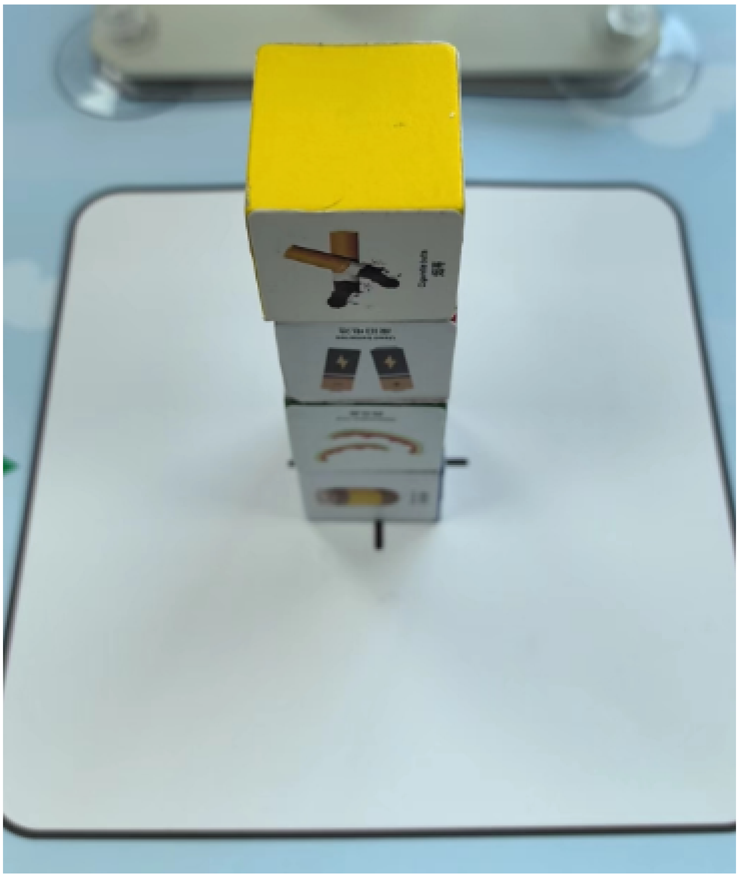
Porter

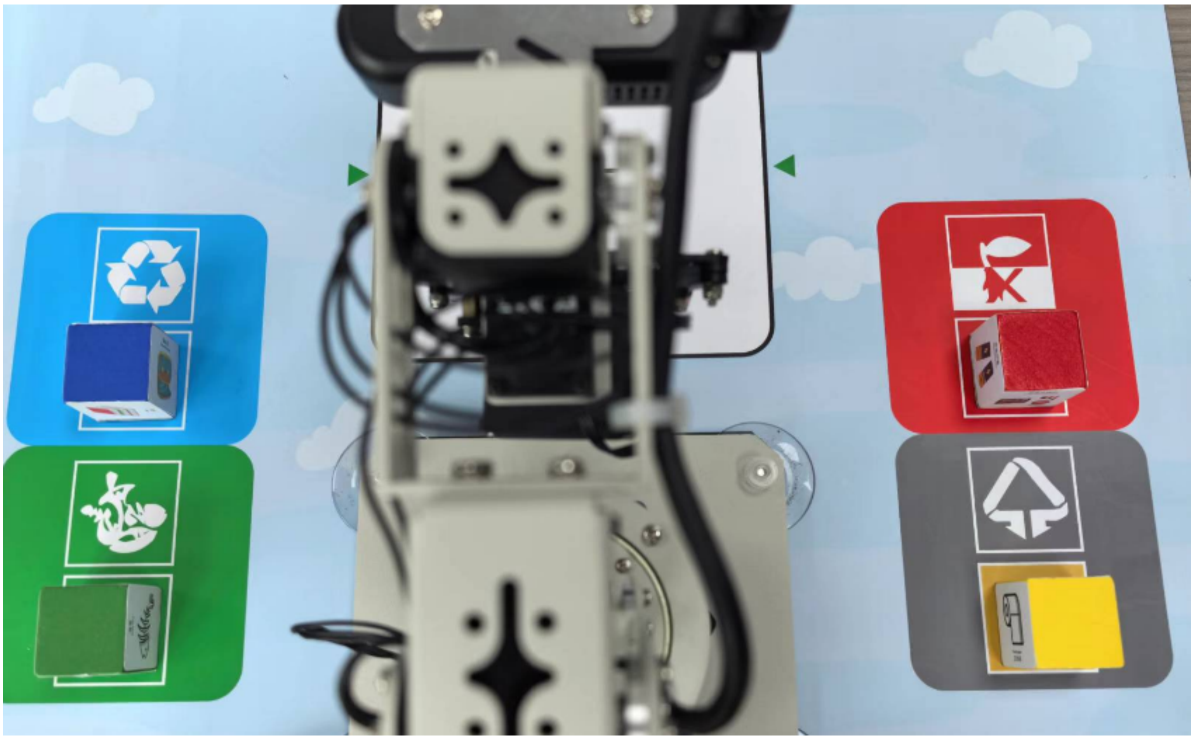
1. Introduction

This experiment aims to stack four different colored blocks in the order of blue, green, red, and yellow from bottom to top, then place them on the central cross block. After running the code, the robotic arm will execute in sequence: grab the fourth layer block and place it in the yellow area, grab the third layer block and place it in the red area, grab the second layer block and place it in the green area, and grab the bottom layer block and place it in the blue area.

The block arrangement method is shown in the following figure:



After executing the code, the robotic arm will move the blocks to corresponding positions. The final effect is shown in the following figure:



2. Code Content

Code path:

```
~/dofbot_pro/dofbot_ctrl/scripts/10.move_block.ipynb
```

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

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

from dofbot_utils.robot_controller import Robot_Controller
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))
        else :
            Arm.Arm_serial_servo_write(id, p[i], s_time)
        time.sleep(.01)
    time.sleep(s_time/1000)

# Robotic arm moves upward
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

p_layer_4 = robot.P_CENTER_4
p_layer_3 = robot.P_CENTER_3
p_layer_2 = robot.P_CENTER_2
p_layer_1 = robot.P_CENTER
```

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

```
# Move the fourth layer block to the yellow area
arm_move(p_top, 1000)
arm_move(p_layer_4, 1000)
arm_clamp_block(1)
arm_move(p_top, 1000)
arm_move(p_Yellow, 1000)
arm_clamp_block(0)
time.sleep(.1)
arm_move_up()
arm_move(p_mould, 1100)
```

```
# Move the third layer block to the red area
arm_move(p_top, 1000)
arm_move(p_layer_3, 1000)
arm_clamp_block(1)
arm_move(p_top, 1000)
arm_move(p_Red, 1000)
arm_clamp_block(0)
time.sleep(.1)
arm_move_up()
arm_move(p_mould, 1100)
```

```
# Move the second layer block to the green area
arm_move(p_top, 1000)
arm_move(p_layer_2, 1000)
arm_clamp_block(1)
arm_move(p_top, 1000)
arm_move(p_Green, 1000)
arm_clamp_block(0)
time.sleep(.1)
arm_move_up()
arm_move(p_mould, 1100)
```

```
# Move the first layer block to the blue area
arm_move(p_top, 1000)
arm_move(p_layer_1, 1000)
arm_clamp_block(1)
arm_move(p_top, 1000)
arm_move(p_Blue, 1000)
arm_clamp_block(0)
time.sleep(.1)
arm_move_up()
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 nature porter function effect.