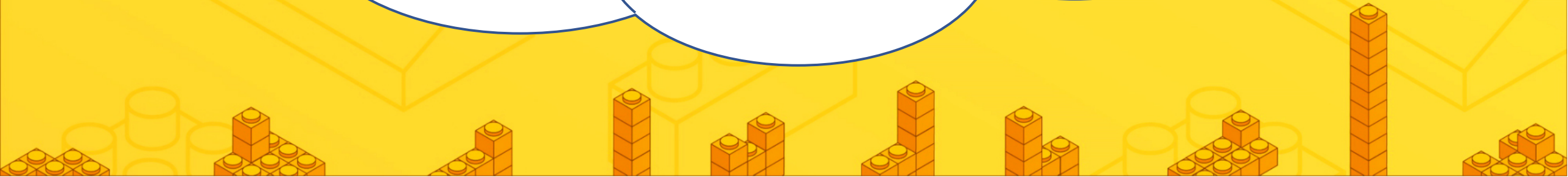




Information Transmission

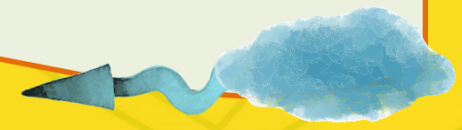


Target

- **Study Task 1: Information Transmission Rules**
- **Combine the previously learned concepts, including color detection, movement, and the use of the robotic arm to retrieve task items, to complete the content of Task 1.**



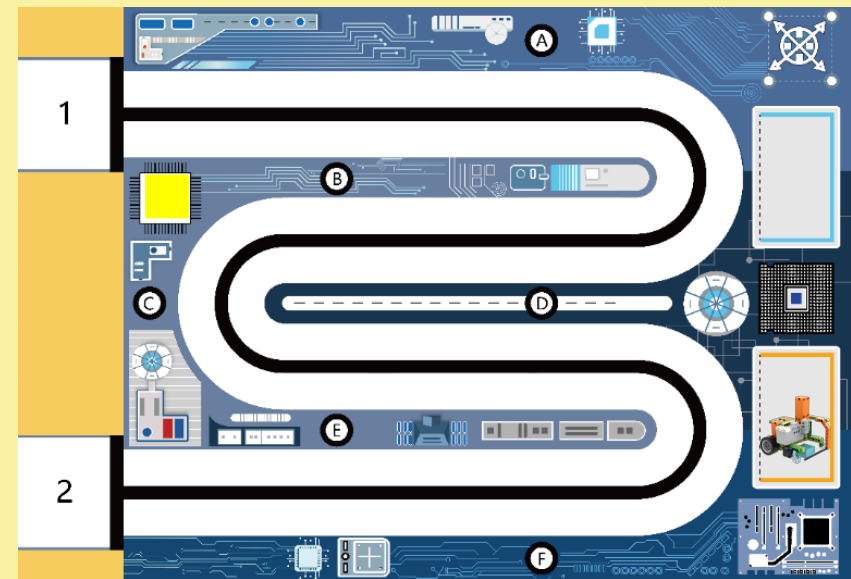
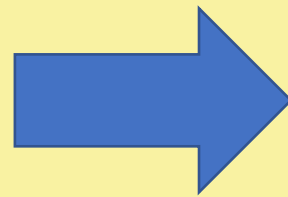
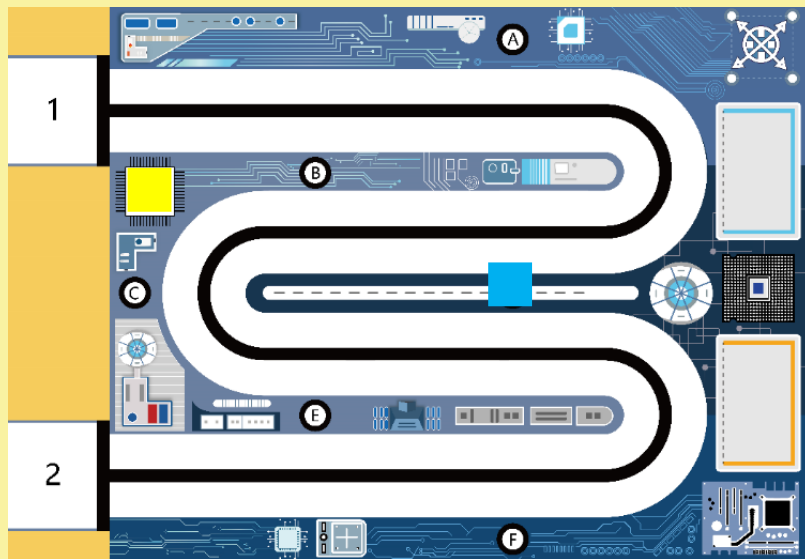
01 Task





Task

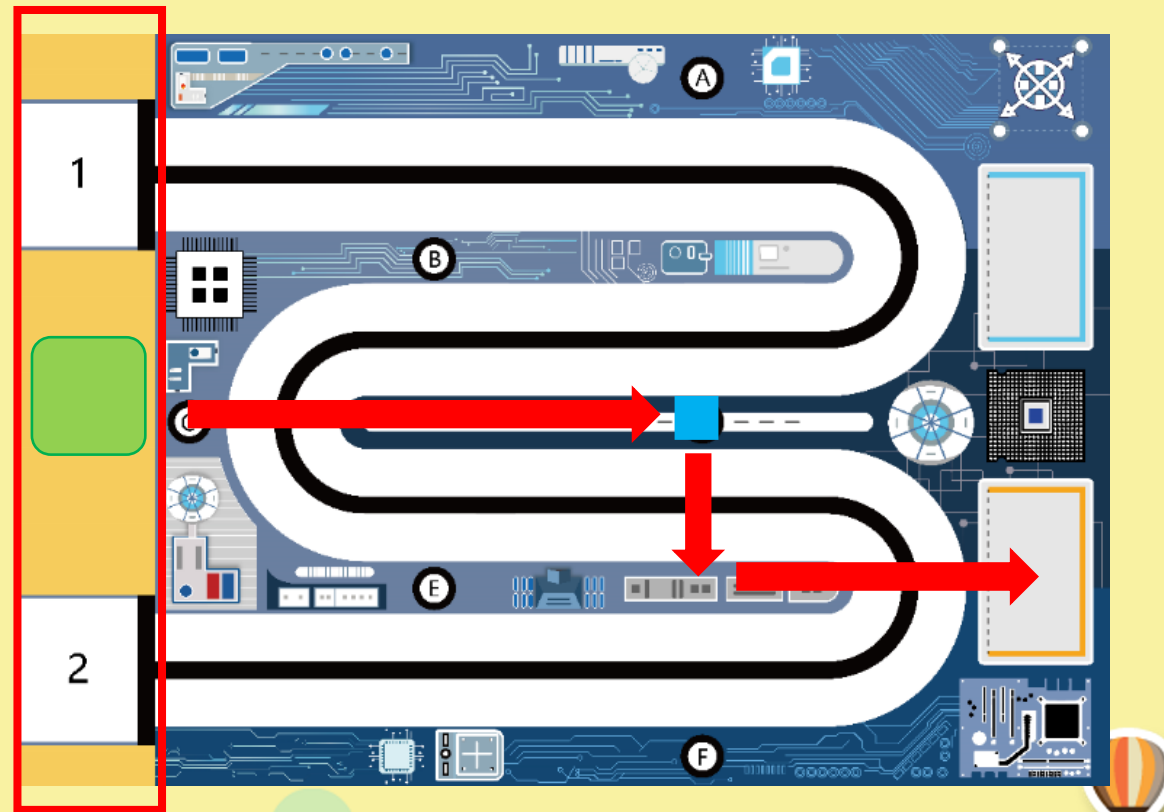
- The device starts from **any position** in the **preparation area** and needs to deliver a specified block to the designated location as quickly as possible.
- Before the competition begins, a position from A, B, C, E, or F will be randomly selected as the location to place the block. After each round of registration, the information will be extracted by color to indicate which of the yellow or green relay stations is the destination.
- For example, if position D is selected and the information color is a yellow block, then the destination is the yellow relay station.



Task Analysis

- The device starts from any position in the preparation area and needs to deliver a specified block to the designated location as quickly as possible.

- Using position D as an example, the robot can start from the green position shown in the diagram and follow the red route.



Task Analysis

- If the block representing the destination is moved to the left side of the robot, then move the color sensor to the left side as well.



Task

Referrable Program

Do you remember the decision-making program from the last lesson? Let's implement it using a different approach.



```
when clicked
  set all traffic lights to closed
  set 1# ext servo's origin
  set 2# ext servo's origin
  wait 0.3 seconds
  set all traffic lights to closed
  repeat until 1# color sensor's color code < 50
    set 1# ext servo to keep running at 20 (-100-100)% speed on anticlockwise
    set 2# ext servo to keep running at 20 (-100-100)% speed on clockwise
  stop all ext servo(s)
  if 1# color sensor's color code = 4 then
    set all lights color to blue
  else
    set all lights color to yellow
```

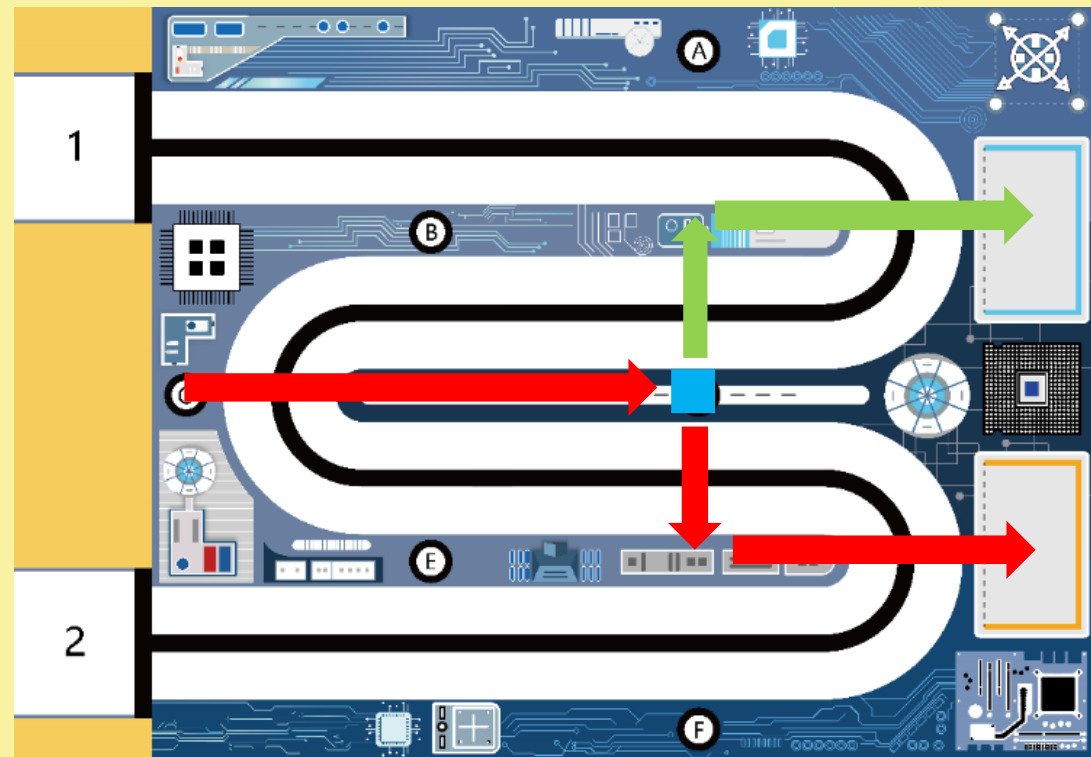
The image shows a Scratch script for a traffic light program. It starts with a 'when clicked' event block. The first three blocks are 'set all traffic lights to closed', 'set 1# ext servo's origin', and 'set 2# ext servo's origin'. This is followed by a 'wait 0.3 seconds' block. The next block is 'set all traffic lights to closed'. Then, a 'repeat until' loop is used with the condition '1# color sensor's color code < 50'. Inside the loop, there are two 'set ext servo to keep running at 20 (-100-100)% speed' blocks: one for '1#' on 'anticlockwise' and one for '2#' on 'clockwise'. After the loop, there is a 'stop all ext servo(s)' block. Finally, an 'if' block checks '1# color sensor's color code = 4'. If true, it sets 'all lights color to blue'. If false, it sets 'all lights color to yellow'.



Task Segmentation

Task 2: Complete the Actions for Detecting Yellow and Blue Areas

- Don't rush. Take it step by step and debug carefully.





Task

Referrable Program

```
define arm
  set 1# ext servo's origin
  set 2# ext servo's origin
  set 3# ext servo's origin
  wait 0.3 seconds
  set 1# ext servo to keep running at 20 (-100~100)% speed on anticlockwise
  set 2# ext servo to keep running at 20 (-100~100)% speed on clockwise
  wait until built-in line tracker's 1 value < 15
  stop 1# ext servo(s)
  stop 2# ext servo(s)
  set 3# ext servo to rotate relative angle 90 degrees at 30 (0~100)% power
  wait until is 3# ext servo done
```

To handle movement, it's advisable to use "My Blocks". Here's how to use the movement module with the yellow endpoint as an example:

```
define move a b
  set 1# ext servo's origin
  set 2# ext servo's origin
  wait 0.3 seconds
  set 1# ext servo to rotate relative angle a degrees at 30 (0~100)% speed
  set 2# ext servo to rotate relative angle b degrees at 30 (0~100)% speed
  wait until is 1# ext servo done
  wait until is 2# ext servo done
```

```
when clicked
  arm
  move 215 215
  move -500 500
  move -215 -215
  move -1000 1000
```





Task

Referrable Program

After completing Task 1, simply add the blue and yellow actions following the decision-making step, then choose to run the sequence.

```
if [1# color sensor's color code] = 4 then
  arm
  move -215 -215
  move -500 500
  move 215 215
  move -1000 1000
else
  arm
  move 215 215
  move -500 500
  move -215 -215
  move -1000 1000
```

