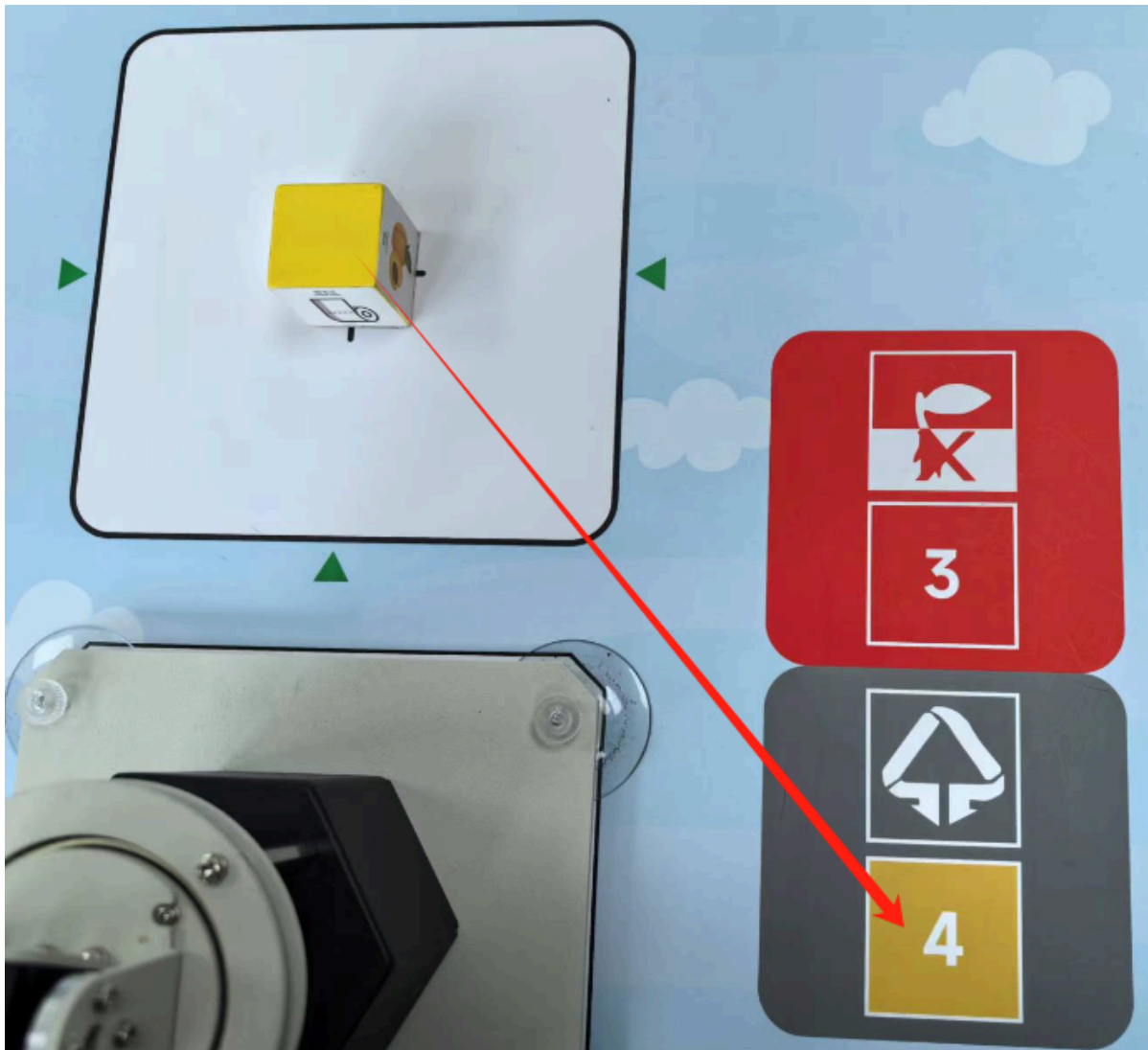


# Robotic Arm Clamping Blocks

## 1. Introduction to Playing

The purpose of this experiment is to move blocks from the middle cross area to the surrounding blocks of different colors. First, put the yellow block in the cross area, and then run the code unit to the sixth unit in sequence (grab a block from the cross block position and put it in the yellow block position). At this time, the robot arm will automatically grab the block in the cross area, then put it in the yellow area, and then return to the ready position. Before running the seventh code unit, you need to put the red block in the cross area, and then run the seventh unit (grab a block from the cross block position and put it in the red block position). In this way, the red block will also be grabbed to the red area, and the operation method for other blocks is the same.



## 2. Code content

Code path:

```
~/dofbot_pro/dofbot_ctr1/scripts/09.clamp_block.ipynb
```

The following code content needs to be executed according to each actual step, and cannot be run all at once. Before clamping the building block, you need to put the building block in the middle cross building block position, and only one building block can be placed at a time.

```
#!/usr/bin/env python3
#coding=utf-8
import time
from Arm_Lib import Arm_Device

# Create a robot object
Arm = Arm_Device()
time.sleep(.1)

from dofbot_utils.robot_controller import Robot_Controller
robot = Robot_Controller()
```

```
# Define the clamping block function, 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, 135, 400)
    time.sleep(.5)

# Define the function of moving the robot arm, and control the movement of servos
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)

# Move the robot arm 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 for different positions
p_mould = robot.P_LOOK_AT
p_top = robot.P_TOP
p_Brown = robot.P_CENTER
p_Yellow = robot.P_YELLOW
p_Red = robot.P_RED
p_Green = robot.P_GREEN
p_Blue = robot.P_BLUE
```

```
# Move the robot arm to a position ready for grasping
arm_clamp_block(0)
arm_move(p_mould, 1000)
time.sleep(1)
```

```
# Grab a block from the center of the recognition area and place it on the  
position of the yellow 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 block from the center of the recognition area and place it on the red  
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 block from the center of the recognition area and place it on the green  
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 block from the center of the recognition area and place it on the blue  
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 the Arm object
```