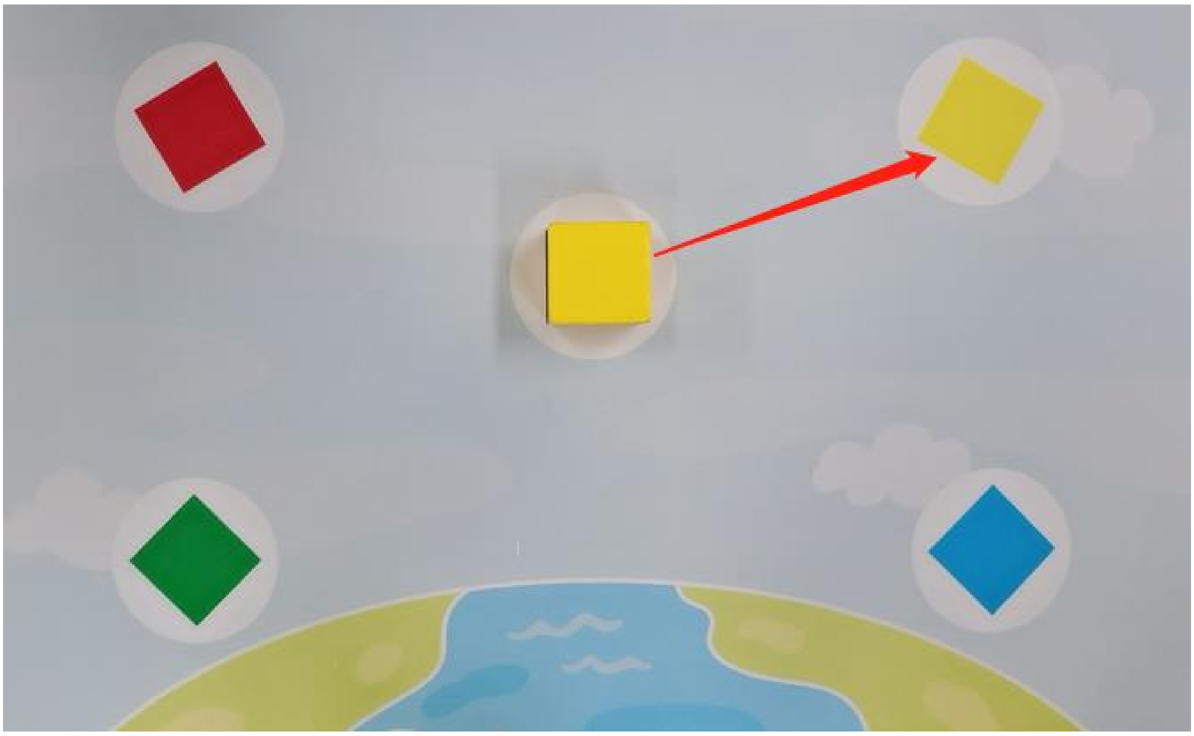
6.9 DOFBOT grab block

1.Game introduction

The purpose of this experiment is to move the blocks from the gray area in the middle to the surrounding square areas of different colors. First put the yellow block into the gray area, and then run the code unit to the sixth unit in sequence (grab a building block from the gray building block and place it on the yellow building block),At this time, the robot arm will automatically grab the block placed in the gray area, then place it in the yellow area, and then return to the ready position. Before running the seventh code unit, you need to place the red block in the gray area, and then run the seventh unit (grab a building block from the gray building block and place it on the red building block),In this way, red squares will also be captured to the red area, and other squares will be operated in the same way.



2.Code content

Code path：/home/jetson/Dofbot/3.ctrl_Arm/9.clamp_block.ipynb

The following code content needs to be executed according to the actual step, and cannot be run all at once. Before picking up the building blocks, you need to place the building blocks on the gray building blocks in the middle, and only one block can be placed at a time.

```
#!/usr/bin/env python3
#coding=utf-8
import time
from Arm_Lib import Arm_Device
# Create robot arm object
Arm = Arm_Device()
time.sleep(.1)
```

```
# Define the function of clamping building blocks,enable=1: clamp, =0: release
def arm_clamp_block(enable):
 if enable == 0:
 Arm.Arm_serial_servo_write(6, 60, 400)
 else:
 Arm.Arm_serial_servo_write(6, 130, 400)
 time.sleep(.5)
# Define the mobile robot arm function and control the movement of servos No. 1-5
at the same time, p=[S1,S2,S3,S4,S5]
def arm_move(p, s_time = 500):
 for i in range(5):
 id = i + 1
 if id == 5:
 time.sleep(.1)
 Arm.Arm_serial_servo_write(id, p[i], int(s_time*1.2))
 else :
 Arm.Arm_serial_servo_write(id, p[i], s_time)
 time.sleep(.01)
 time.sleep(s_time/1000)
# Robotic arm moves up
def arm_move_up():
 Arm.Arm_serial_servo_write(2, 90, 1500)
 Arm.Arm_serial_servo_write(3, 90, 1500)
 Arm.Arm_serial_servo_write(4, 90, 1500)
 time.sleep(.1)
```

```
# Define variable parameters at different locations
p_mould = [90, 130, 0, 0, 90]
p_top = [90, 80, 50, 50, 270]
p_Brown = [90, 53, 33, 36, 270]
p_Yellow = [65, 22, 64, 56, 270]
p_Red = [117, 19, 66, 56, 270]
p_Green = [136, 66, 20, 29, 270]
p_Blue = [44, 66, 20, 28, 270]
```

```
# Let the robotic arm move to a position ready to grab
arm_clamp_block(0)
arm_move(p_mould, 1000)
time.sleep(1)
```

```
# Grab a building block from the gray building block and place it on the yellow
building block.
arm_move(p_top, 1000)
arm_move(p_Brown, 1000)
arm_clamp_block(1)
arm_move(p_top, 1000)
arm_move(p_Yellow, 1000)
arm_clamp_block(0)
arm_move(p_mould, 1000)
time.sleep(1)
```

```
# Grab a building block from the gray building block and place it on the red building block.
arm_move(p_top, 1000)
arm_move(p_Brown, 1000)
arm_clamp_block(1)
arm_move(p_top, 1000)
arm_move(p_Red, 1000)
arm_clamp_block(0)
arm_move_up()
arm_move(p_mould, 1100)
time.sleep(1)
```

```
# Grab a building block from the gray building block and place it on the green building block.
arm_move(p_top, 1000)
arm_move(p_Brown, 1000)
arm_clamp_block(1)
arm_move(p_top, 1000)
arm_move(p_Green, 1000)
arm_clamp_block(0)
arm_move_up()
arm_move(p_mould, 1100)
time.sleep(1)
```

```
# Grab a building block from the gray building block and place it on the blue building block.
arm_move(p_top, 1000)
arm_move(p_Brown, 1000)
arm_clamp_block(1)
arm_move(p_top, 1000)
arm_move(p_Blue, 1000)
arm_clamp_block(0)
arm_move_up()
arm_move(p_mould, 1100)
time.sleep(1)
```

```
del Arm # Release Arm object
```