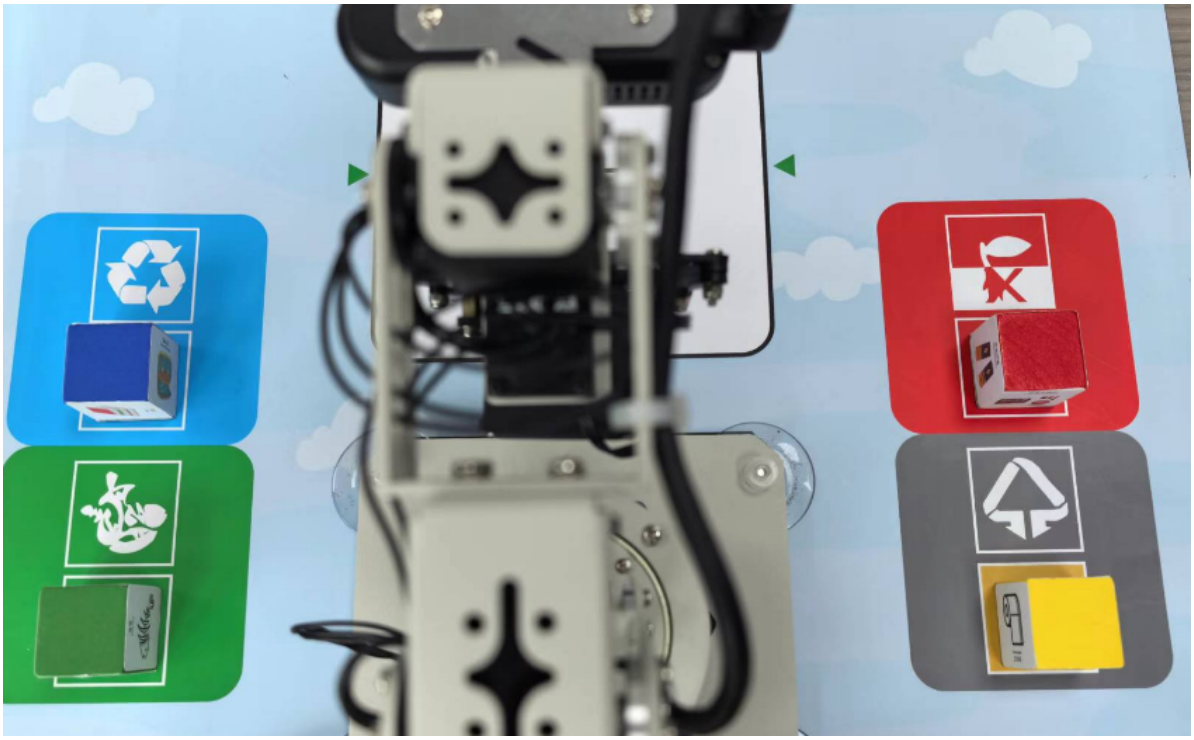
Stack Blocks

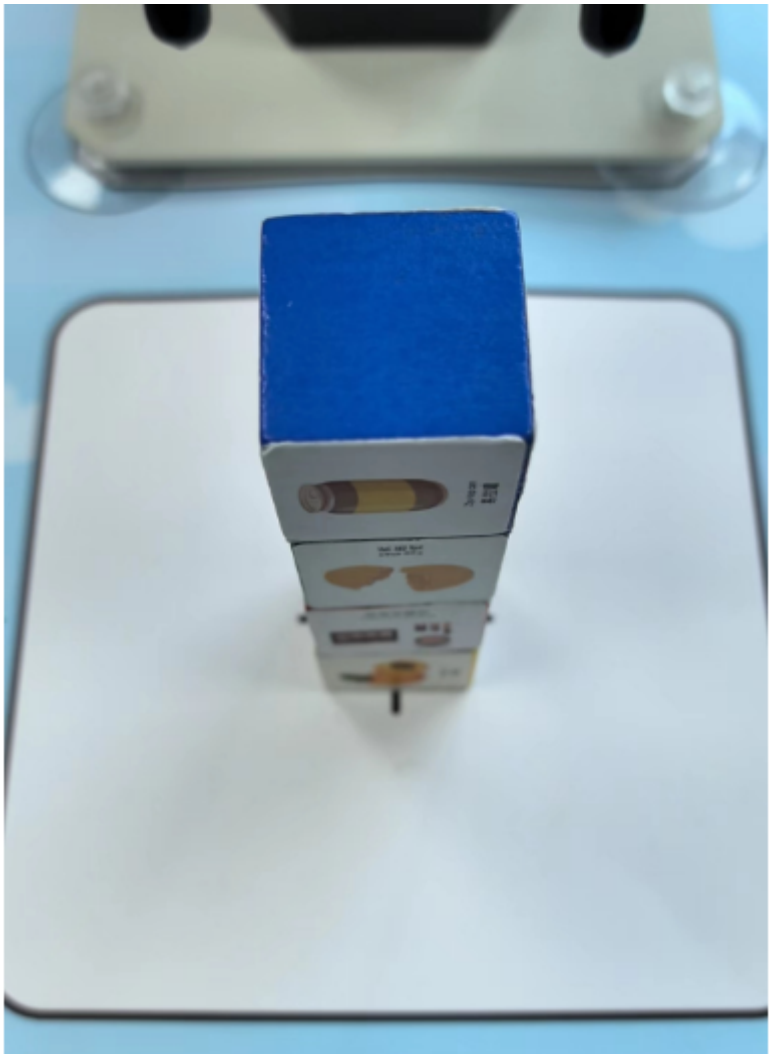
1. Introduction

This experiment is the opposite of the previous lesson 'Nature Porter'. It aims to grab blocks from different surrounding areas in the order of yellow, red, green, and blue, then stack them in the central cross area.

The block arrangement method is shown in the following figure:



After executing the code, the robotic arm will stack the blocks together. The final effect is shown in the following figure:



2. Code Content

Code path:

```
~/dofbot_pro/dofbot_ctrl/scripts/11.heap_up.ipynb
```

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

# Create robotic arm object
Arm = Arm_Device()
time.sleep(.1)
robot = Robot_Controller()
```

```
# Define block grabbing function, enable=1: grab, =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 robotic arm movement function, simultaneously controls servos 1-5
movement, 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))
        elif id == 1 :
            Arm.Arm_serial_servo_write(id, p[i], int(3*s_time/4))

        else:
            Arm.Arm_serial_servo_write(id, p[i], int(s_time))
        time.sleep(.01)
    time.sleep(s_time/1000)
```

```
# Define variable parameters for different positions
p_mould = robot.P_LOOK_AT
p_top = robot.P_TOP
p_layer_4 = robot.P_CENTER_HEAP_L4
p_layer_3 = robot.P_CENTER_HEAP_L3
p_layer_2 = robot.P_CENTER_HEAP_L2
p_layer_1 = robot.P_CENTER_HEAP_L1
p_Yellow = robot.P_YELLOW
p_Red = robot.P_RED
p_Green = robot.P_GREEN
p_Blue = robot.P_BLUE
```

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

```
# Grab the block from the yellow area and stack it at the bottom center
position.
arm_move(p_top, 1000)
arm_move(p_Yellow, 1000)
arm_clamp_block(1)
Arm.Arm_serial_servo_write(2, 90, 1000)
time.sleep(1)
arm_move(p_top, 1000)
arm_move(p_layer_1, 1000)
arm_clamp_block(0)
time.sleep(.1)
arm_move(p_mould, 1100)
```

```
# Grab the block from the red area and stack it at the second layer center
position.
arm_move(p_top, 1000)
arm_move(p_Red, 1000)
arm_clamp_block(1)
Arm.Arm_serial_servo_write(2, 90, 1000)
time.sleep(1)
arm_move(p_top, 1000)
arm_move(p_layer_2, 1000)
arm_clamp_block(0)
time.sleep(.1)
arm_move(p_mould, 1100)
```

```
# Grab the block from the green area and stack it at the third layer center
position.
arm_move(p_top, 1000)
arm_move(p_Green, 1000)
arm_clamp_block(1)
Arm.Arm_serial_servo_write(2, 90, 1000)
time.sleep(1)
arm_move(p_top, 1000)
arm_move(p_layer_3, 1000)
arm_clamp_block(0)
time.sleep(.2)
arm_move(p_mould, 1100)
```

```
# Grab the block from the blue area and stack it at the fourth layer center
position.
arm_move(p_top, 1000)
arm_move(p_Blue, 1000)
arm_clamp_block(1)
Arm.Arm_serial_servo_write(2, 90, 1000)
time.sleep(1)
arm_move(p_top, 1000)
arm_move(p_layer_4, 1000)
arm_clamp_block(0)
time.sleep(.2)
arm_move(p_mould, 1100)
```

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

Open the program file in jupyter lab and click the "Run entire notebook" button on the jupyter lab toolbar to see the robotic arm's block stacking function effect.