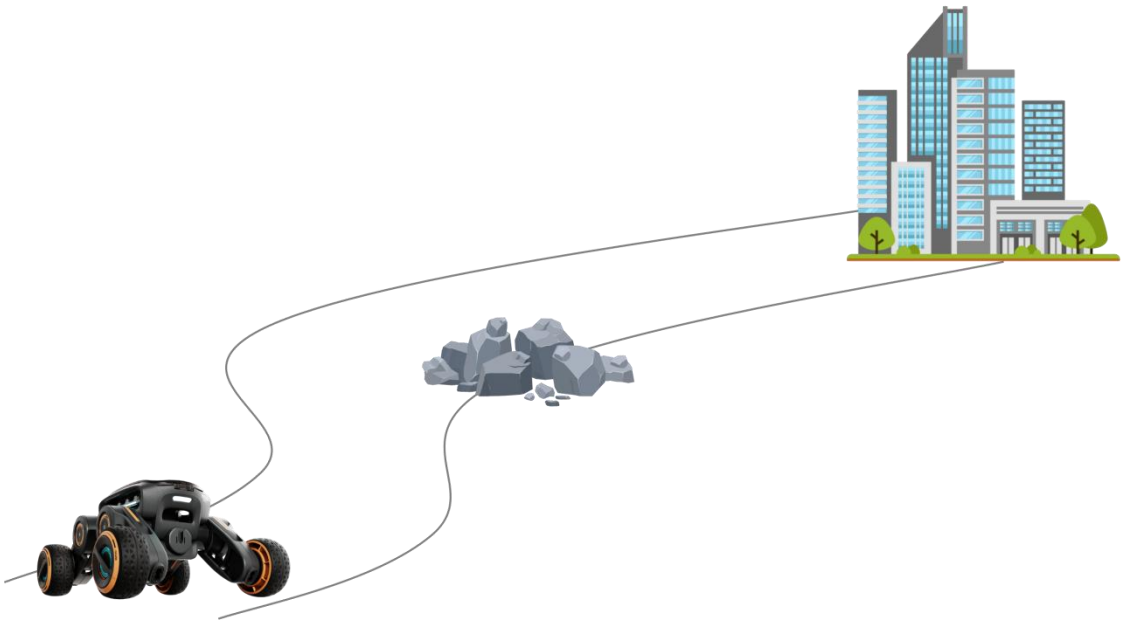


Figure 2.1 UGOT Transforming Car delivering superchips

I. Super UGOT

To ensure successful and accurate delivery of superchips to their destination, we remotely control the UGOT Transforming Car with a Bluetooth controller.

Figure 2.2 Bluetooth controller

To enable quick and accurate delivery via the Bluetooth controller, we need to solve the following problems:

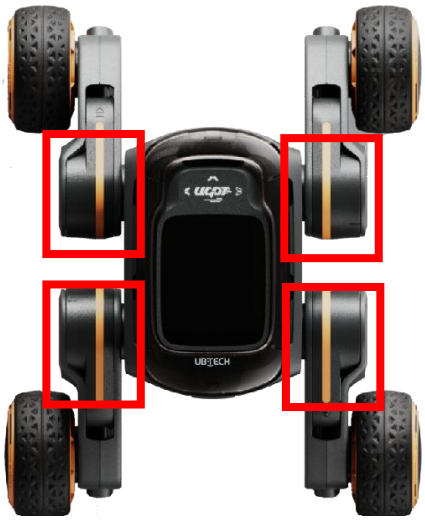
1. How do we program the motion control of the UGOT Transforming Car?
2. How can we remotely control the motions of the UGOT Transforming Car via a Bluetooth controller?

II. Happy Learning

i. Background Knowledge

1. Characteristics of UGOT Transforming Car

The UGOT Transforming Car is a four-wheel drive vehicle that can independently adjust the height of its body in order to navigate through different types of terrain. It can pass through passages with height restrictions, climb over roadblocks, climb steps, or traverse irregular terrain with height differences. It has off-road capabilities on complex roads.

	Servo: 1. Lower or raise the body of the UGOT Transforming Car. 2. Raise or lower a wheel arm of the UGOT Transforming Car.
	Motor: 1. Enable the UGOT Transforming Car to move forwards, move backwards, turn left, and turn right at different speeds. 2. Rotate one or more wheels of the UGOT Transforming Car at a specific speed.

2. Bluetooth and Bluetooth controller

Bluetooth is a short-range wireless communication technology that allows connectivity and data transfer between devices such as mobile phones, computers, and living room stereos. Bluetooth technology is widely used in headphones, onboard systems, smart homes and more.

Advantages of Bluetooth technology:

Low power consumption: The low-power mode of Bluetooth technology

effectively prolongs the use of the device.

Ease of use: Bluetooth connection can be completed with simple pairing, without requiring a complicated setup procedure.

Wide application: Bluetooth technology is widely used in mobile phones, computers, cars, smart TVs, and other devices.

Safety and reliability: Bluetooth technology has a certain degree of security, which can encrypt and protect the transmitted data to prevent data leakage.

Disadvantages of Bluetooth technology:

Limited transmission distance: As Bluetooth technology uses short-range communication, the transmission distance is relatively limited, generally within 10 metres.

Slower transfer speed: Compared to other wireless technologies, Bluetooth technology has slower transfer speeds, which may result in long transfer times for large files.

A Bluetooth controller is a controller that uses Bluetooth technology for wireless connectivity. When the Bluetooth controller and the UGOT Transforming Car are connected, the controller acts as a transmitter to send the control signal, and UGOT acts as a receiver to receive the signal.

Figure 2.3 Bluetooth controller introduction

Steps for connecting Bluetooth controller to UGOT

Step 1: Switch on the Bluetooth function of UGOT in the settings. 	Step 2: Press and hold the on/off button of the Bluetooth controller until the indicator light illuminates, indicating that it is successfully switched on. 
Step 3: Press and hold the connection button of Bluetooth; the rapidly flashing indicator light indicates that pairing is in progress. 	Step 4: Bring the Bluetooth controller close to the UGOT's main controller until the indicator light stays illuminated, indicating that it is successfully paired. 

ii. Coding Knowledge

1. Transforming Car moves forwards/backwards

We can use the 'transform_move_speed(direction, speed)' method to control the UGOT Transforming Car to move forwards or backwards at a specific speed. The details are as follows:

Method: transform_move_speed(direction, speed)

Function: controls the UGOT Transforming Car to move forwards or backwards at a specified speed

Parameters:

direction: direction; the number 0 indicates forward, and the number 1 indicates backward

speed: speed of motion; range: 5–80 cm/s

2. Transforming Car turns left/right

We can use the 'transform_turn_speed(turn, speed)' method to control the UGOT Transforming Car to turn left or right at a specific speed. The details are as follows:

Method: transform_turn_speed(turn, speed)

Function: controls the UGOT Transforming Car to turn left or right at a specified speed

Parameters:

turn: direction; the number 2 indicates a left turn, and the number 3 indicates a right turn

speed: speed of rotation; range: 5–280 degrees/s

3. Transforming Car stops

We can use the 'transform_stop()' method to control the UGOT Transforming Car to stop moving. The details are as follows:

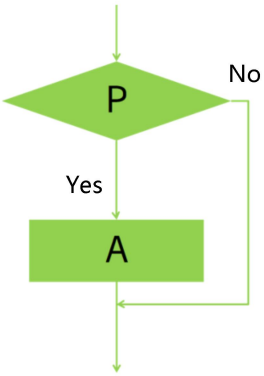
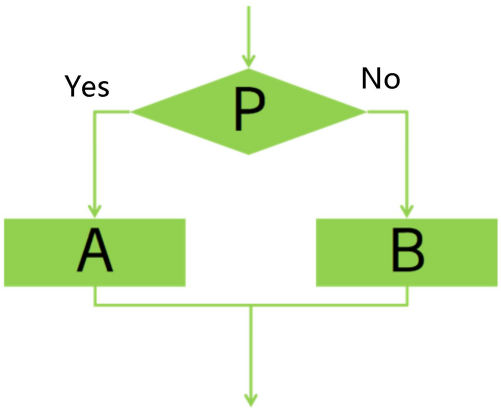
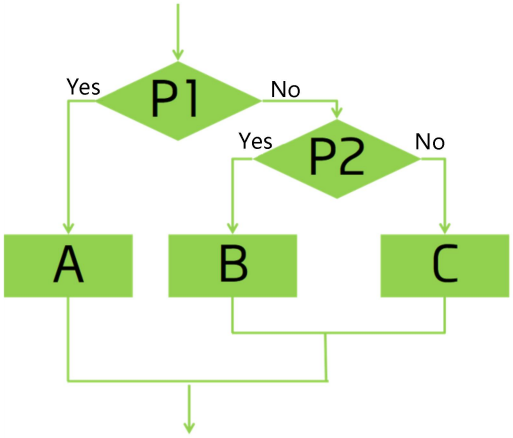
Method: transform_stop()

Function: controls the UGOT Transforming Car to stop moving

4. Branch structure

In the Python programming language, a branch structure controls the flow of a program. Conditional judgement statements can be used to execute different code blocks based on specific conditions.

Here are three common branch structures:

Branch structure	Writing format	Flowchart	Introduction
Single-branch conditional judgement	**'if' expression P:** **Code block A**		If condition P is met, execute code block A, otherwise skip code block A and go down.
Double-branch conditional judgement	**'if' expression P:** **Code block A** **else:** **Code block B**		If condition P is met, code block A is executed and if not, code block B is executed.
Multiple-branch conditional judgement	**'if' expression P1:** **Code block A** **'elif' expression P2:** **Code block B** **else:** **Code block C**		Judgement is made from top to bottom; once one of the conditions is executed, the others will not be executed.

Here are some common notes when using branch structures in Python:

(1) Indentation: In a branch structure, it is necessary to ensure that the code blocks under the conditional statement are correctly indented. Indentation is usually done using four spaces or a Tab space. Incorrect indentation can lead to syntax errors or errors in code logic.

(2) Conditional expressions: In a branch structure, it is necessary to ensure that the result of a conditional expression is a Boolean value (True or False).

(3) Readability and simplicity: Good code should feature good readability and simplicity. When writing branch structures, try to use clear and concise conditions and code blocks and avoid overly complex logic and lengthy code.

5. Get button information

We can use the 'get_joypad_pressing_buttons()' method to get information about the button pressed on the Bluetooth controller. The details are as follows:

> Method: get_joypad_pressing_buttons()
>
> Function: obtains the information about the button pressed on the Bluetooth controller. The return value is a list, for example: [], ['X']

Concept of lists: In Python, a list is an ordered set of elements separated by commas and identified by [].

Examples of common lists: empty list [], integer list [1, 2, 3], and string list ['a', 'b']

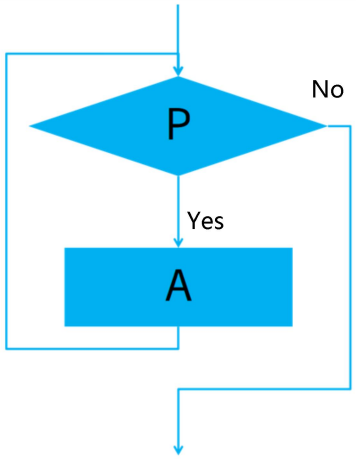
6. Comparison operators

Comparison operators in Python are used to compare two values for equality or size. When comparing two values, the result of the comparison operator is a Boolean value (True or False). The common comparison operators used in Python are listed below:

Symbol	Name	Introduction
==	Equal	Tests if two values are equal
! =	Not equal	Tests if two values are not equal
>	Greater than	Tests if the value on the left is greater than the value on the right
> =	Greater than or equal to	Tests if the value on the left is greater than or equal to the value on the right
<	Less than	Tests if the value on the left is less than the value on the right
< =	Less than or equal to	Tests if the value on the left is less than or equal to the value on the right

'==' in the lesson is used to test if two values are equal. For example, a==b indicates to judge whether the value of a is equal to the value of b, while a=b indicates assigning the value of b to a. In Python, '=' is an assignment operator.

7. 'while' loop

Loop structure	Writing format	Flowchart
'while' loop	'while' expression P: Code block A	

The 'while' loop repeats a segment of the code block when certain conditions are met. Take the flowchart as an example. During execution, a judgement is made as to whether condition P is true. If the condition is true, code block A is executed, and then re-judge condition P until it is false, which ends the loop.

In order to prevent infinite loops, you need to make sure that the loop condition can be changed to False at some point, or you can manually terminate the loop with a break statement inside the loop.

III. Creative Factory

With the knowledge that we have gained, let's program to remotely control the motions of the UGOT Transforming Car with a Bluetooth controller.

1. Task 1

Task overview: Program to control the motions of the UGOT Transforming Car.

```
#Initialisation
from ugot import ugot
import time
got=ugot.UGOT()
got.initialize("192.168.1.115")

got.transform_move_speed(0,20) #The UGOT Transforming Car moves forwards at a speed of 20 cm/s
time.sleep(5)                  #Wait 5 seconds
got.transform_move_speed(1,20) #The UGOT Transforming Car moves backwards at a speed of 20 cm/s
time.sleep(5)
got.transform_turn_speed(2,40) #The UGOT Transforming Car turns left at a speed of 40 degrees/s
time.sleep(5)
got.transform_turn_speed(3,40) #The UGOT Transforming Car turns right at a speed of 40 degrees/s
time.sleep(5)
got.transform_stop()           #The UGOT Transforming Car stops
```

Figure 2.4 Demo program of Task 1

2. Task 2

Task overview: Program to implement motion control of UGOT Transforming Car with a Bluetooth controller.

```python
#Initialisation
from ugot import ugot
got=ugot.UGOT()
got.initialize("192.168.1.115")

while True:  #Infinite loop
  a=got.get_joypad_pressing_buttons()


  if a==['X']:
    got.transform_move_speed(0,20)
  if a==['B']:
    got.transform_move_speed(1,20)
  if a==['Y']:
    got.transform_turn_speed(2,40)
  if a==['A']:
    got.transform_turn_speed(3,40)
  if a==['R2']:
    got.transform_stop()
    break  #Break loop
```

Figure 2.5 Demo program for Task 2

IV. Knowledge Summary

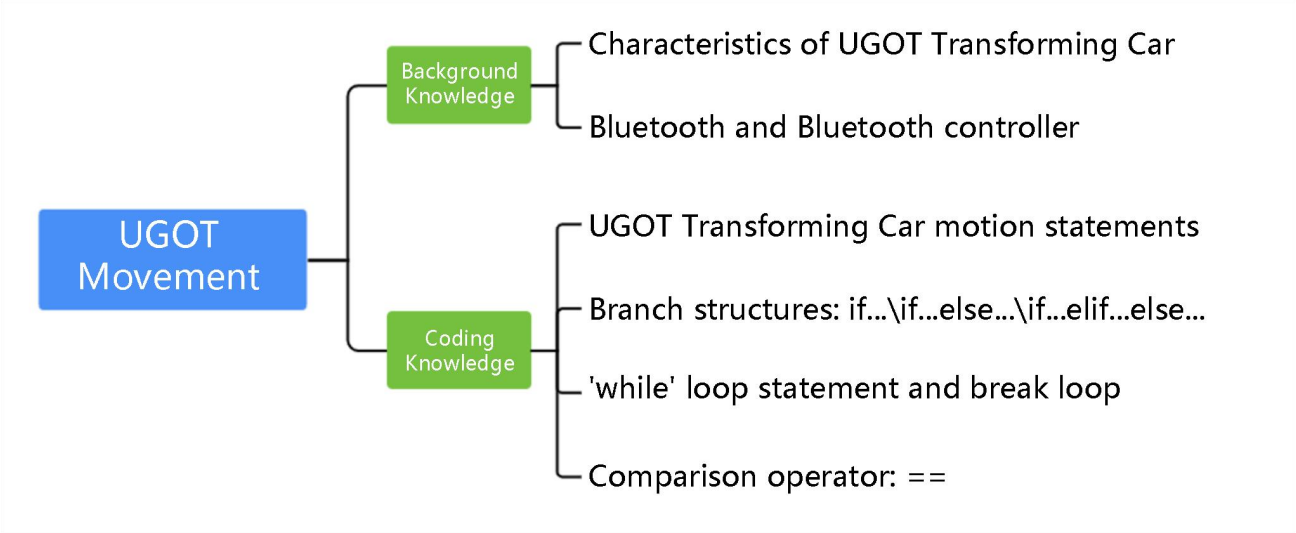


Figure 2.6 Knowledge summary

V. Extension

i. Practice & Innovation

Complete the following extension task:

Program to lower the body by pressing button L1 and raise the body by pressing button L2.

Method: transform_set_chassis_height(height)

Function: set the height of the UGOT Transforming Car

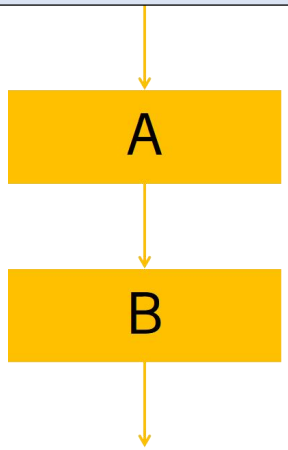
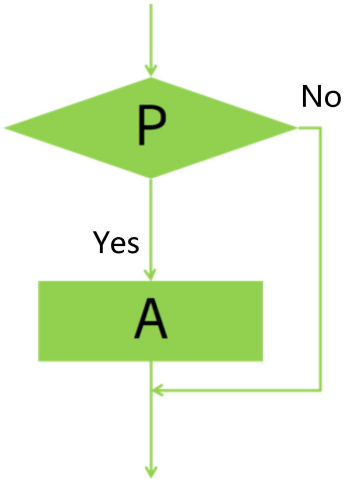
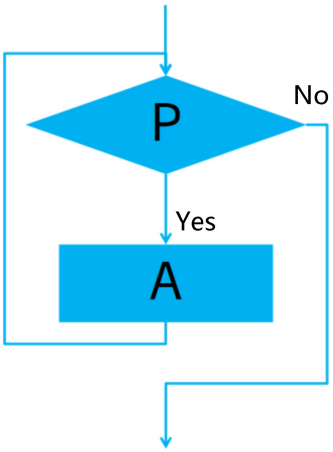
Parameters:

height: height, with a range of 1 to 7 cm

(Default height of the UGOT Transforming Car is 4 cm)

ii. Additional Knowledge

Three major structures of programming

Name	Flowchart	Introduction
Sequence structure		The program executes in a top-down sequence, with each statement being executed only once.
Branch structure		The execution path is chosen based on certain conditions, rather than the order in which the statements appear.
Loop structure		When certain conditions are met, specific statements are executed repeatedly until the condition is not met and the loop ends.