shape robotics
SMART LEARNING

CODING LIBRARY

100 Coding Sequences

for Fun and Meaningful Lessons

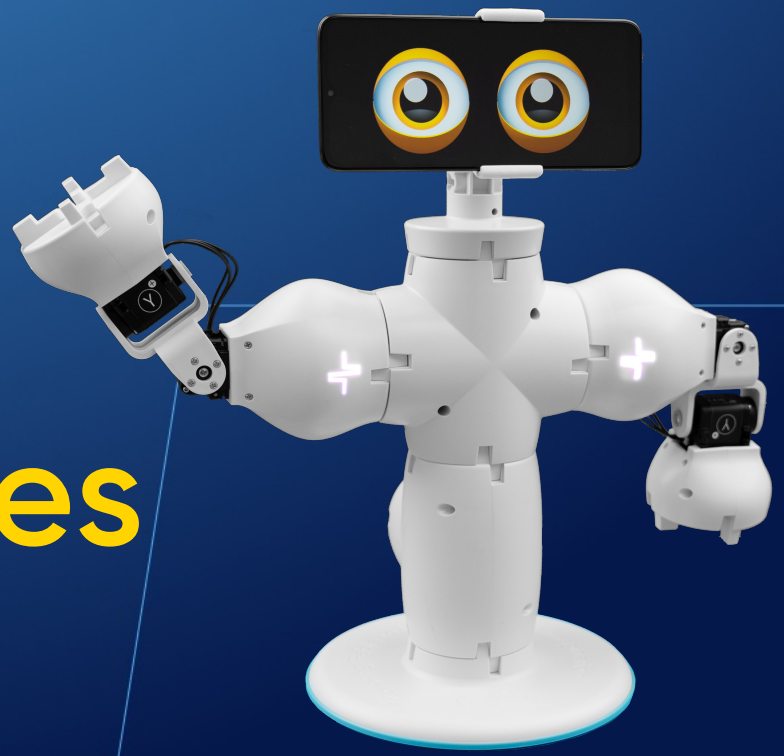


Table of Contents

shape robotics
SMART LEARNING

Keep in Mind

shape robotics
SMART LEARNING

Before programming

1. The robots are charged and functional, which can be verified directly in the application.

2. The hub is connected to the device you will be programming from. When connected, it will light up with one of six colours and the code will be visible in the application.

3. The Fable Blockly app is installed and up to date on the programming device. You can download the latest version **here**.

1. The robots have updated firmware. If needed, you can update the firmware from the app.
   Access the following links for: **Hub**, **Spin**, **Joint**.
5. The Fable Face app is installed and running on your phone.
   You can download it **here**.

5. Ensure that your internet connection is working.

6. Make sure that Bluetooth is active and working on the device you will be programming from.

While running the programs

1. The robots and the Hub should have the same color displayed.

2. Use the appropriate codes for the robots in the application.

3. Be mindful of Torque overload messages!
   These messages indicate when a motor is overloaded and requires adjustments in the program, the subassembly it is located in, or the operating environment.

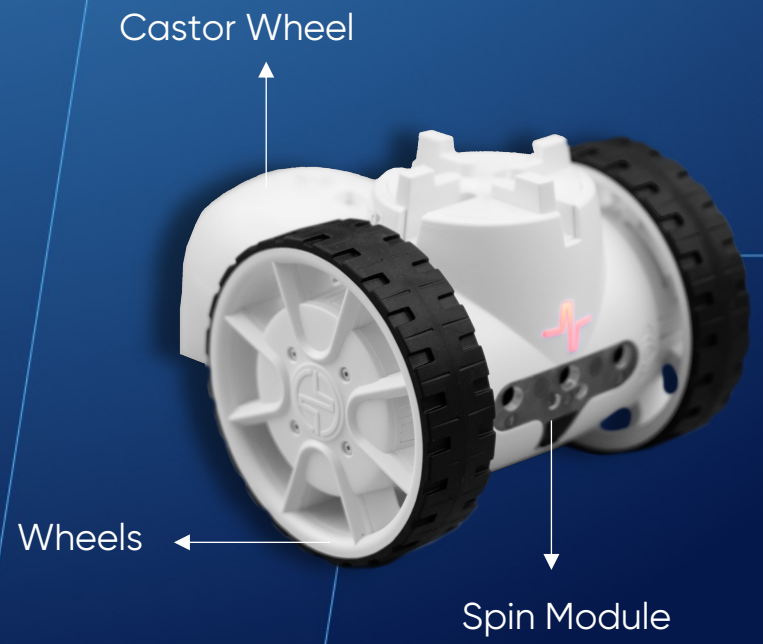
4. A Joint angle of zero degrees denotes a vertical position.

shape robotics
SMART LEARNING

Code 1:
Spin's Directions

- This program enables the robot to move forward, backward, and turn right or left.

WATCH VIDEO



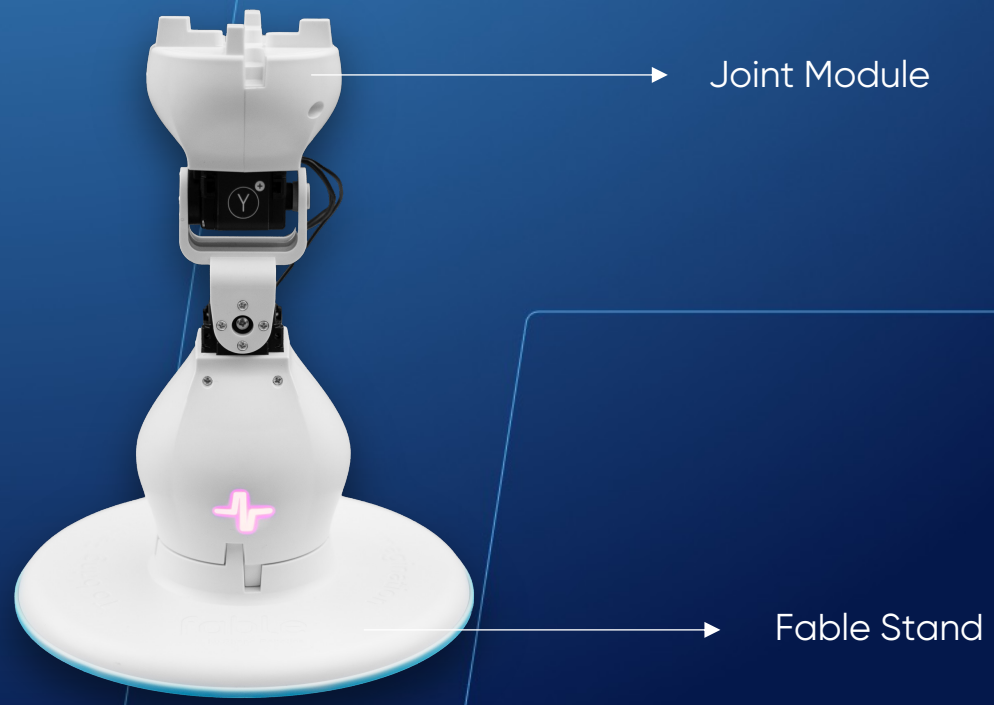
shape robotics
SMART LEARNING

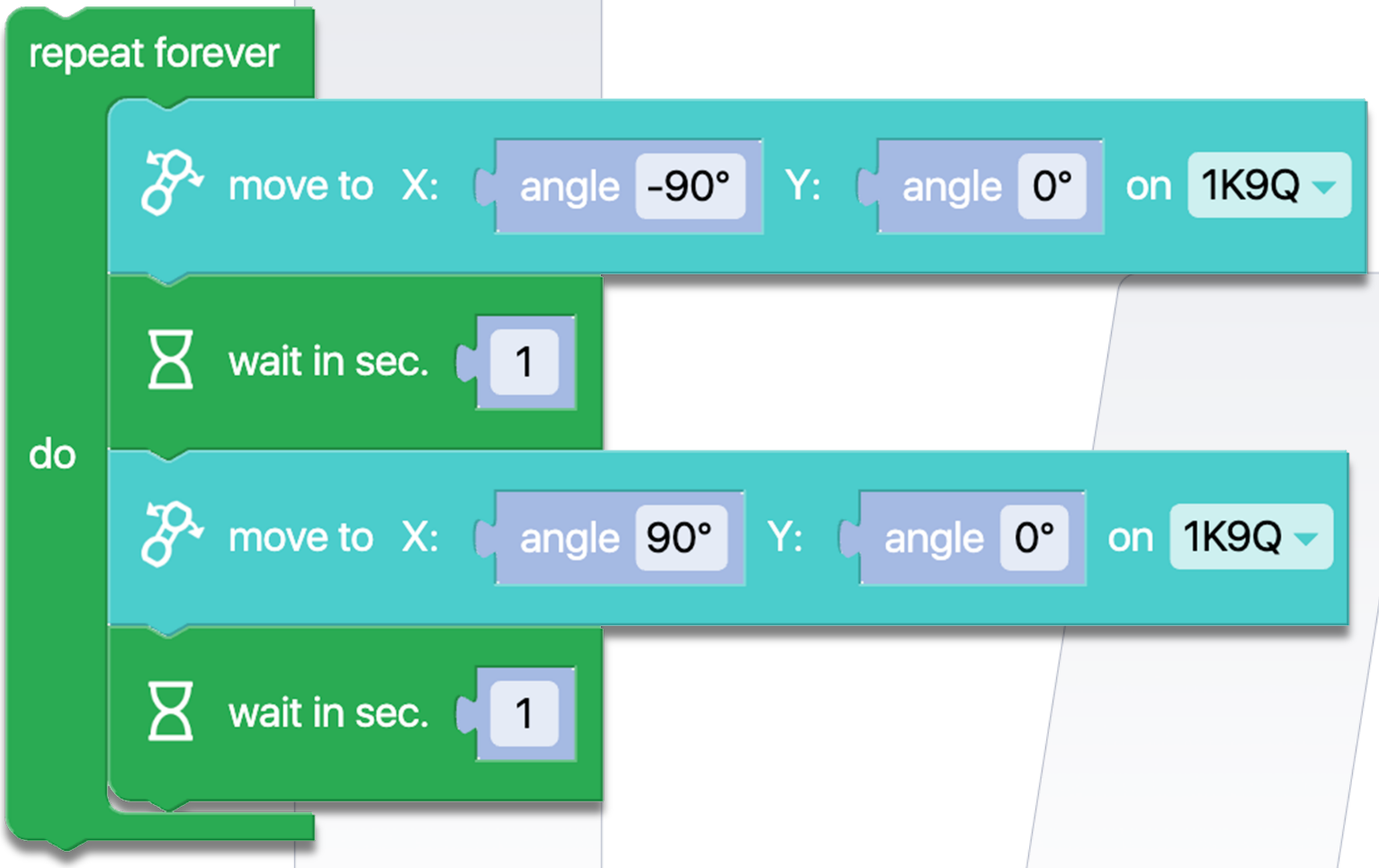
shape robotics
SMART LEARNING

Code 2:
Joint's Directions

- The Joint module performs a repetitive motion ranging from -90 degrees to 90 degrees with the assistance of the servo X motor, while the Y servo motor remains stationary. By incorporating a cutout and a customized cardboard hand, the robot can bid farewell by waving its hand.

WATCH VIDEO



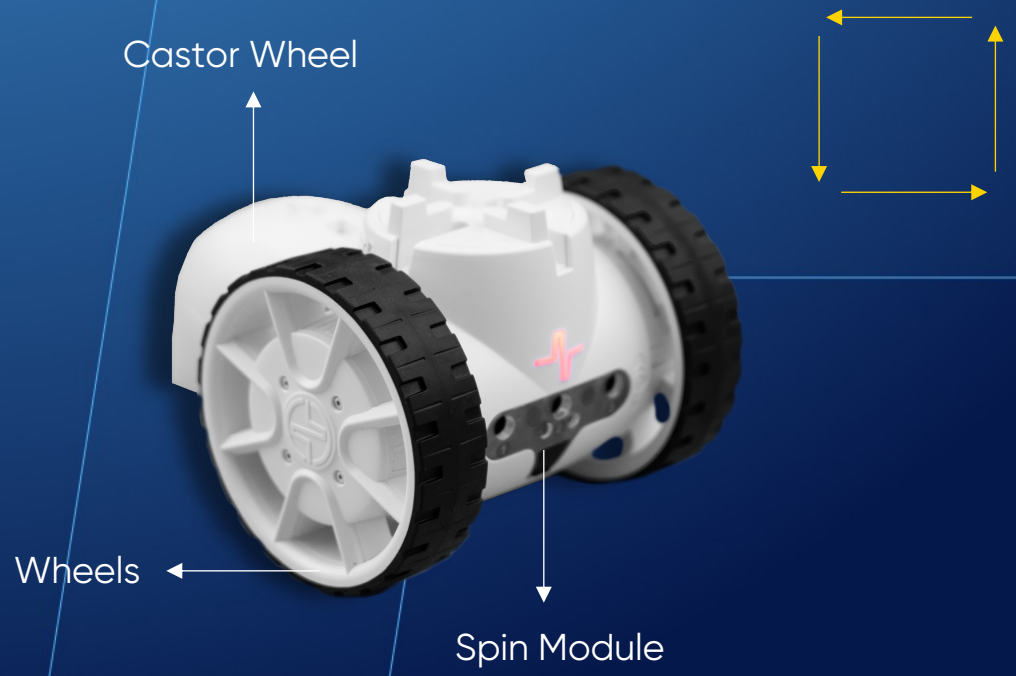

repeat forever
move to X: angle -90° Y: angle 0° on 1K9Q
wait in sec. 1
do
move to X: angle 90° Y: angle 0° on 1K9Q
wait in sec. 1

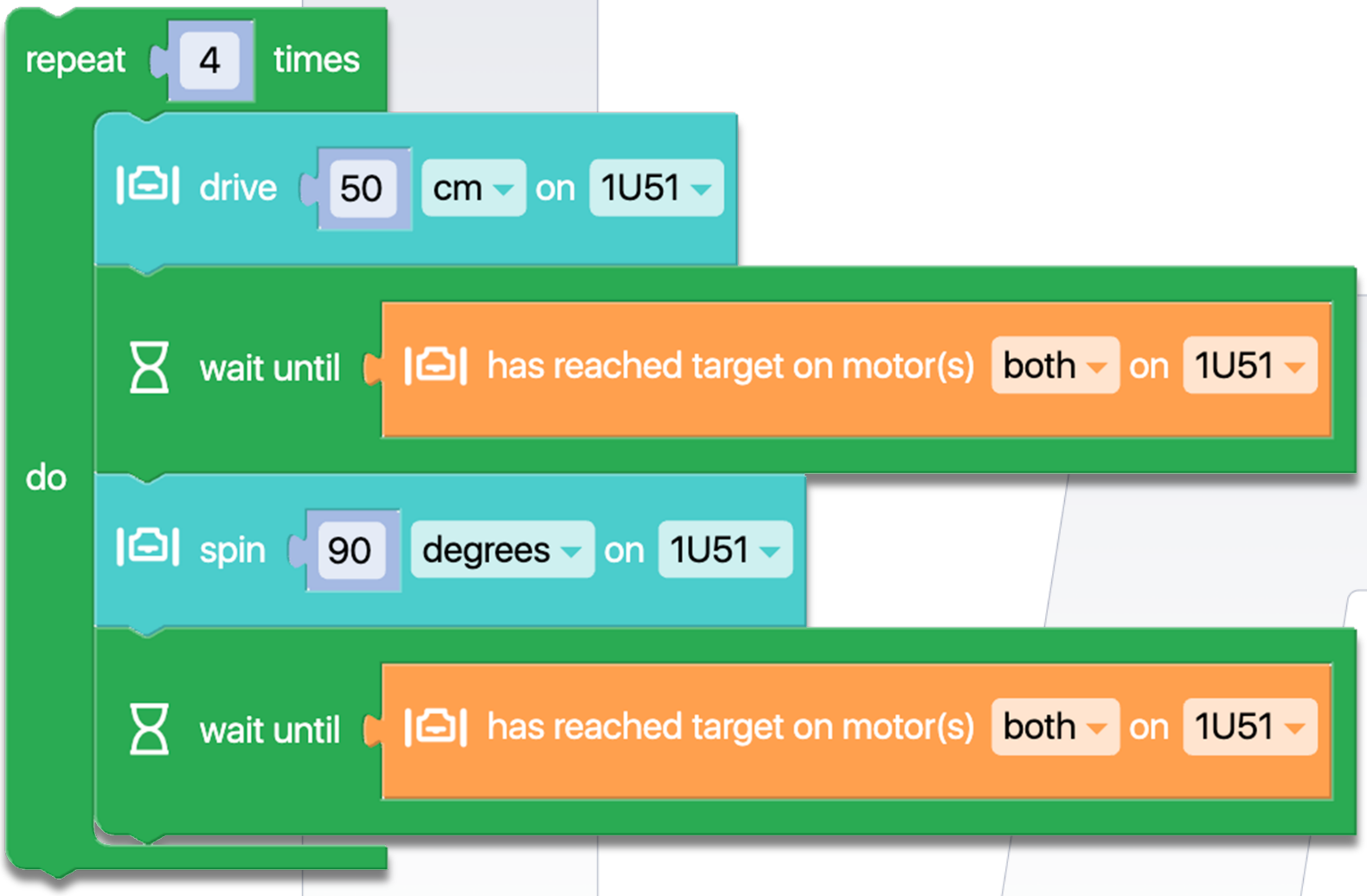
shape robotics
SMART LEARNING

Code 3:
Driving in a Square

- The Spin module moves along the perimeter of a square, with each side measuring 50 centimeters. The orange block ensures that each preceding command has already been executed.

WATCH VIDEO

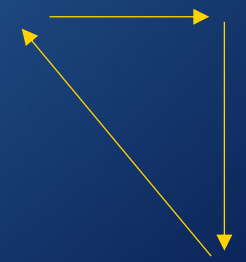


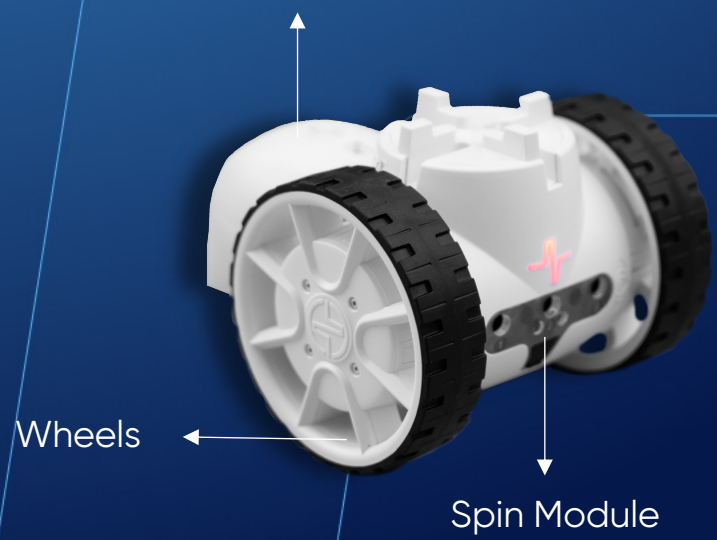

repeat
4
times
drive
50
cm
on
1U51
wait until
has reached target on motor(s)
both
on
1U51
do
spin
90
degrees
on
1U51
wait until
has reached target on motor(s)
both
on
1U51

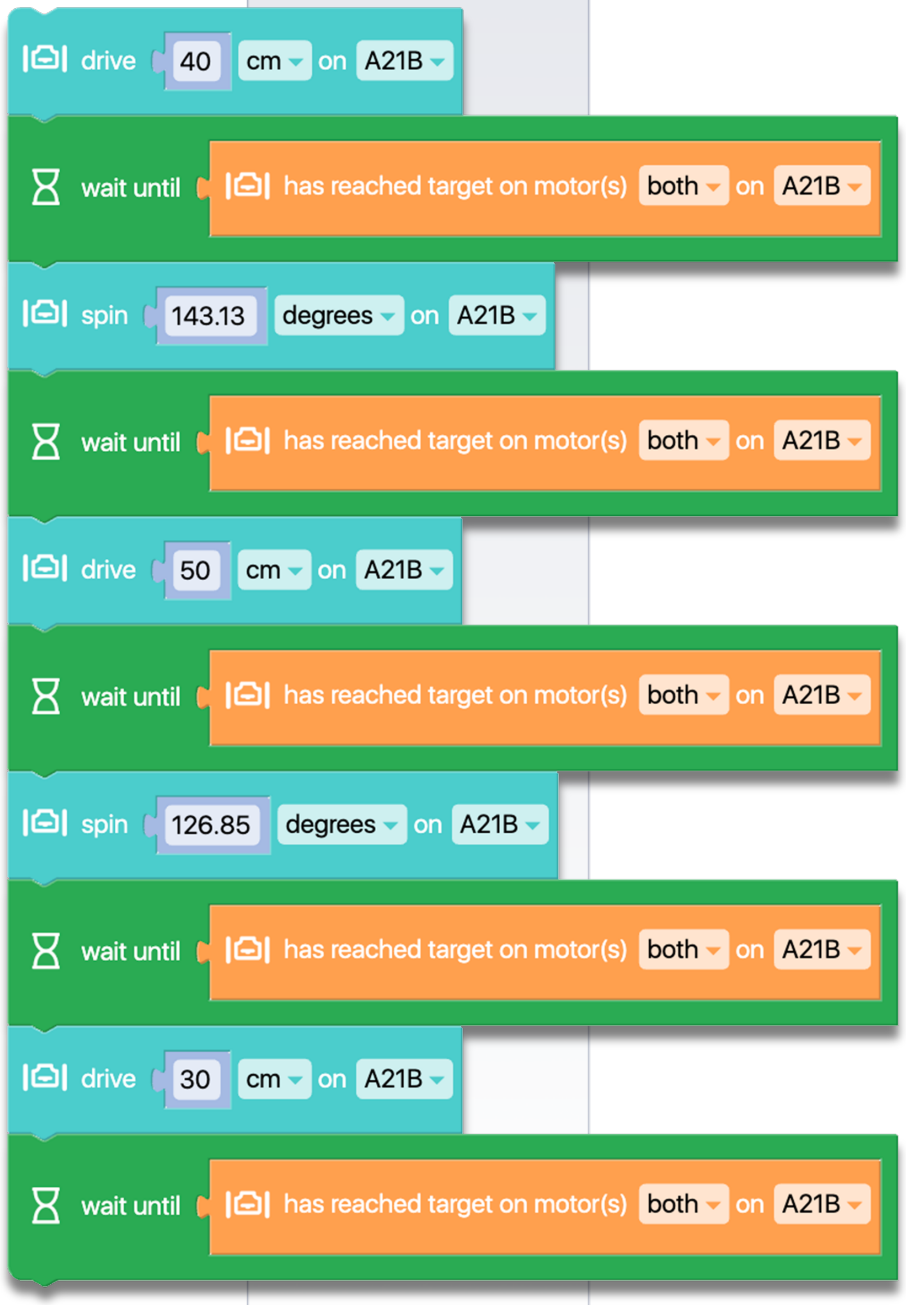
Code 4:
Driving in a Pythagorean Triangle

- The triangle is a Pythagorean one with sides measuring 30, 40, and 50 cm, respectively. Hence, the triangle is a right triangle and follows the formula: the square of the hypotenuse equals the sum of the squares of the legs.
- In this triangle, the angles that are not right angles have the values: 36.87° and 53.13°. Since the rotation is on the outside, the values of the rotation angles are 180° minus the angle values.

WATCH VIDEO







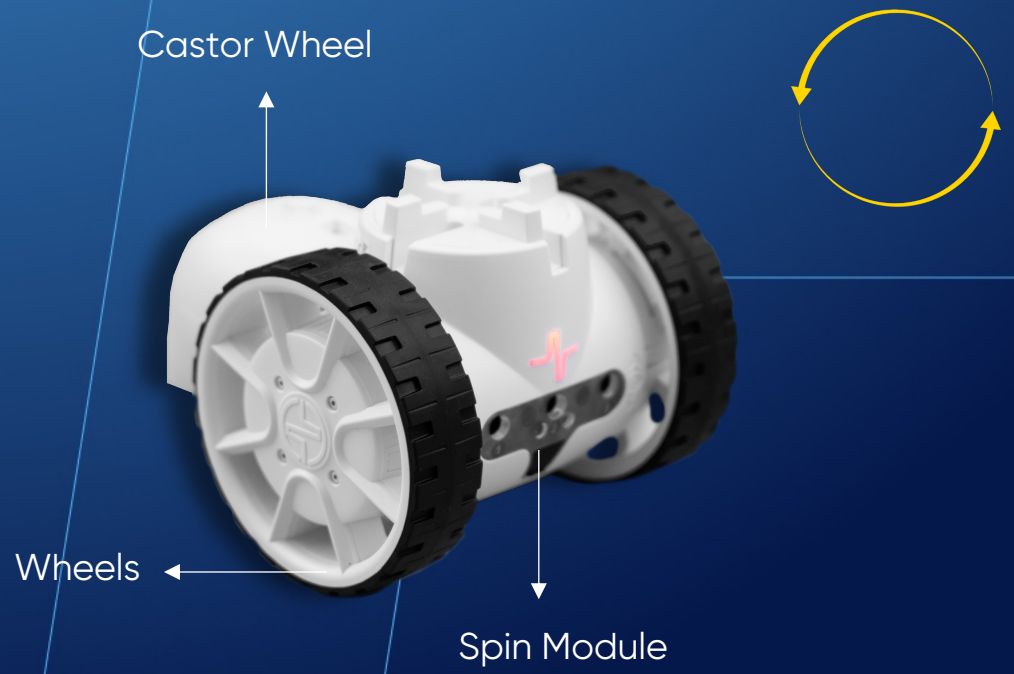
shape robotics
SMART LEARNING

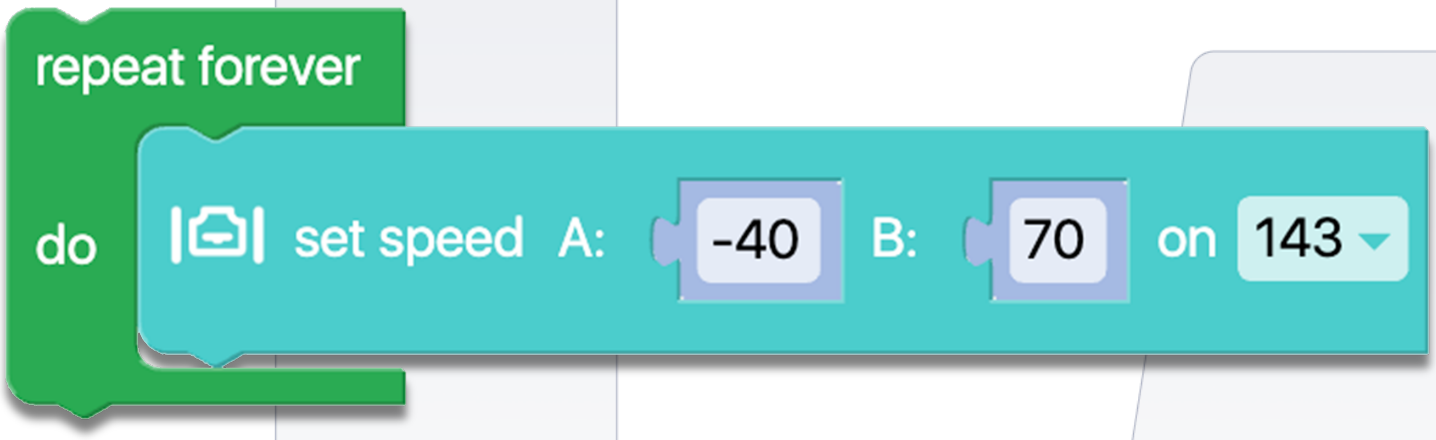
shape robotics
SMART LEARNING

Code 5:
Driving in a Circle

- The code sequence code commands the Spin module to move the two motors at different speeds, resulting in a forward circular motion.

WATCH VIDEO



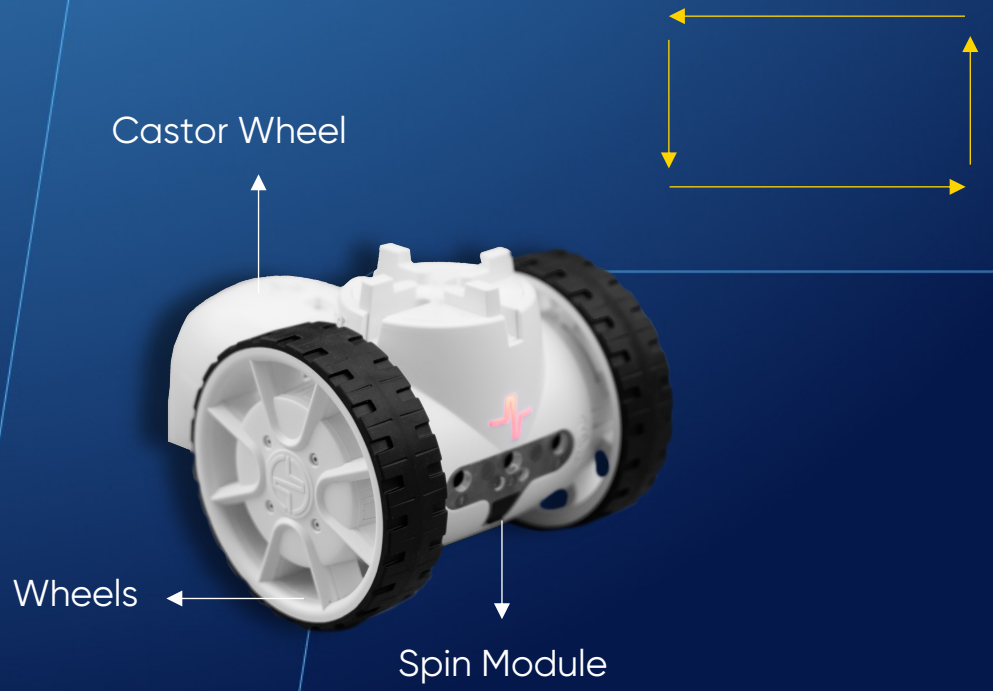

repeat forever
do
set speed A:
-40
B:
70
on
143

shape robotics
SMART LEARNING

Code 6:
Driving in a Rectangular form

- The Spin module moves along the perimeter of a 50cm by 20cm rectangle. The orange block prevents starting a new order before completing the previous one.

WATCH VIDEO

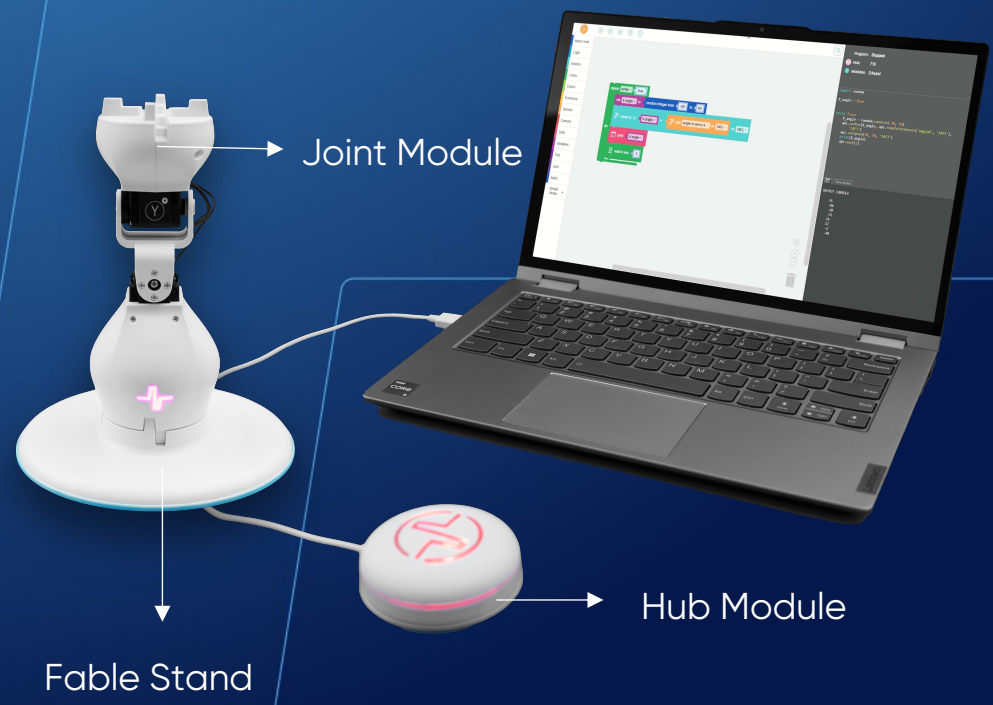


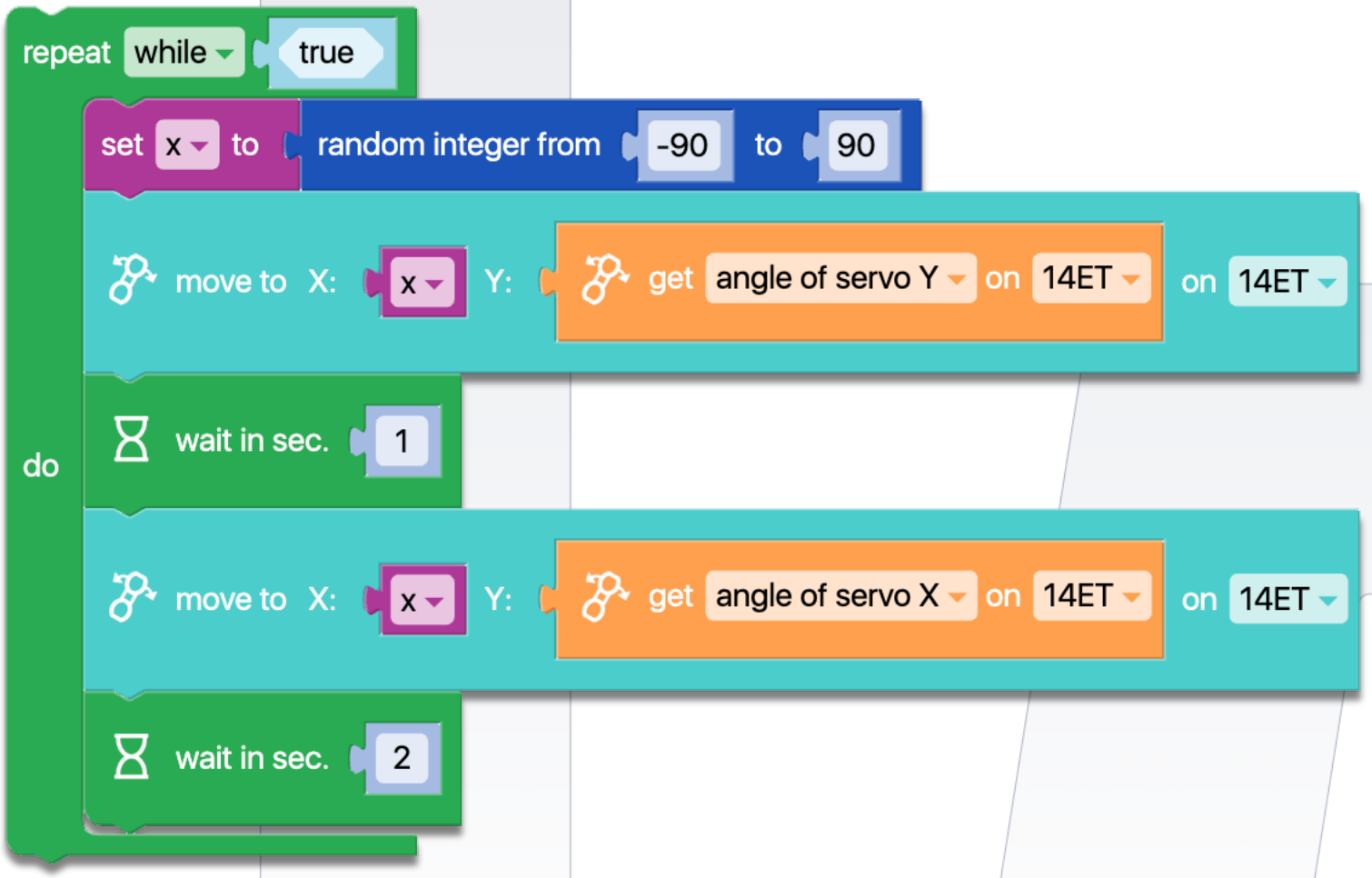
shape robotics
SMART LEARNING

Code 7:
Mirroring motions through Joint's servo-motors

- The program generates a variable and assigns it a random value ranging from -90 to 90 degrees, every two seconds. This variable is then used to move the Joint module. As a result, motor X will be moved to the position specified by the variable, and motor Y will mirror the same movement as motor X.

WATCH VIDEO



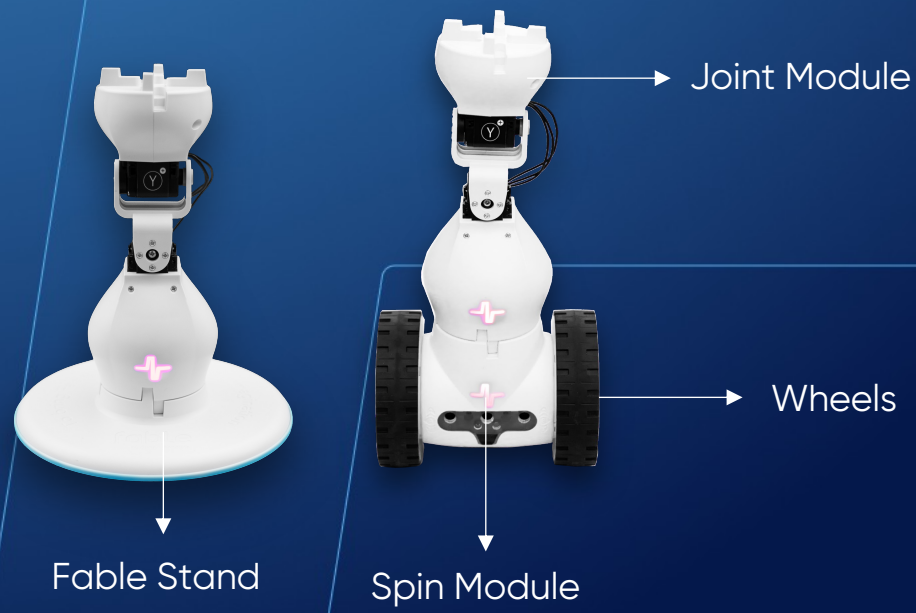

repeat while true
set x to random integer from -90 to 90
move to X: x Y: get angle of servo Y on 14ET on 14ET
do
wait in sec. 1
move to X: x Y: get angle of servo X on 14ET on 14ET
wait in sec. 2

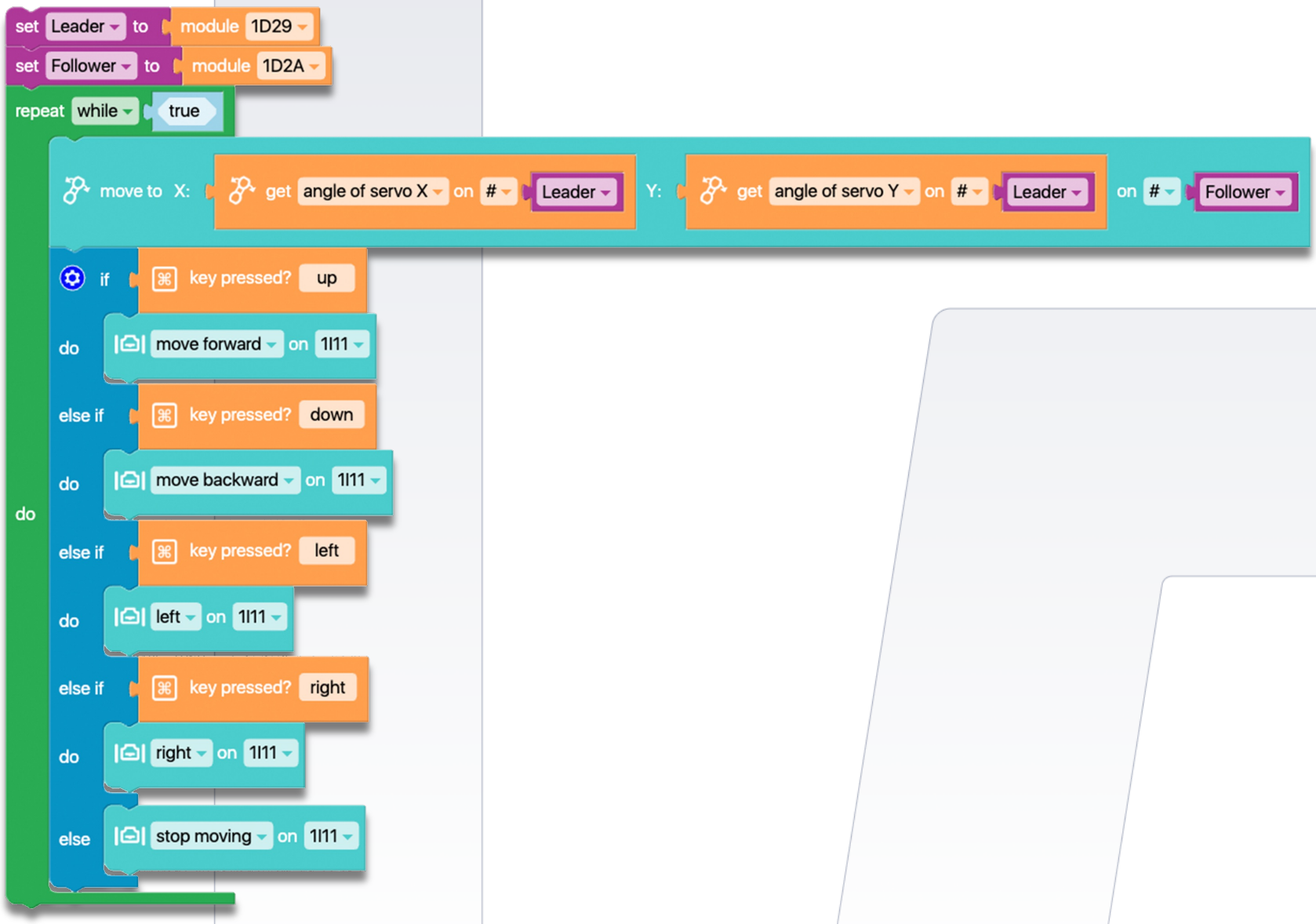
shape robotics
SMART LEARNING

Code 8:
Combination between Spin Key Control and Remote Joint-to-Joint Control

- The program controls the movement of a Spin module using key inputs, as well as the attached Joint module, which is operated by another Joint module. This subassembly can be utilized for remote control purposes.

WATCH VIDEO



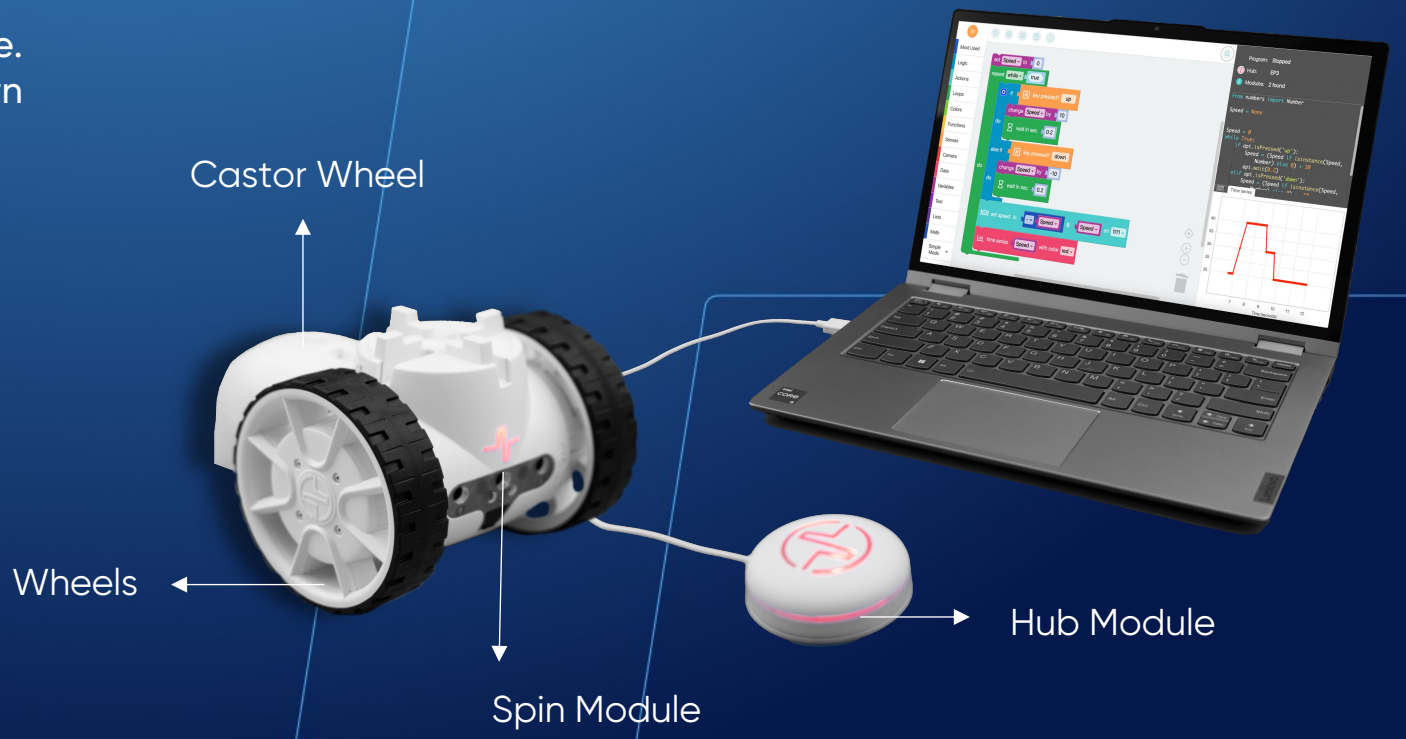

set Leader to module 1D29
set Follower to module 1D2A
repeat while true
move to X: get angle of servo X on # Leader Y: get angle of servo Y on # Leader on # Follower
if key pressed? up
do move forward on 1l11
else if key pressed? down
do move backward on 1l11
else if key pressed? left
do left on 1l11
else if key pressed? right
do right on 1l11
else stop moving on 1l11
do

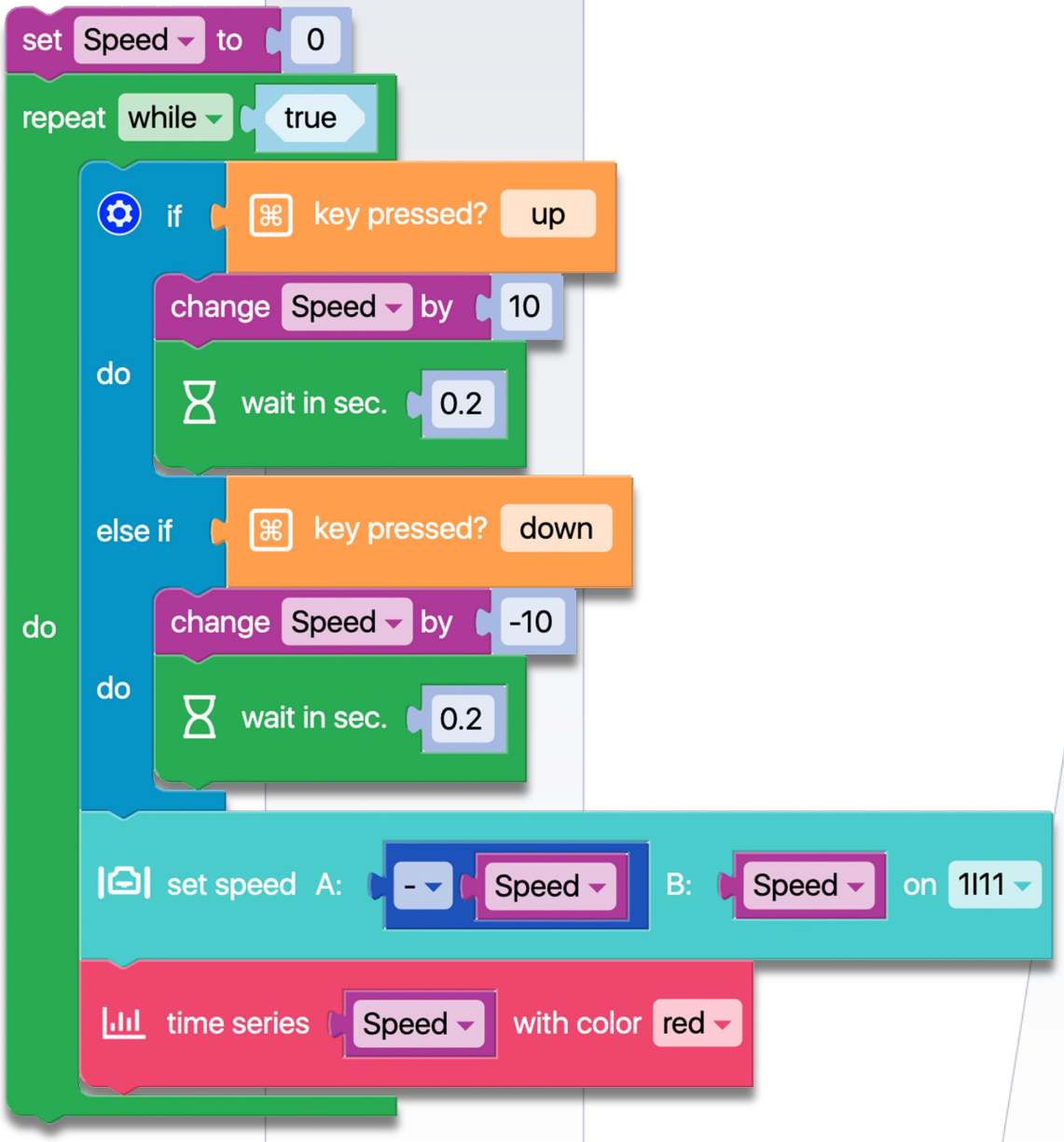
shape robotics
SMART LEARNING

Code 9:
Speed Control for Spin

- The program controls the speed of the Spin module. The Speed variable changes with each up or down key press, while the graph displays its value.

WATCH VIDEO



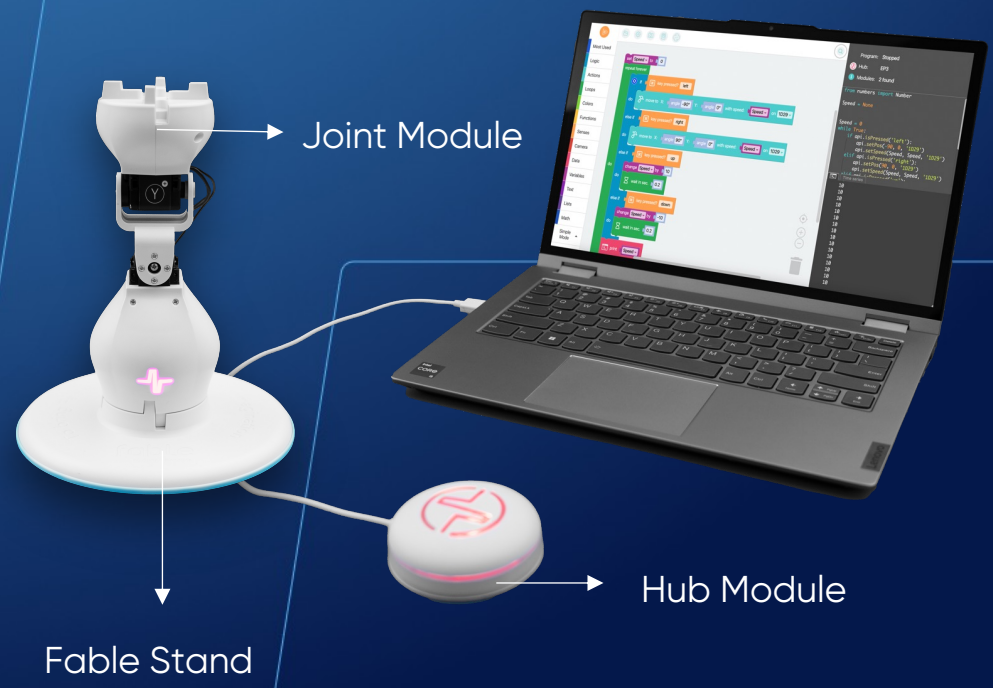

set Speed to 0
repeat while true
if key pressed? up
change Speed by 10
do
wait in sec. 0.2
else if key pressed? down
change Speed by -10
do
wait in sec. 0.2
do
set speed A: - Speed B: Speed on 1l11
time series Speed with color red

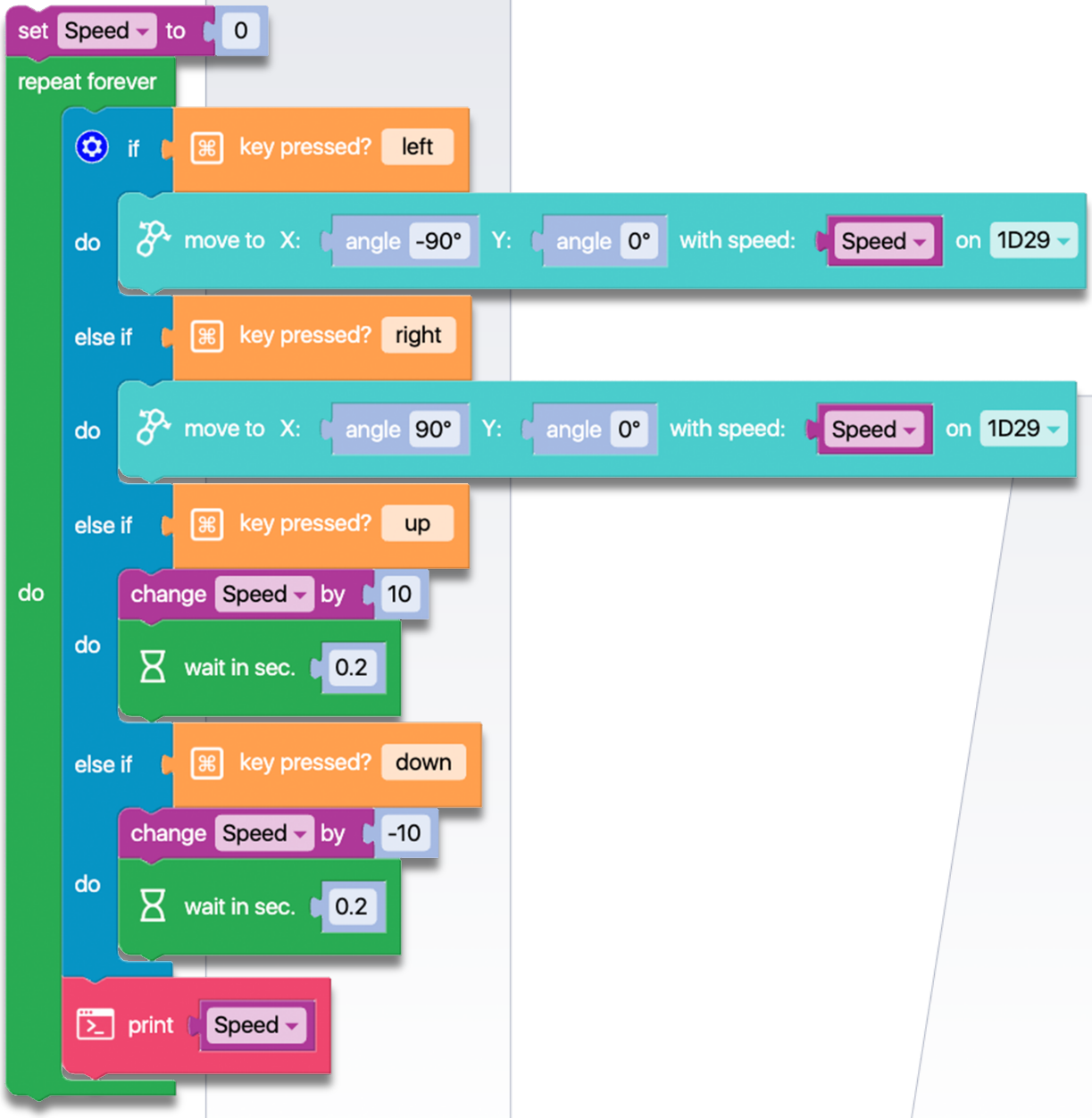
shape robotics
SMART LEARNING

Code 10:
Speed Control for Joint

- The program sequence includes commands for the X motor of a Joint module to move to -90 degrees and 90 degrees, respectively, when the left and right arrow keys are pressed. The speed is determined by a value that increases or decreases by 10 units when the up or down arrow key is pressed. This value is stored in the Speed variable and displayed in the Output console.

WATCH VIDEO





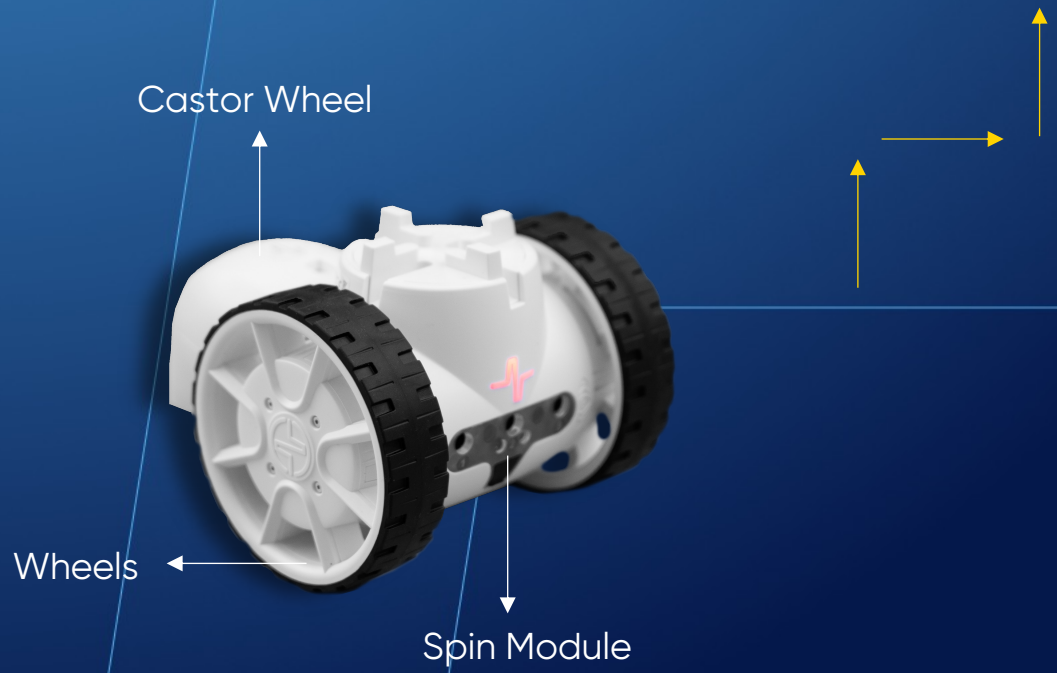
shape robotics
SMART LEARNING

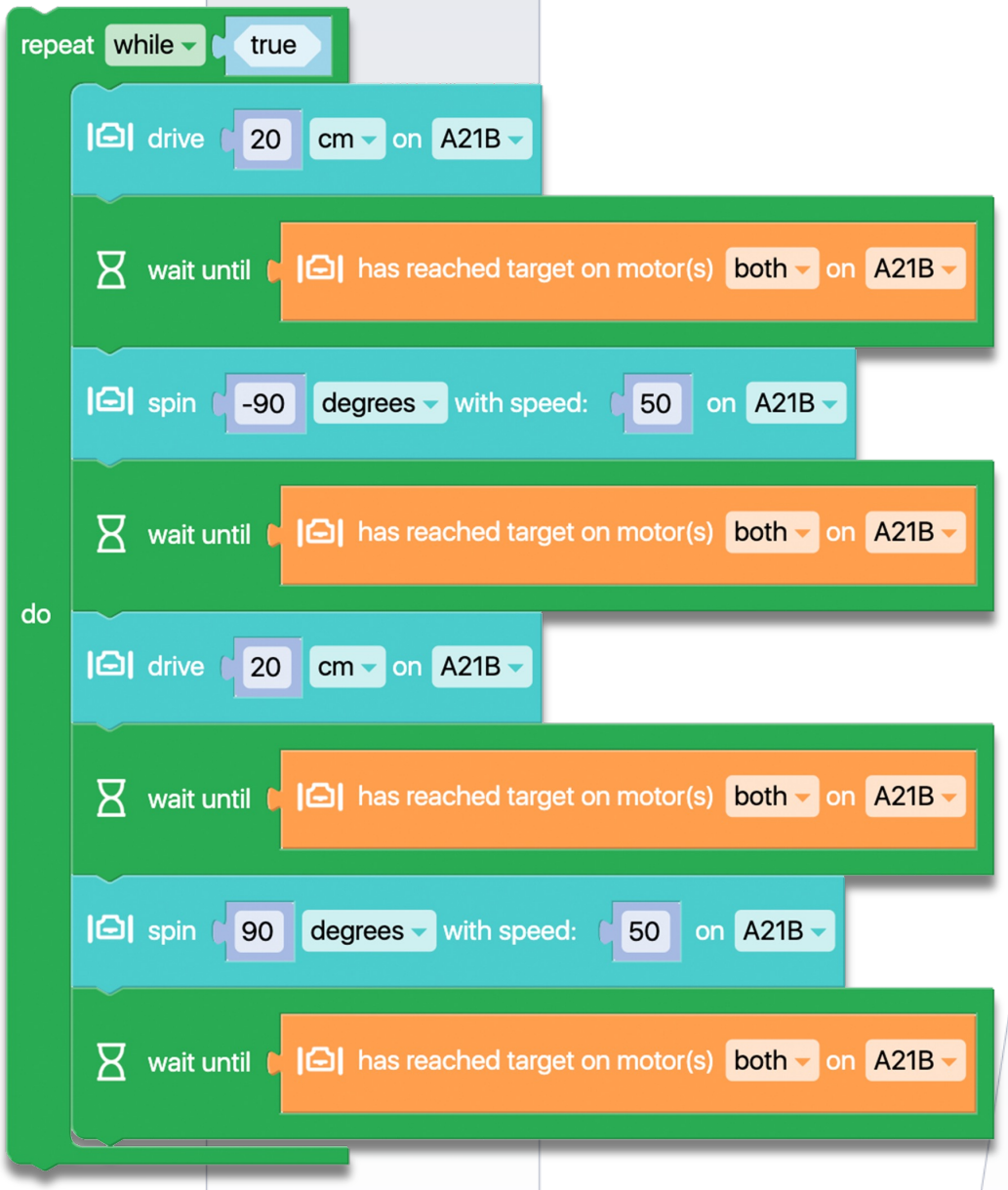
shape robotics
SMART LEARNING

Code 11:
Driving in Zig Zag 90°

- The Spin Robot will move in a pre-determined pattern, a straight line path with alternating right and left 90 degree turns. Changing the angle of rotation creates zig-zag movement patterns.

WATCH VIDEO



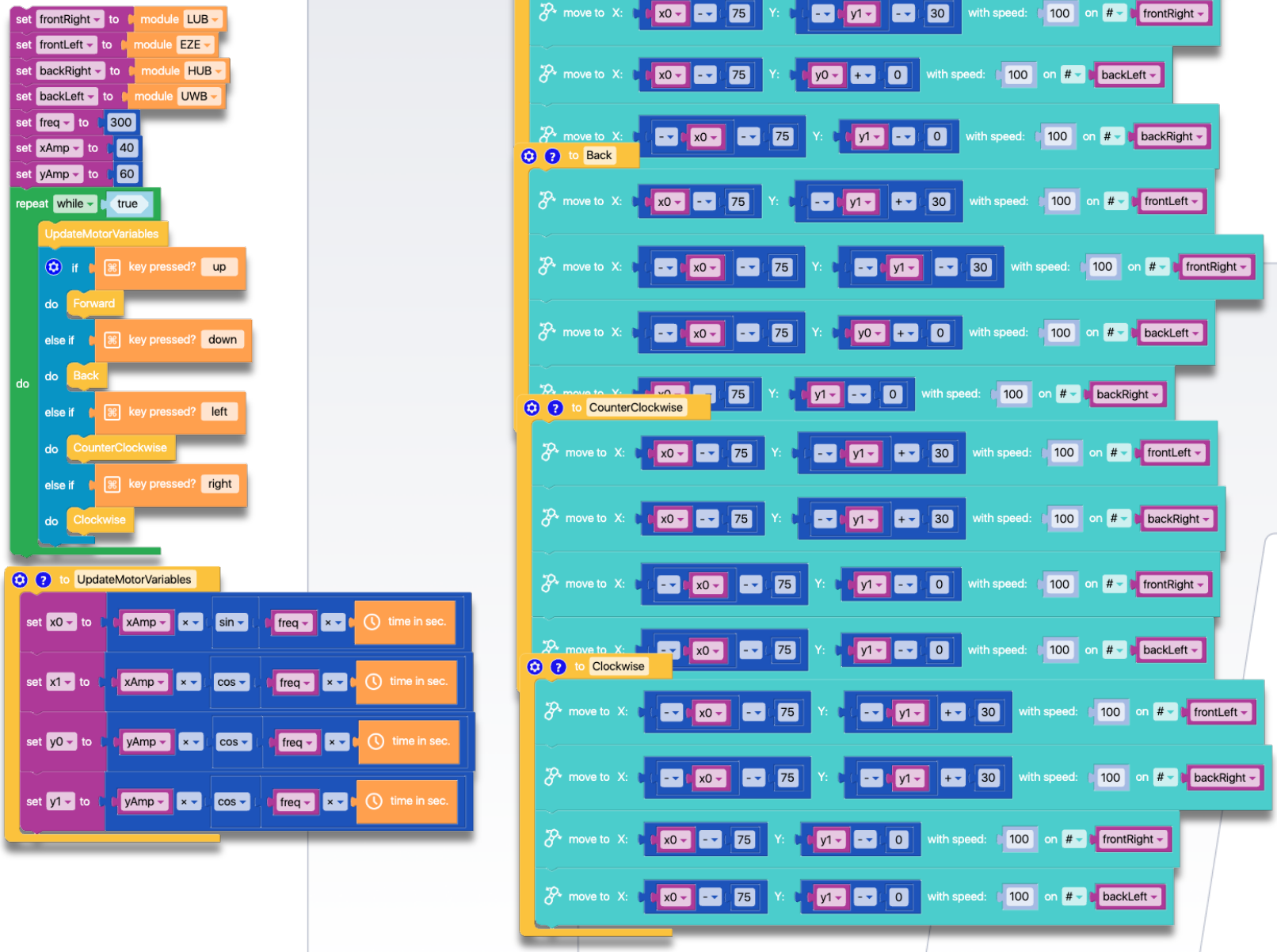

repeat while true
drive 20 cm on A21B
wait until has reached target on motor(s) both on A21B
spin -90 degrees with speed: 50 on A21B
wait until has reached target on motor(s) both on A21B
do
drive 20 cm on A21B
wait until has reached target on motor(s) both on A21B
spin 90 degrees with speed: 50 on A21B
wait until has reached target on motor(s) both on A21B

shape robotics
SMART LEARNING

Code 12:
Quadruped

- The program uses four Joint modules to perform a four-legged robot movement. Given the large number of variables and their use in many places in the command blocks, it is recommended to assemble the robot carefully, as any incorrect placement of a Joint module will result in an unsuccessful movement. From the keys, the robot can be controlled for both forward/backward movement and rotation.

WATCH VIDEO

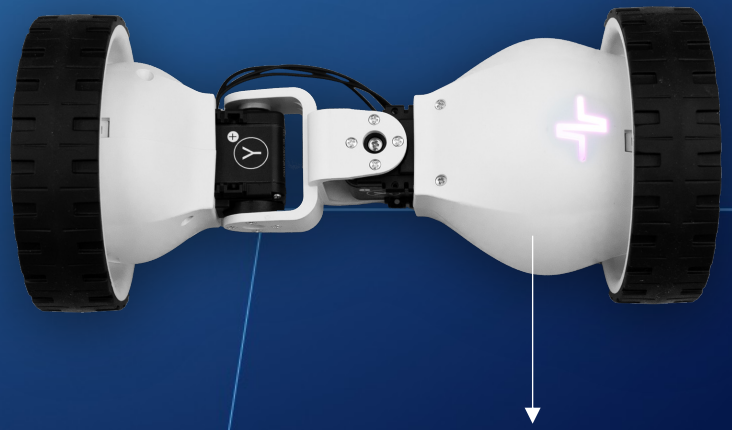

set frontRight to module LUB
set frontLeft to module EZE
set backRight to module HUB
set backLeft to module UWB
set freq to 300
set xAmp to 40
set yAmp to 60
repeat while true
do
UpdateMotorVariables
if key pressed? up
do Forward
else if key pressed? down
do Back
else if key pressed? left
do CounterClockwise
else if key pressed? right
do Clockwise
to UpdateMotorVariables
set x0 to xAmp × sin freq × time in sec.
set x1 to xAmp × cos freq × time in sec.
set y0 to yAmp × cos freq × time in sec.
set y1 to yAmp × cos freq × time in sec.
to Forward
move to X: - x0 - 75 Y: - y1 + 30 with speed: 100 on # frontLeft
move to X: x0 - 75 Y: - y1 - 30 with speed: 100 on # frontRight
move to X: x0 - 75 Y: y0 + 0 with speed: 100 on # backLeft
move to X: - x0 - 75 Y: y1 - 0 with speed: 100 on # backRight
to Back
move to X: x0 - 75 Y: - y1 + 30 with speed: 100 on # frontLeft
move to X: - x0 - 75 Y: - y1 - 30 with speed: 100 on # frontRight
move to X: - x0 - 75 Y: y0 + 0 with speed: 100 on # backLeft
75 Y: y1 - 0 with speed: 100 on # backRight
to CounterClockwise
move to X: x0 - 75 Y: - y1 + 30 with speed: 100 on # frontLeft
move to X: x0 - 75 Y: - y1 + 30 with speed: 100 on # backRight
move to X: - x0 - 75 Y: y1 - 0 with speed: 100 on # frontRight
move to X: x0 - 75 Y: y1 - 0 with speed: 100 on # backLeft
to Clockwise
move to X: - x0 - 75 Y: - y1 + 30 with speed: 100 on # frontLeft
move to X: - x0 - 75 Y: - y1 + 30 with speed: 100 on # backRight
move to X: x0 - 75 Y: y1 - 0 with speed: 100 on # frontRight
move to X: x0 - 75 Y: y1 - 0 with speed: 100 on # backLeft

shape robotics
SMART LEARNING

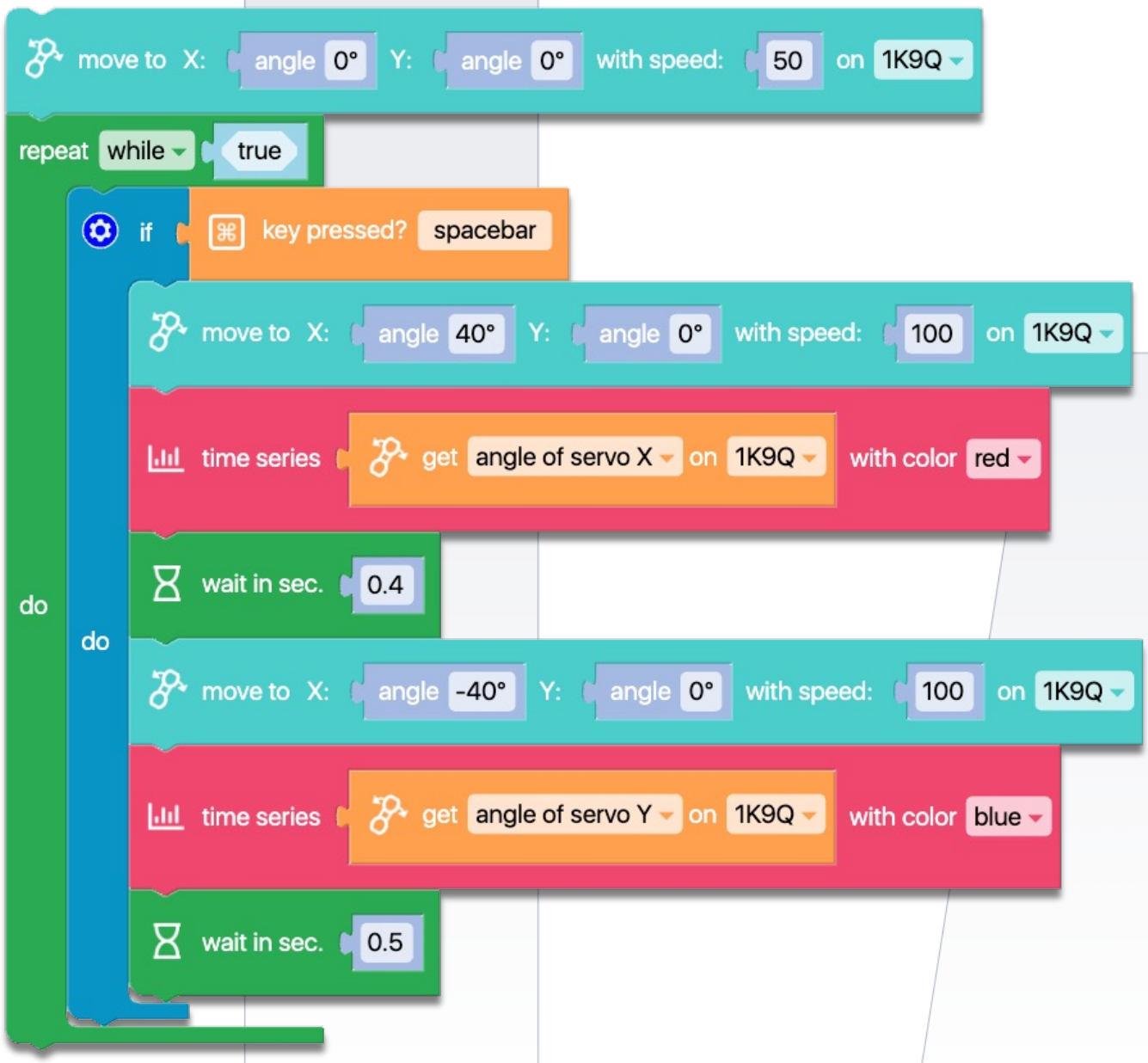
Code 13:
Moving Joint equipped with wheels

- Despite lacking rotating wheels, the Joint module can still be mobilized. This program sequence utilizes the X-motor motion to propel the robot forward by adjusting the gear. The angle value is displayed graphically, facilitating easy comprehension and adjustment of speed or movement mode.

WATCH VIDEO



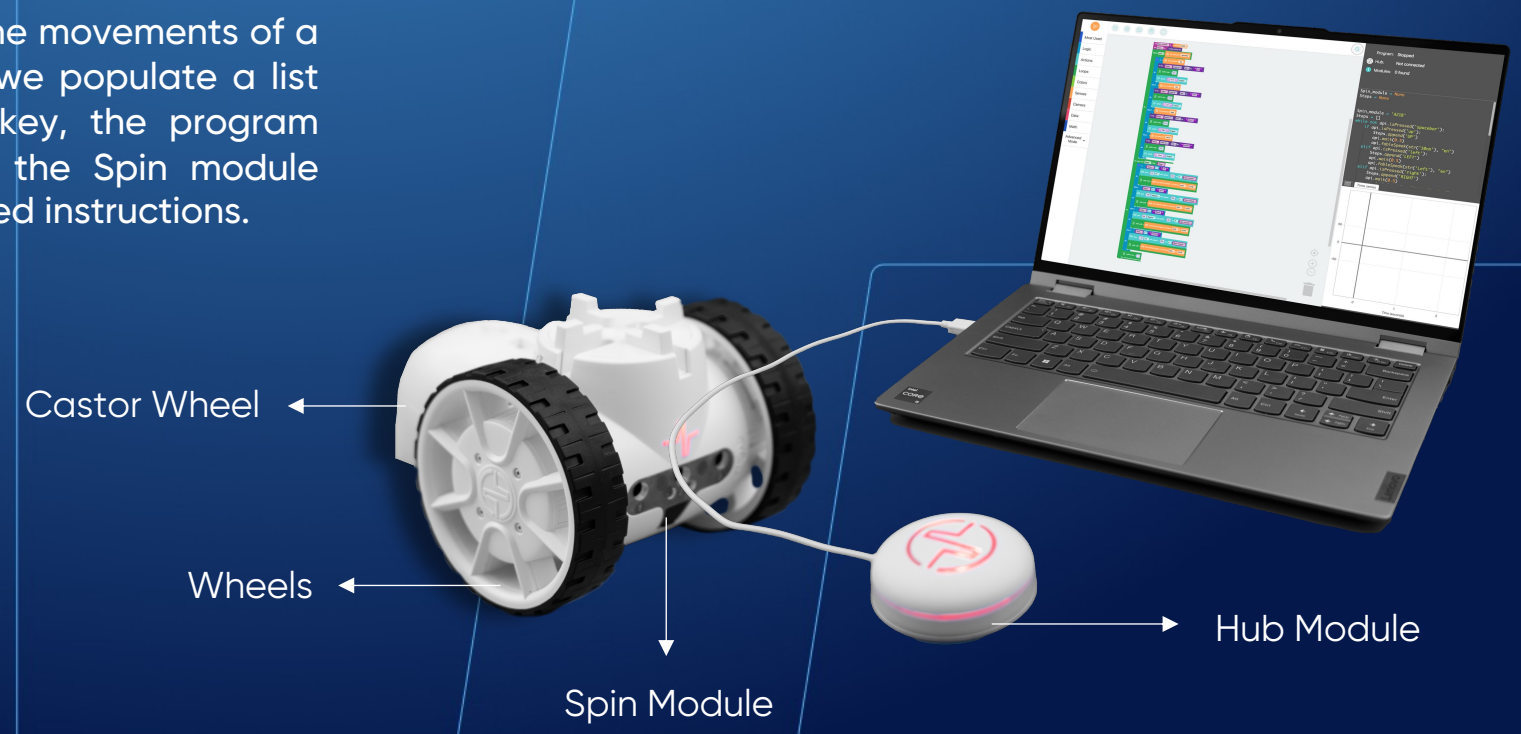
Joint Module

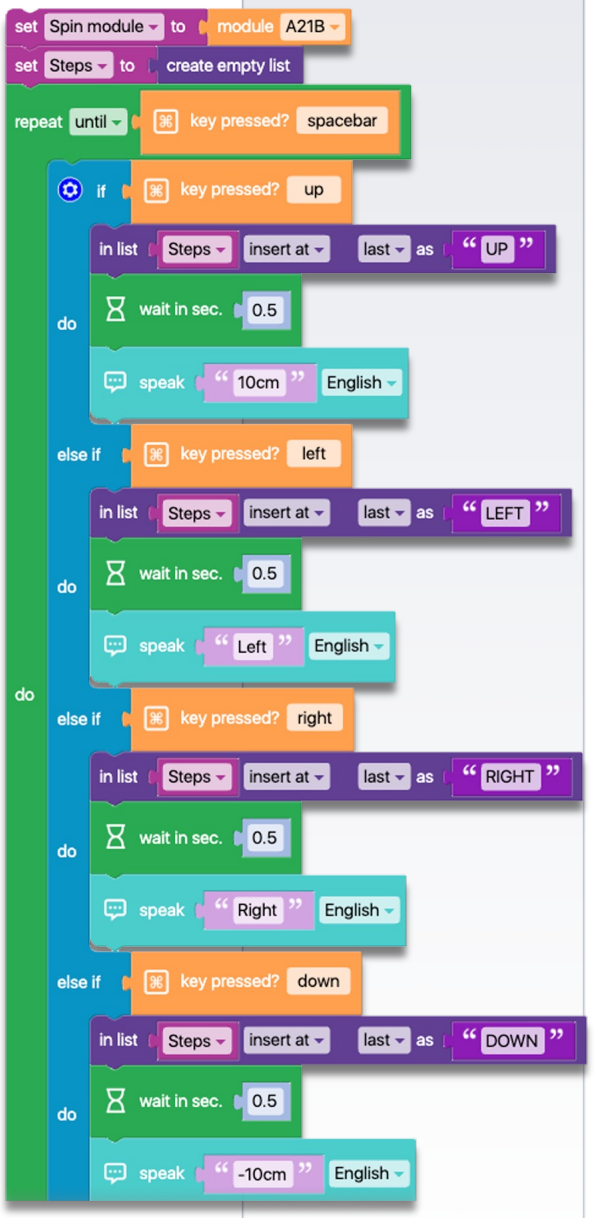

move to X: angle 0° Y: angle 0° with speed: 50 on 1K9Q
repeat while true
do
if key pressed? spacebar
do
move to X: angle 40° Y: angle 0° with speed: 100 on 1K9Q
time series get angle of servo X on 1K9Q with color red
wait in sec. 0.4
move to X: angle -40° Y: angle 0° with speed: 100 on 1K9Q
time series get angle of servo Y on 1K9Q with color blue
wait in sec. 0.5

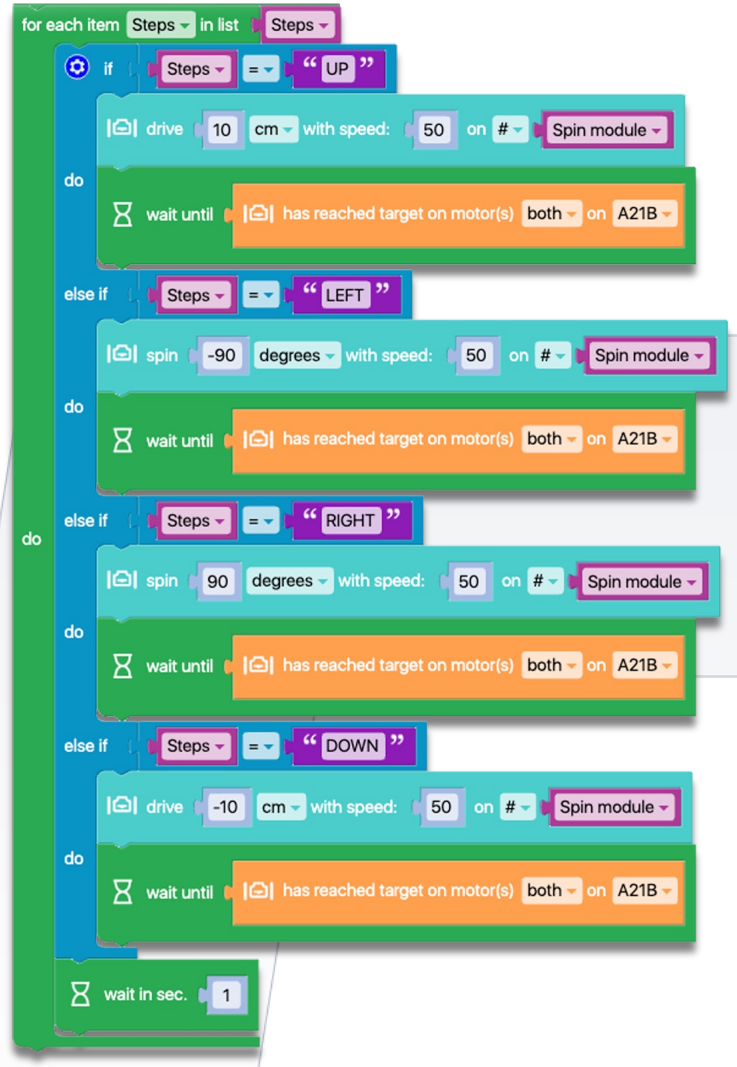
Code 14:
Spin Driving based on Predetermined Directions

- The program enables us to predetermine the movements of a Spin module. By pressing the arrow keys, we populate a list called Steps. Upon pressing the Space key, the program iterates through the list and commands the Spin module to move according to the previously recorded instructions.

WATCH VIDEO




set Spin module to module A21B
set Steps to create empty list
repeat until key pressed? spacebar
do
if key pressed? up
do
in list Steps insert at last as "UP"
wait in sec. 0.5
speak "10cm" English
else if key pressed? left
do
in list Steps insert at last as "LEFT"
wait in sec. 0.5
speak "Left" English
else if key pressed? right
do
in list Steps insert at last as "RIGHT"
wait in sec. 0.5
speak "Right" English
else if key pressed? down
do
in list Steps insert at last as "DOWN"
wait in sec. 0.5
speak "-10cm" English

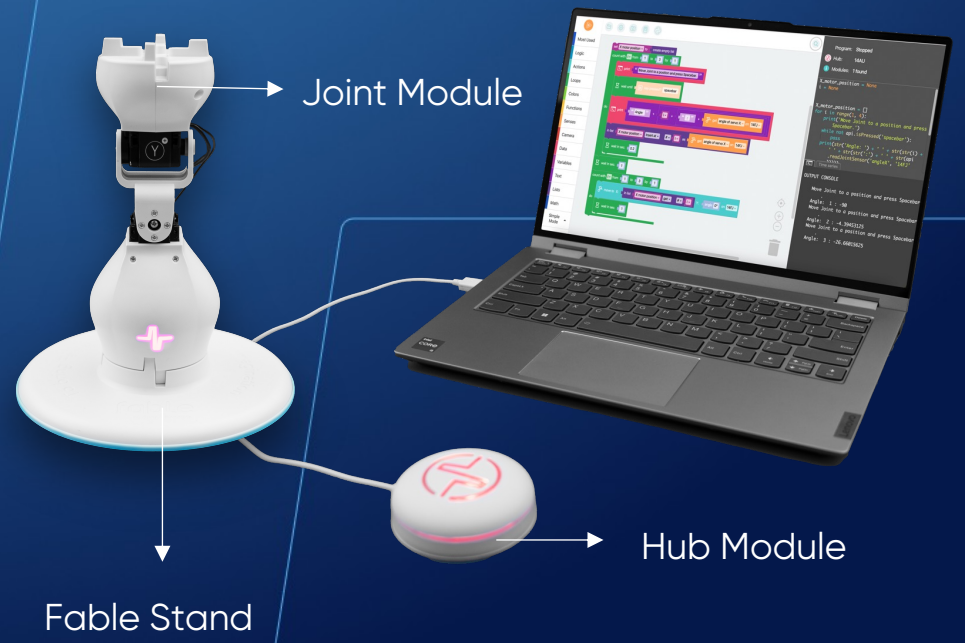

for each item Steps in list Steps
do
if Steps = "UP"
do
drive 10 cm with speed: 50 on # Spin module
wait until has reached target on motor(s) both on A21B
else if Steps = "LEFT"
do
spin -90 degrees with speed: 50 on # Spin module
wait until has reached target on motor(s) both on A21B
else if Steps = "RIGHT"
do
spin 90 degrees with speed: 50 on # Spin module
wait until has reached target on motor(s) both on A21B
else if Steps = "DOWN"
do
drive -10 cm with speed: 50 on # Spin module
wait until has reached target on motor(s) both on A21B
wait in sec. 1

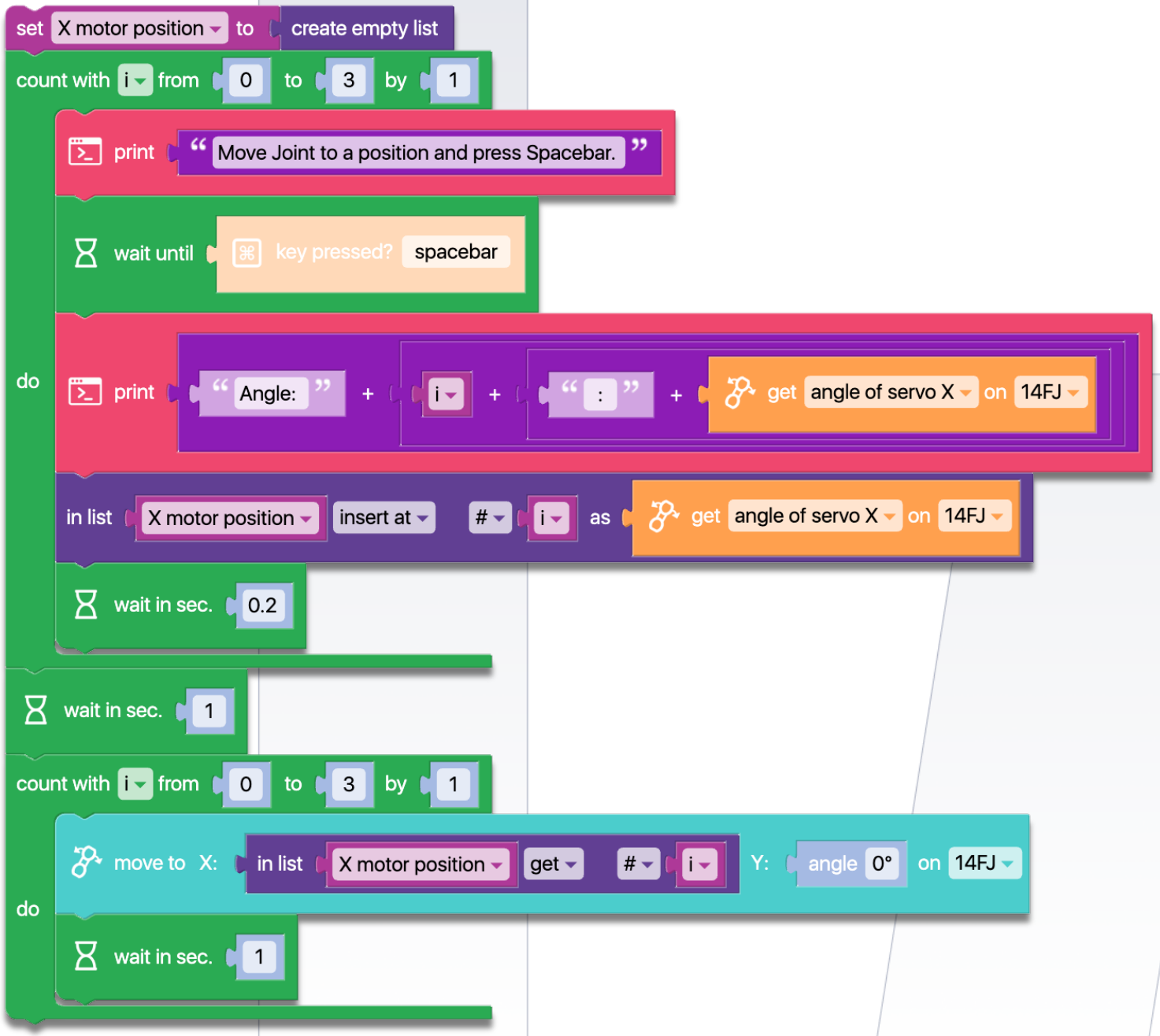
shape robotics
SMART LEARNING

Code 15:
Joint Moving based on Predetermined Directions

- The program stores three values of the X motor angle on the Joint. Position the X motor at an angle and press the Space key to save it. The storage is done using a list, which is then used to retrieve the stored values. With these values, the Joint module will replicate the learned moves.

WATCH VIDEO





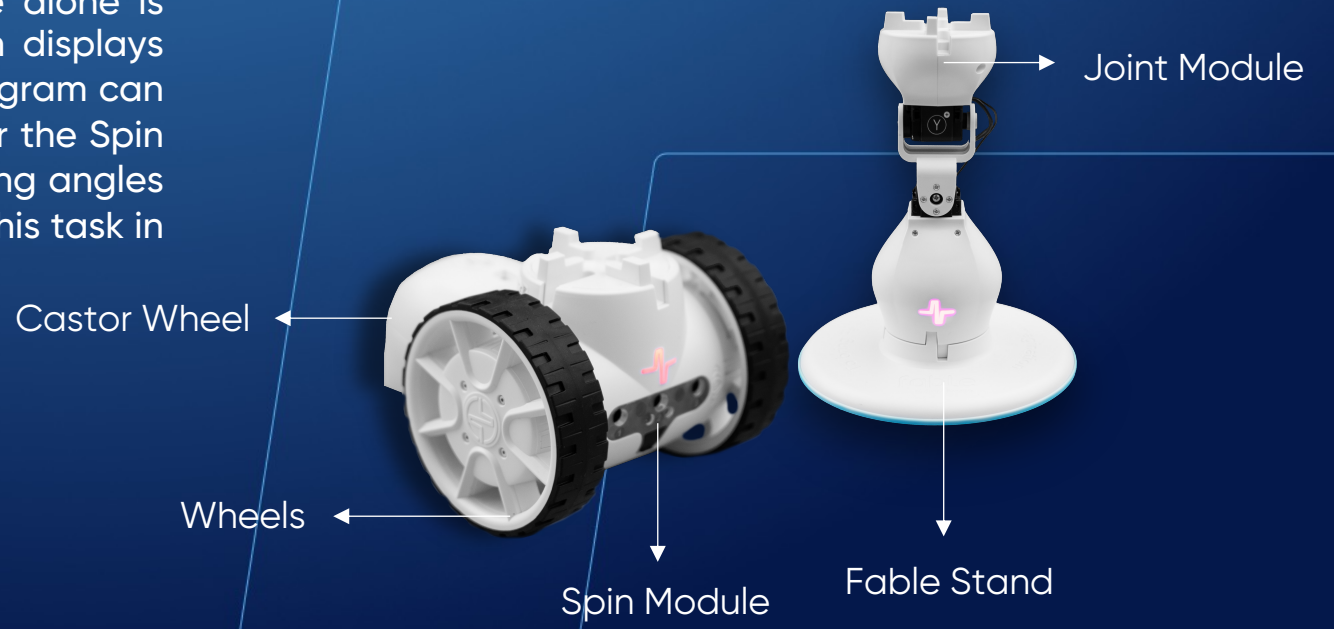
shape robotics
SMART LEARNING

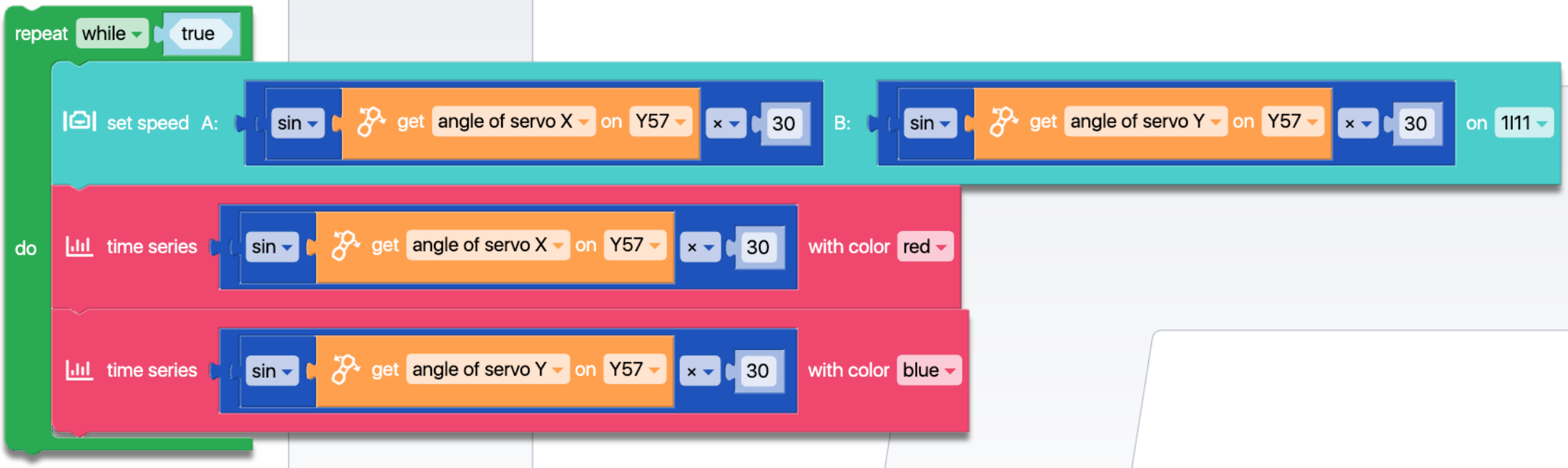
shape robotics
SMART LEARNING

Code 16:
Trigonometry Motion

- The program utilizes the motor angles on the Joint module to calculate their sine values. These values are then multiplied and transmitted to the two Spin motors to control their movements. The multiplication is essential because the sine value alone is insufficient to move the motors effectively. The graph displays data received by the Spin motors. Additionally, this program can be employed as a game - the objective is to maneuver the Spin to a specific location as swiftly as possible (by combining angles for the Joint motors). The individual who accomplishes this task in the shortest time wins the game!

WATCH VIDEO



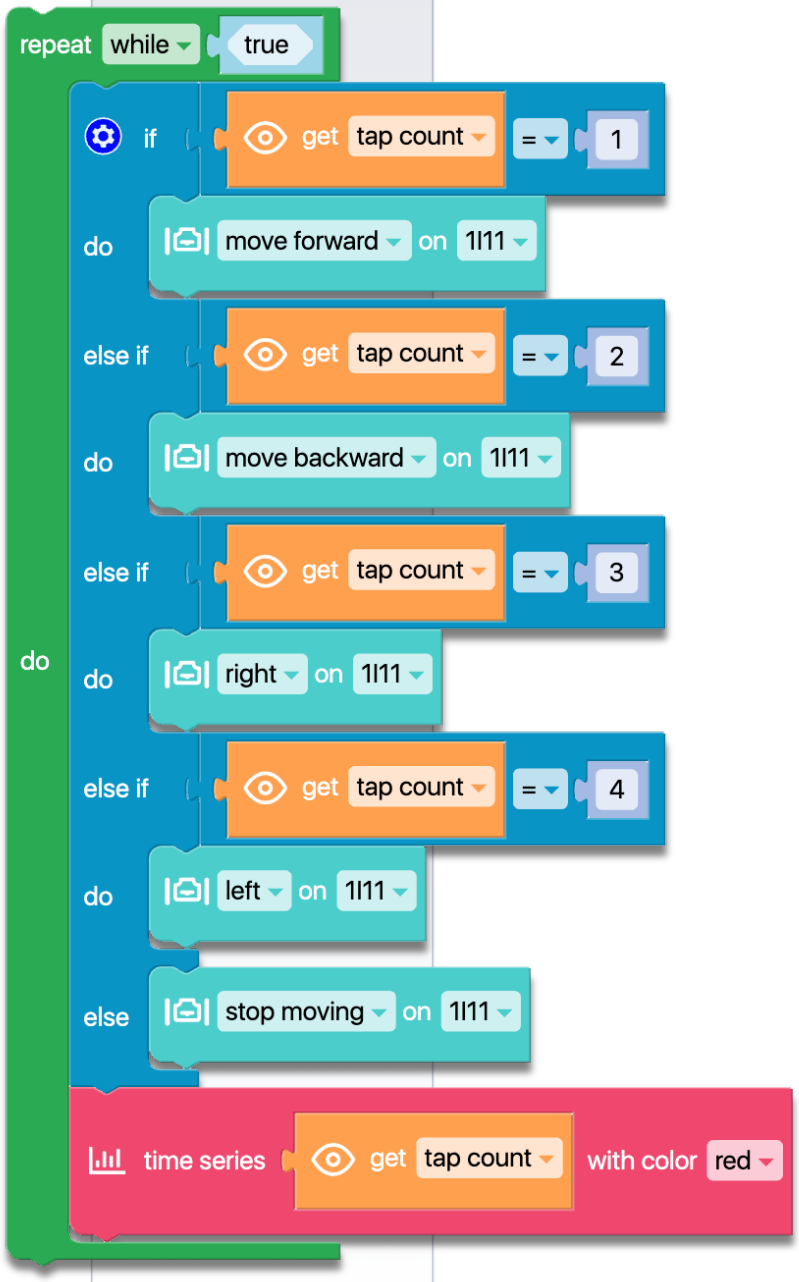

repeat
while
true
do
set speed A:
sin
get
angle of servo X
on
Y57
×
30
B:
sin
get
angle of servo Y
on
Y57
×
30
on
1l11
time series
sin
get
angle of servo X
on
Y57
×
30
with color
red
time series
sin
get
angle of servo Y
on
Y57
×
30
with color
blue

shape robotics
SMART LEARNING

Code 17:
Finger Touch Control for Spin

- The code sequence governs the movement of the Spin module based on the number of fingers detected on the Fable Face phone, which is connected to the computer where the programming is executed. The graph illustrates the number of fingers detected at each moment.

WATCH VIDEO

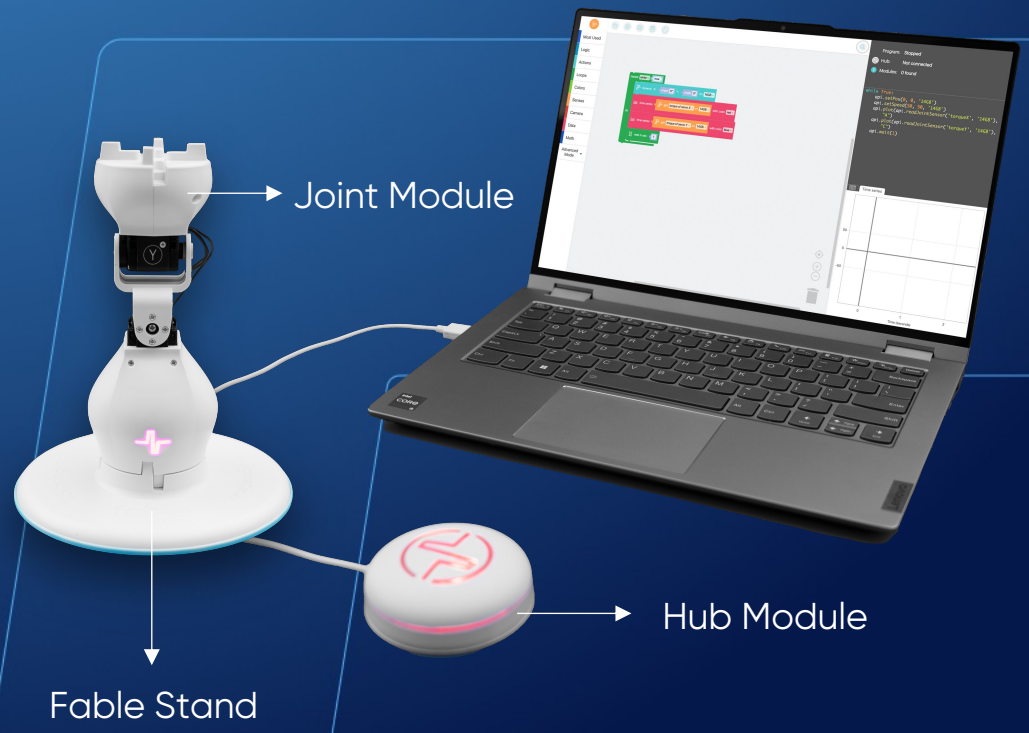

repeat while true
do
if get tap count = 1
do move forward on 1I11
else if get tap count = 2
do move backward on 1I11
else if get tap count = 3
do right on 1I11
else if get tap count = 4
do left on 1I11
else stop moving on 1I11
time series get tap count with color red

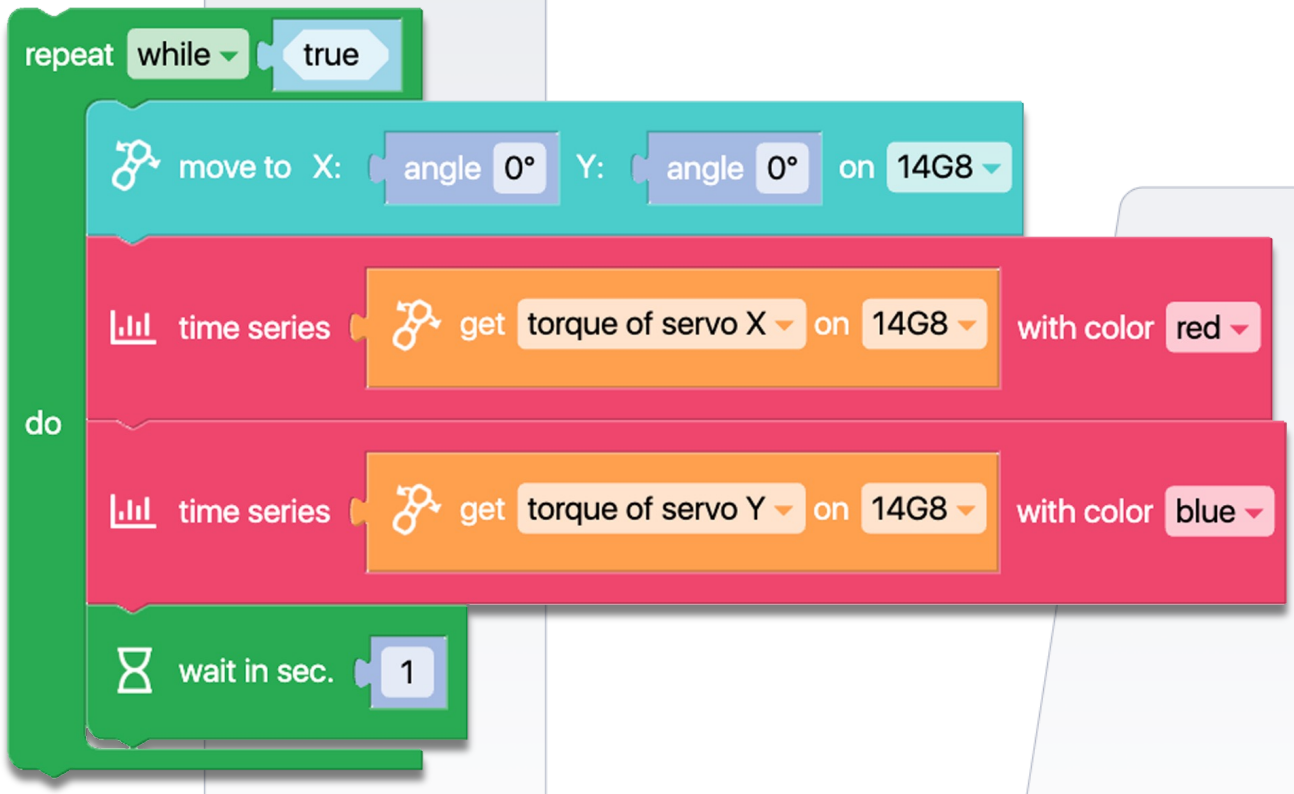
shape robotics
SMART LEARNING

Code 18:
Measuring Joint Torque

- The program continually monitors the torque on the X and Y motors of the Joint robotic mode. Each reading is promptly displayed in the Output Console and can also be saved in a .csv file. To enhance readability, a text block has been employed, incorporating the values read by the sensors. This feature offers a clearer and more comprehensible display. Furthermore, the program can be further improved by implementing a comparison operator command to detect instances where the torque exceeds a predefined threshold. This addition would contribute to a display that is even more user-friendly and easy to interpret.

WATCH VIDEO



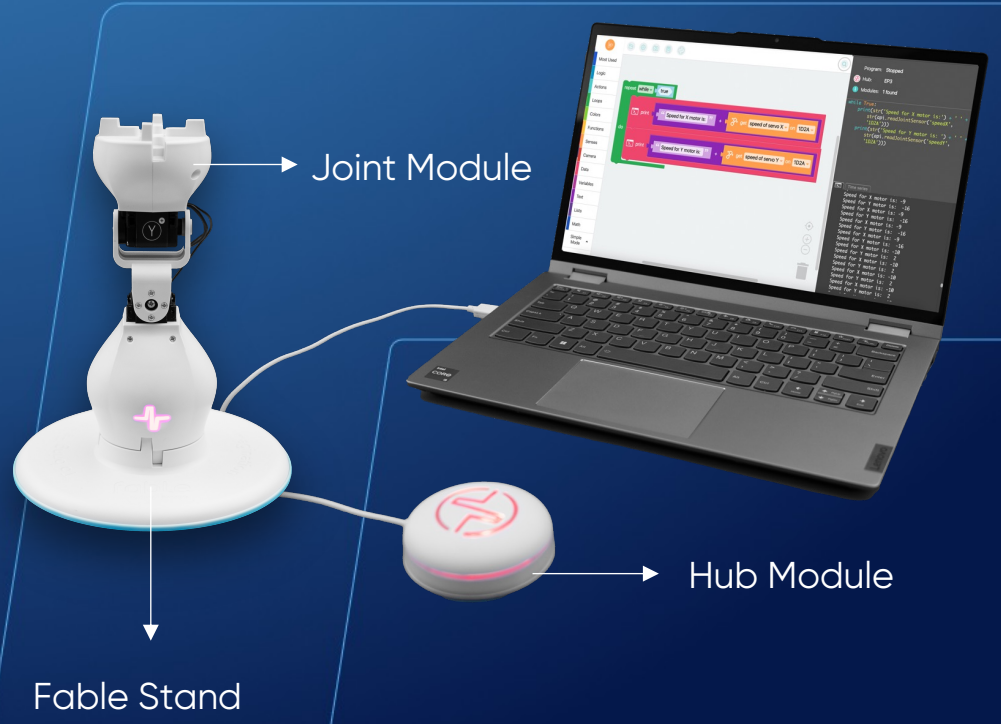

repeat while true
move to X: angle 0° Y: angle 0° on 14G8
do
time series get torque of servo X on 14G8 with color red
time series get torque of servo Y on 14G8 with color blue
wait in sec. 1

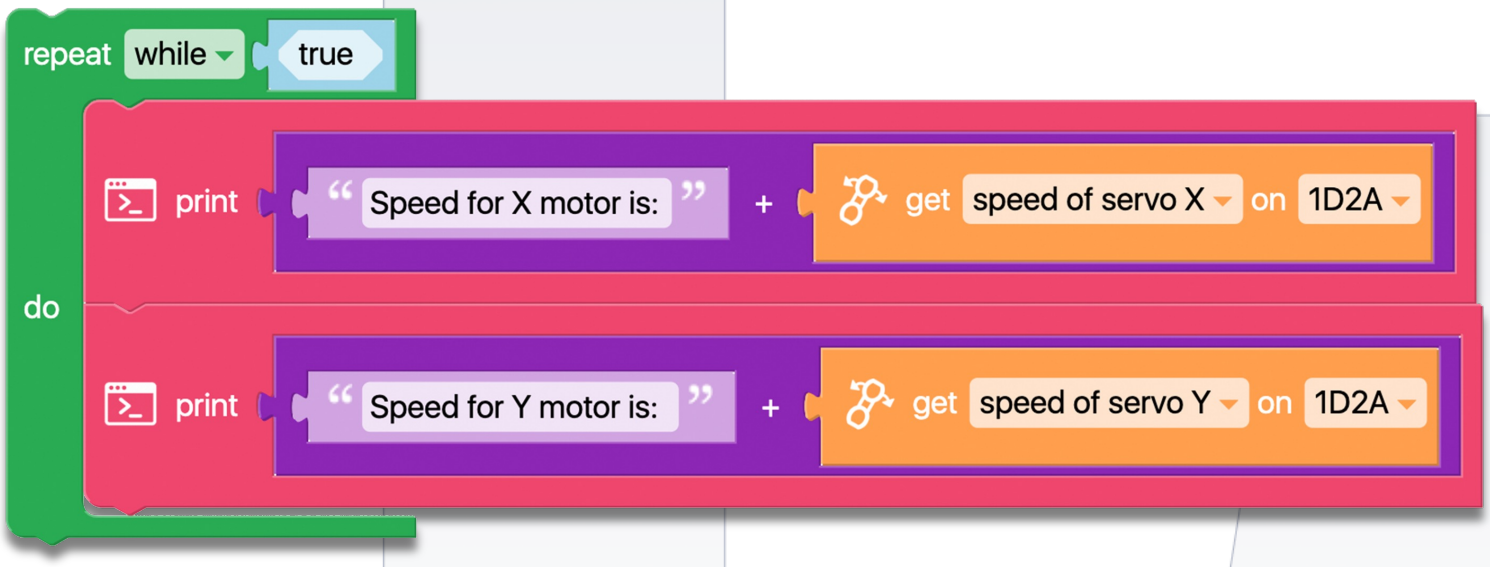
shape robotics
SMART LEARNING

Code 19:
Measuring Joint's servo-motors speed

- The program employs blocks to "Measuring Joint Torque", this time the speed at which the X and Y motors of the Joint mode move are read. The display is presented in numerical format, using the Output console. This program can be enhanced with a comparison operator command to detect when the speed surpasses a certain value.

WATCH VIDEO



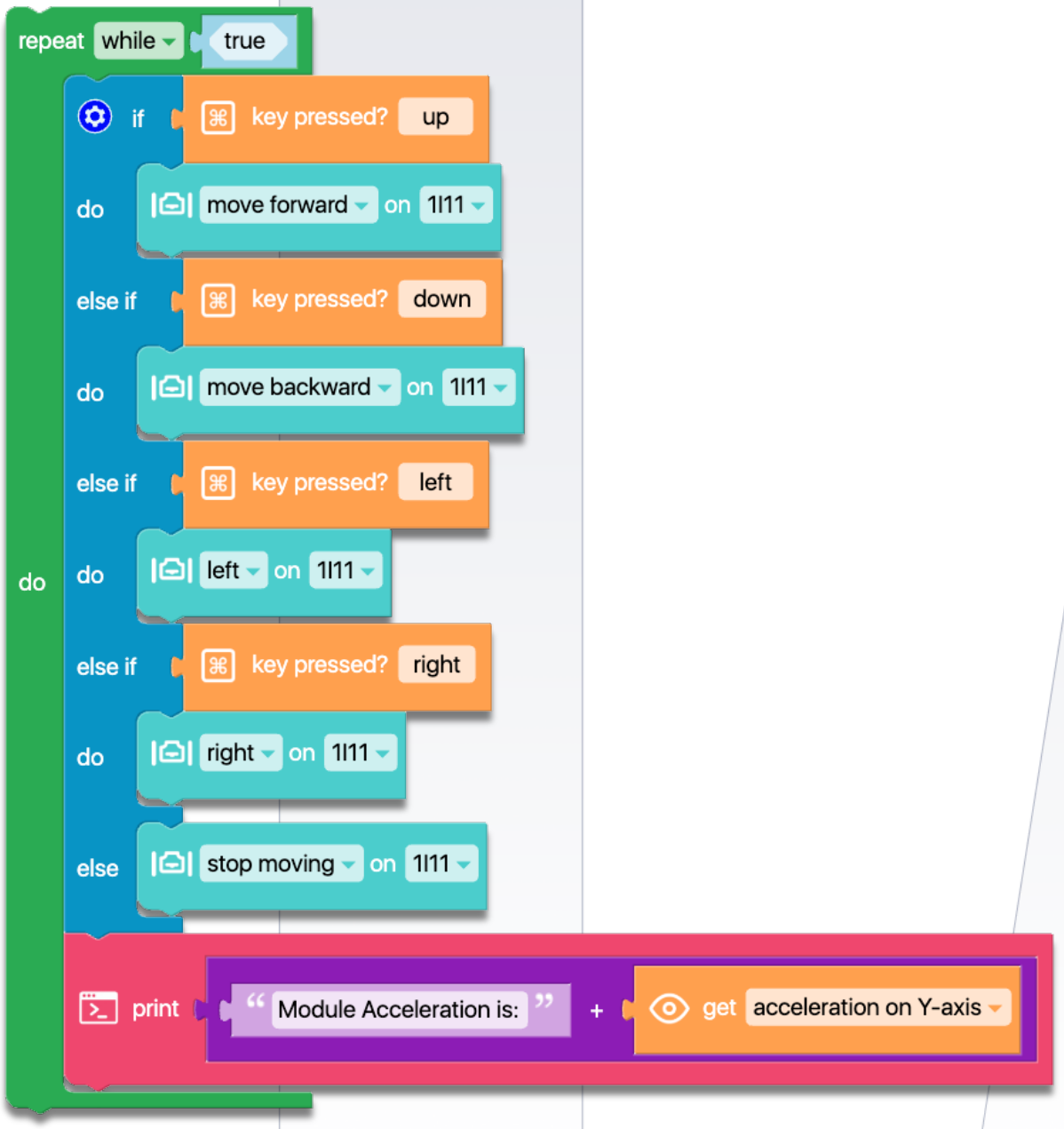

repeat
while
true
do
print
“ Speed for X motor is: ”
+
get
speed of servo X
on
1D2A
print
“ Speed for Y motor is: ”
+
get
speed of servo Y
on
1D2A

shape robotics
SMART LEARNING

Code 20:
Measuring Spin Acceleration

- The program enables you to operate a Spin module using the directional keys on your keyboard. At the same time, it displays the acceleration value recorded on one of the axes of movement on your phone which is placed on the Spin. Please note that Fabel Face is turned on and connected to the Hub.

WATCH VIDEO

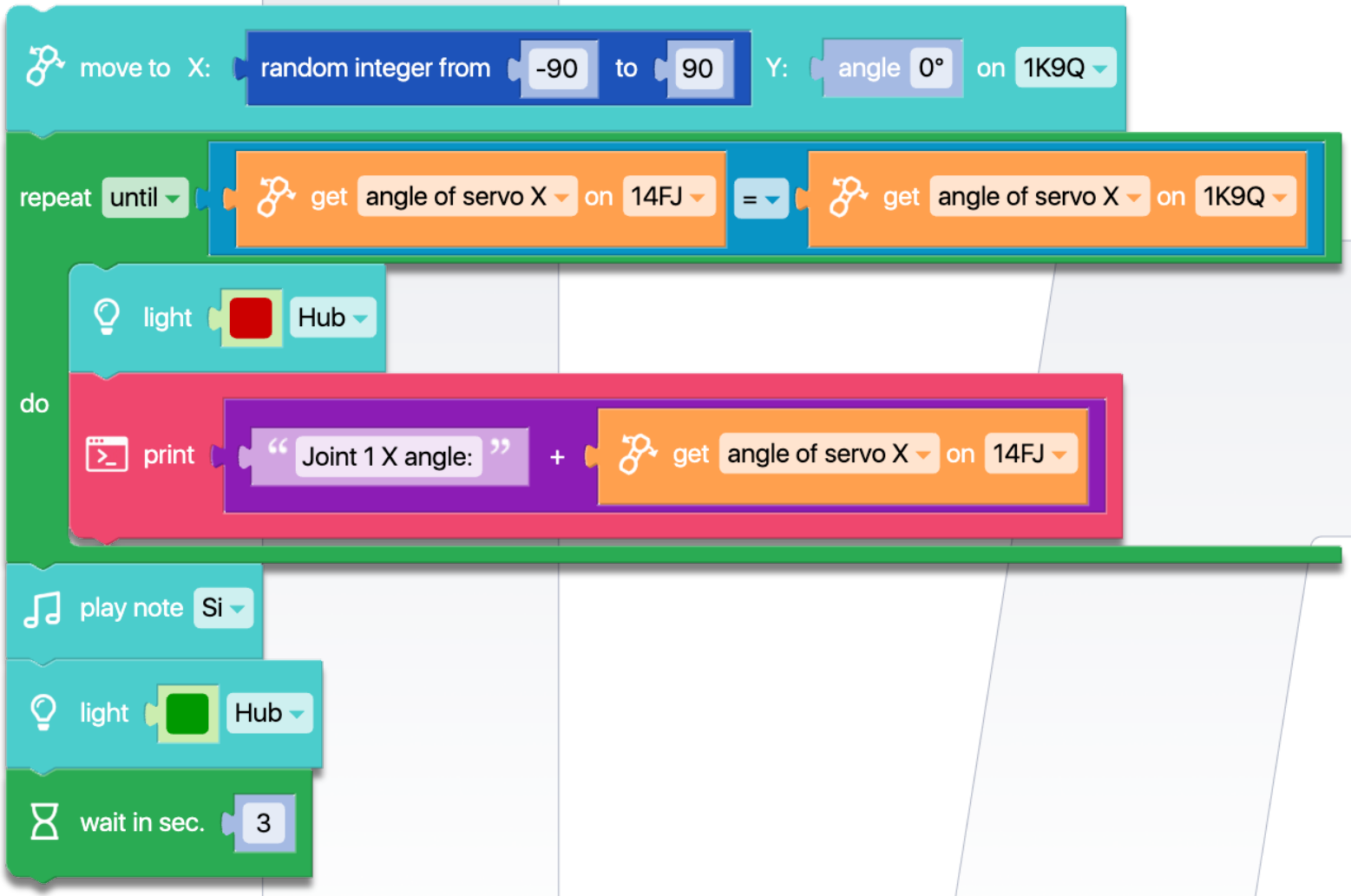


shape robotics
SMART LEARNING

Code 21:
Compare two values

- This program can also function as a game. Upon execution, the X motor of a Joint module will begin at a random angle between -90 and 90 degrees. The goal of the game is for the user to adjust the position of the other Joint module until the angle of the first motor is determined. Once the user identifies the angle, the Output Console will display the angle, the Hub will illuminate in green, and the musical note 'Si' will play. It's crucial to refrain from applying excessive force to the motors by rotating the Joint module too forcefully or rapidly.

WATCH VIDEO

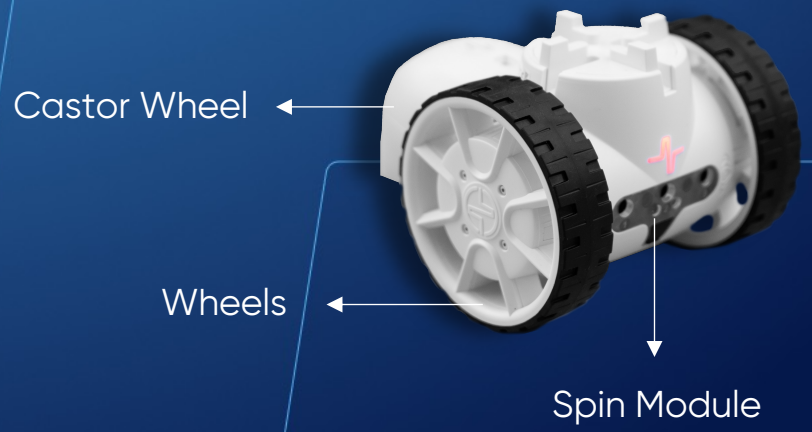

move to X:
random integer from
-90
to
90
Y:
angle
0°
on
1K9Q
repeat
until
get
angle of servo X
on
14FJ
=
get
angle of servo X
on
1K9Q
light
Hub
do
print
“
Joint 1 X angle:
”
+
get
angle of servo X
on
14FJ
play note
Si
light
Hub
wait in sec.
3

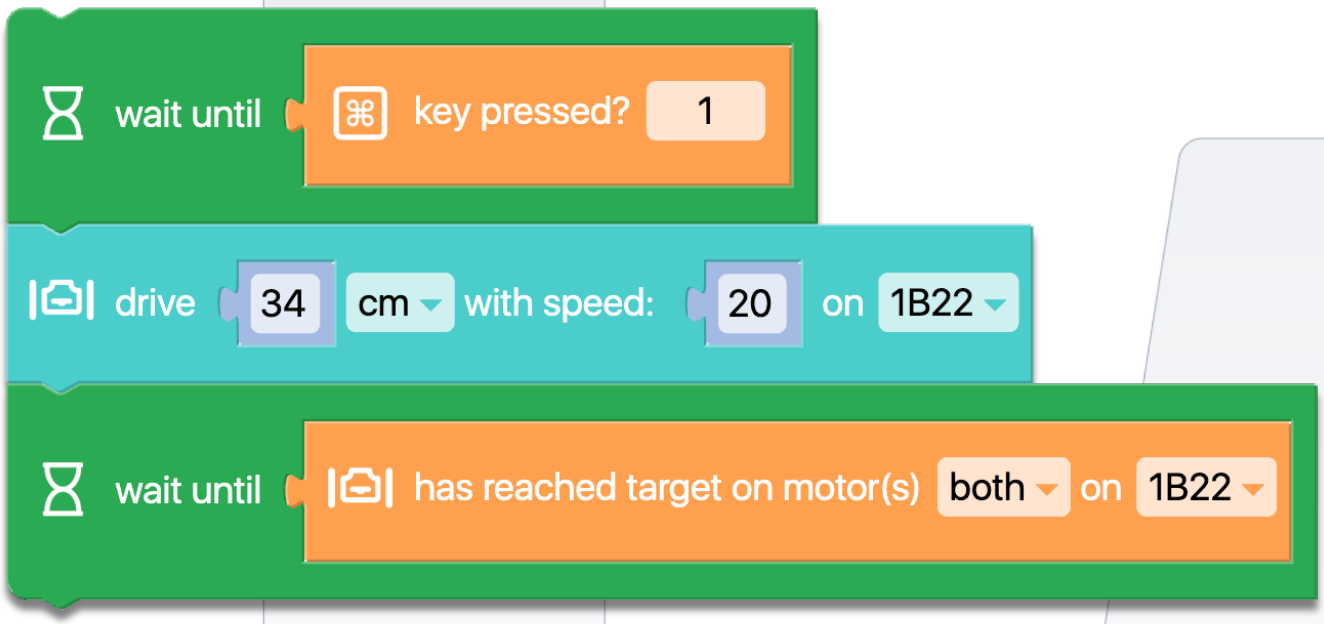
shape robotics
SMART LEARNING

Code 22:
Measuring Fable Wheel Circumference

- This code sequence demonstrates how to measure the circumference of a Fable wheel. We first make a mark on the wheel, then start the code, and the Spin module will move forward for a distance of 34 cm. Upon completion, we observe that the mark has returned to its initial position, indicating that a full revolution has been completed. Therefore, by unrolling the circle, we determine that the circumference of the wheel is 34 cm. This information is valuable when measuring distances traveled.

WATCH VIDEO



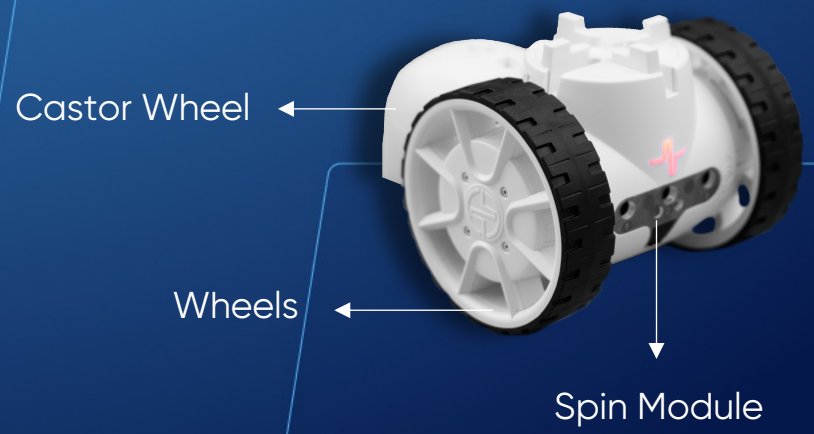

wait until
key pressed?
1
drive
34
cm
with speed:
20
on
1B22
wait until
has reached target on motor(s)
both
on
1B22

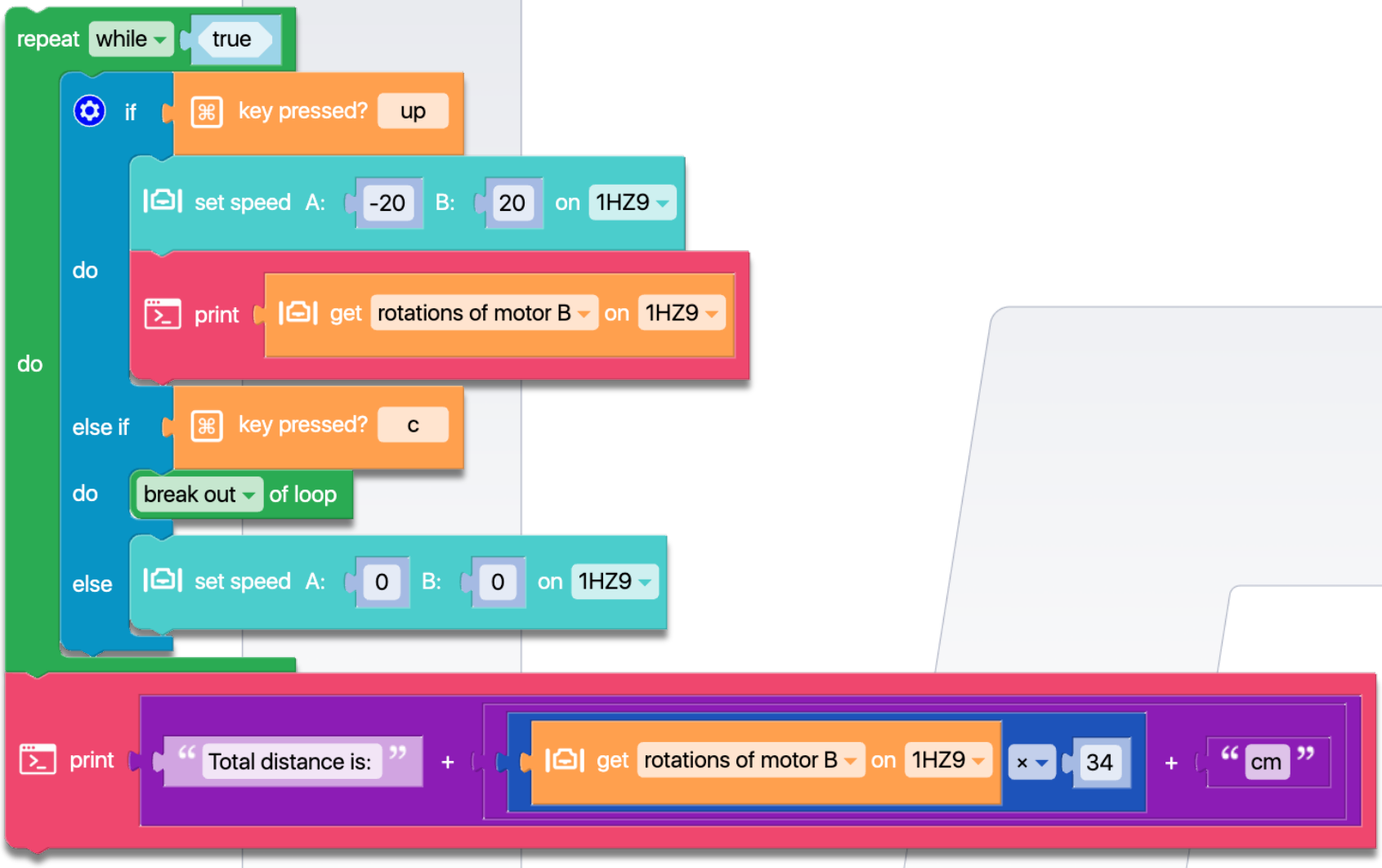
shape robotics
SMART LEARNING

Code 23:
Measure a Distances based on Fable Wheel Circumference

- The program measures the distance traveled by a Spin module, considering the wheel circumference. In our scenario, the wheel has a circumference of 34 cm. We continuously monitor the number of revolutions made by motor B (which runs in the positive direction). Exiting the loop and final calculation occur upon pressing the 'c' key.

WATCH VIDEO

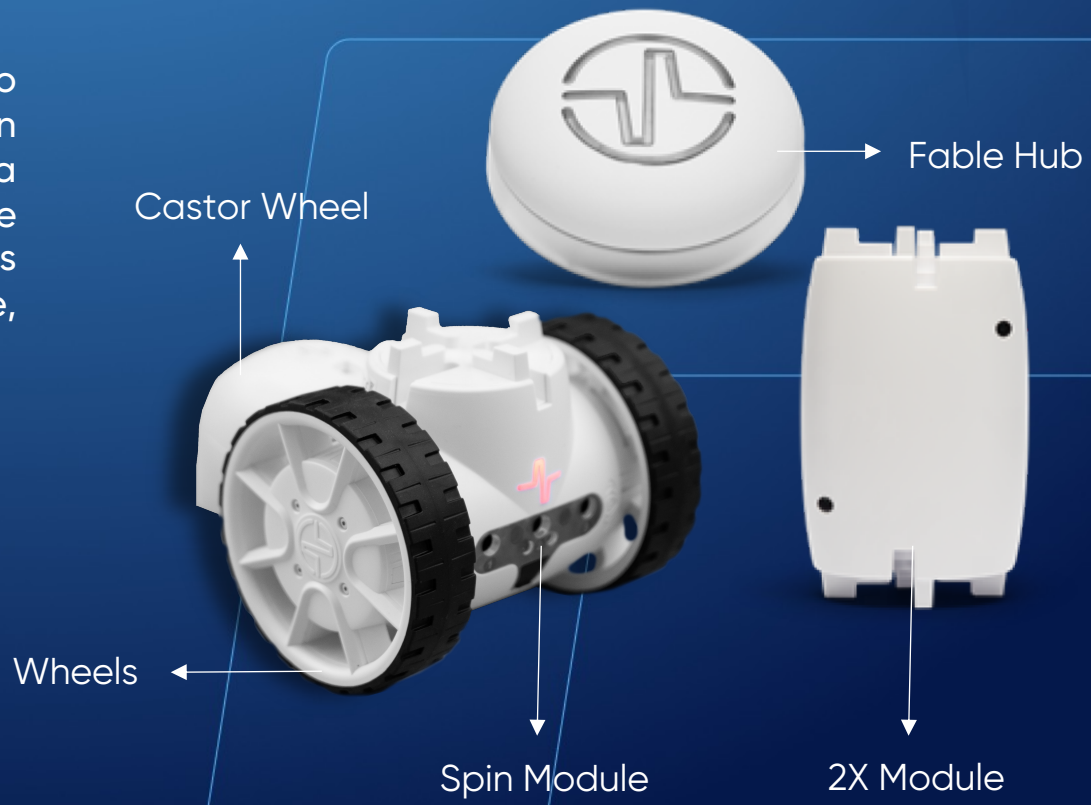


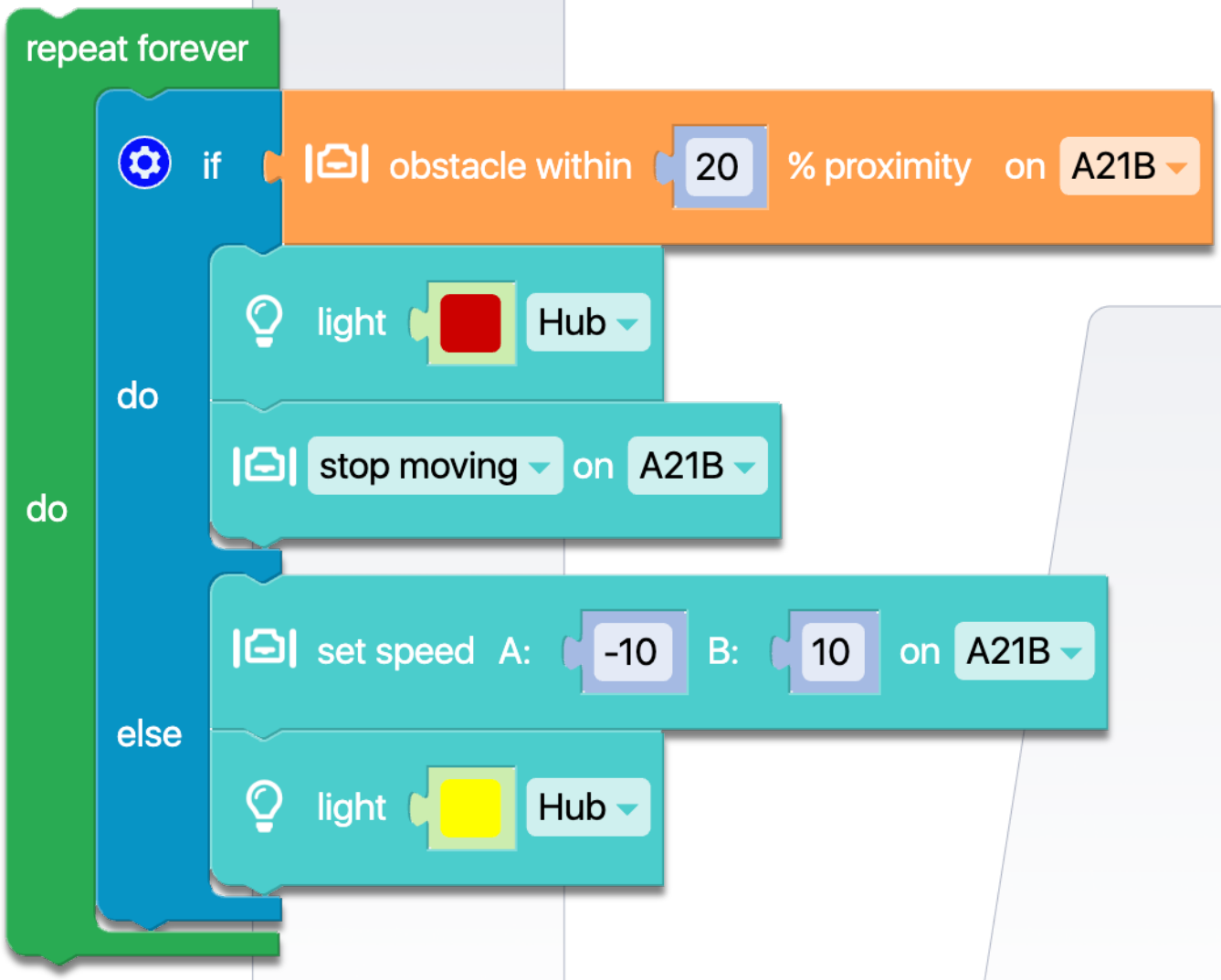

repeat while true
if key pressed? up
set speed A: -20 B: 20 on 1HZ9
do
print get rotations of motor B on 1HZ9
do
else if key pressed? c
do break out of loop
else set speed A: 0 B: 0 on 1HZ9
print "Total distance is:" + get rotations of motor B on 1HZ9 × 34 + "cm"

Code 24:
Detect Obstacle

- The Spin module moves forward only if there are no obstacles in front. At this time, the Hub is illuminated in yellow. The output console continuously displays the data detected by the proximity sensor. When an obstacle appears, the Spin module stops and remains in this state as long as the obstacle remains in front of it. During this time, the Hub is illuminated in red.

WATCH VIDEO

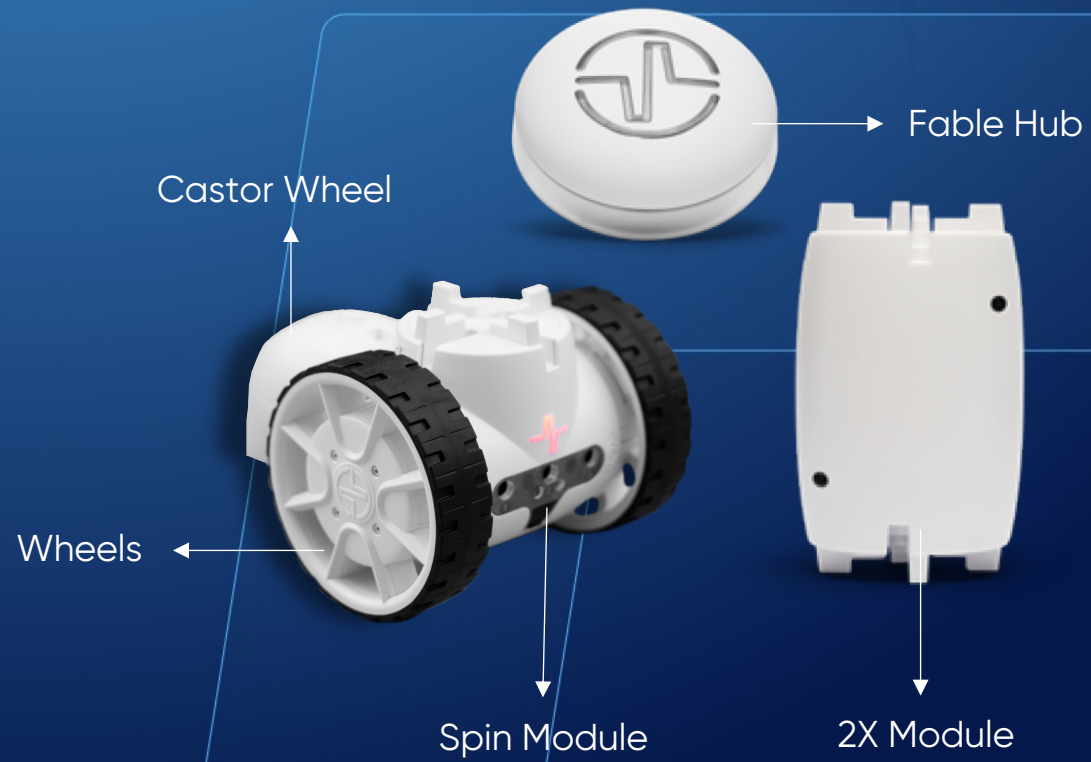


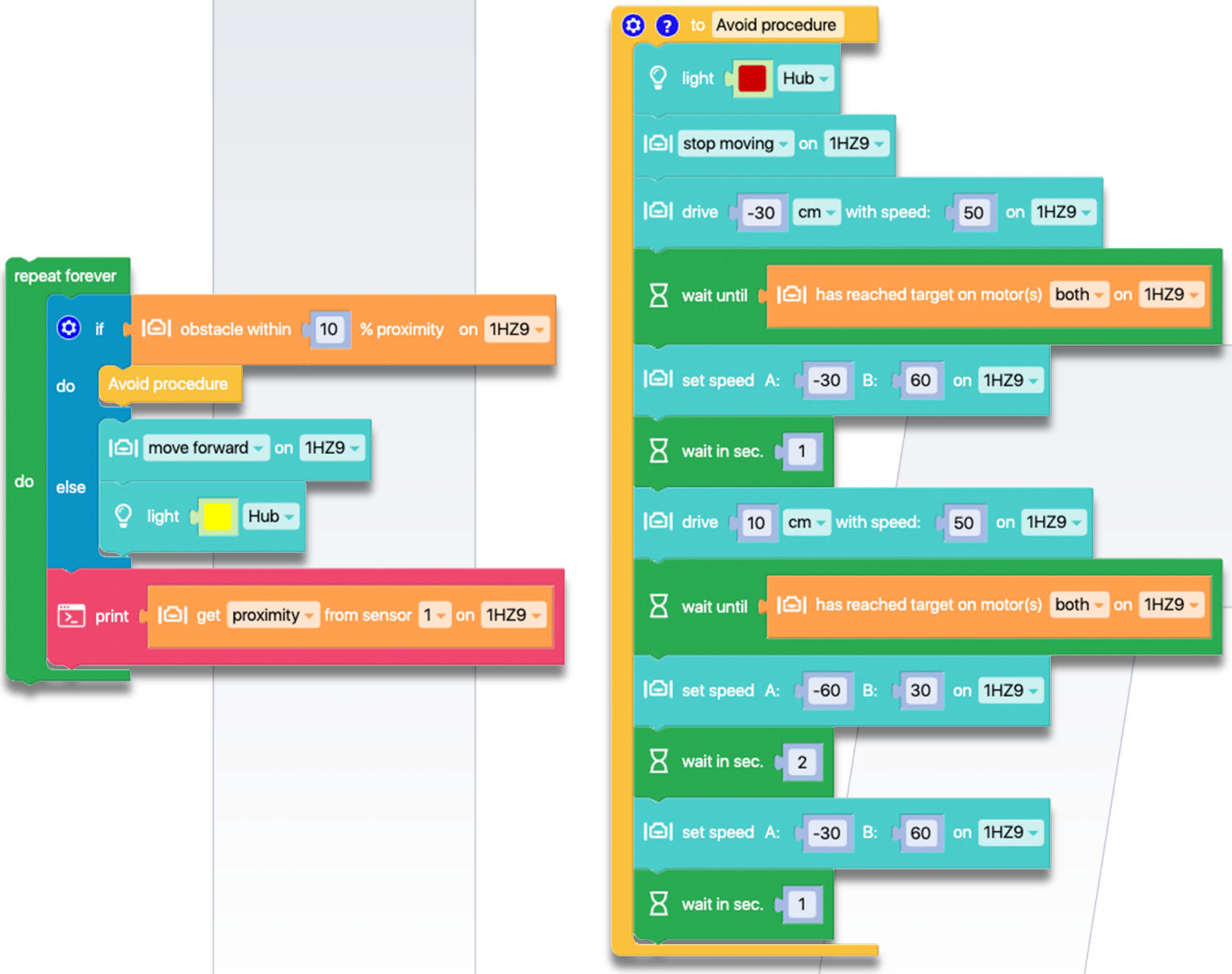

repeat forever
if
obstacle within
20
% proximity
on
A21B
do
light
Hub
stop moving
on
A21B
do
set speed
A:
-10
B:
10
on
A21B
else
light
Hub

Code 25:
Avoid Obstacle – method 1

- The program relies on Code “Detect Obstacle”. The Spin robotic module moves forward unless it encounters an obstacle. When an obstacle is detected, the program triggers the Avoid procedure (named by us) function, which uses basic commands to perform an obstacle avoidance action. The 2X module, which is a Fable accessory, has been considered as an obstacle. Depending on the obstacle you want to avoid, you will need to modify the numerical values in the program.

WATCH VIDEO

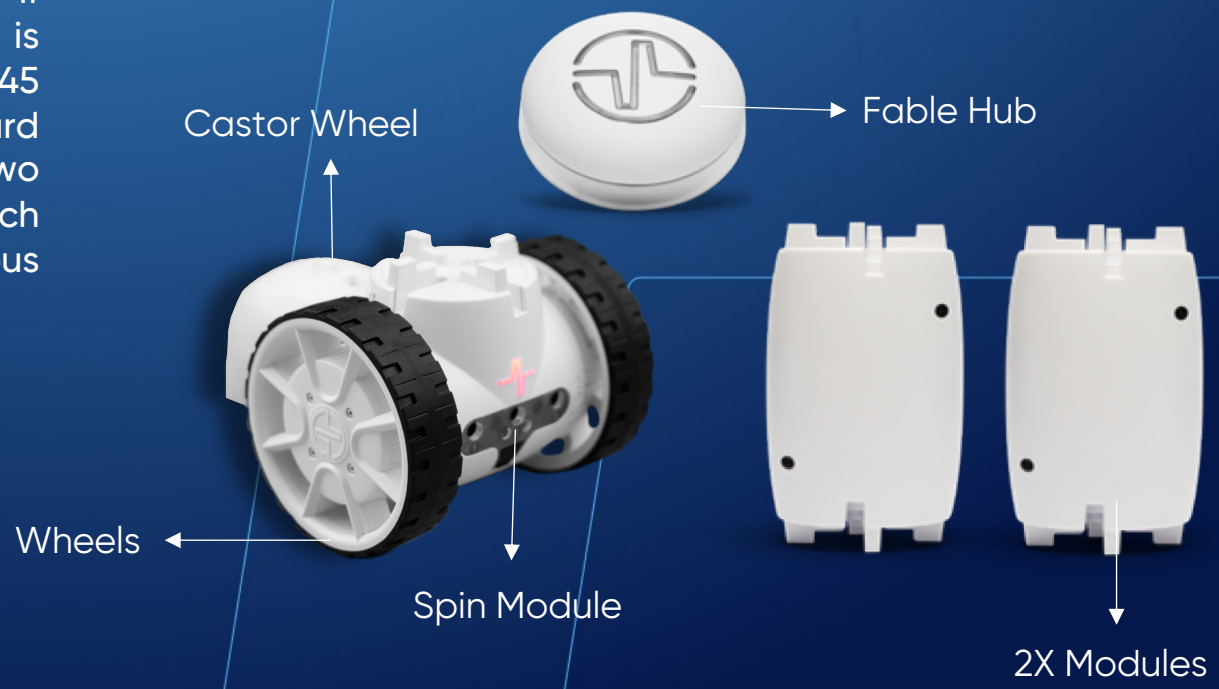


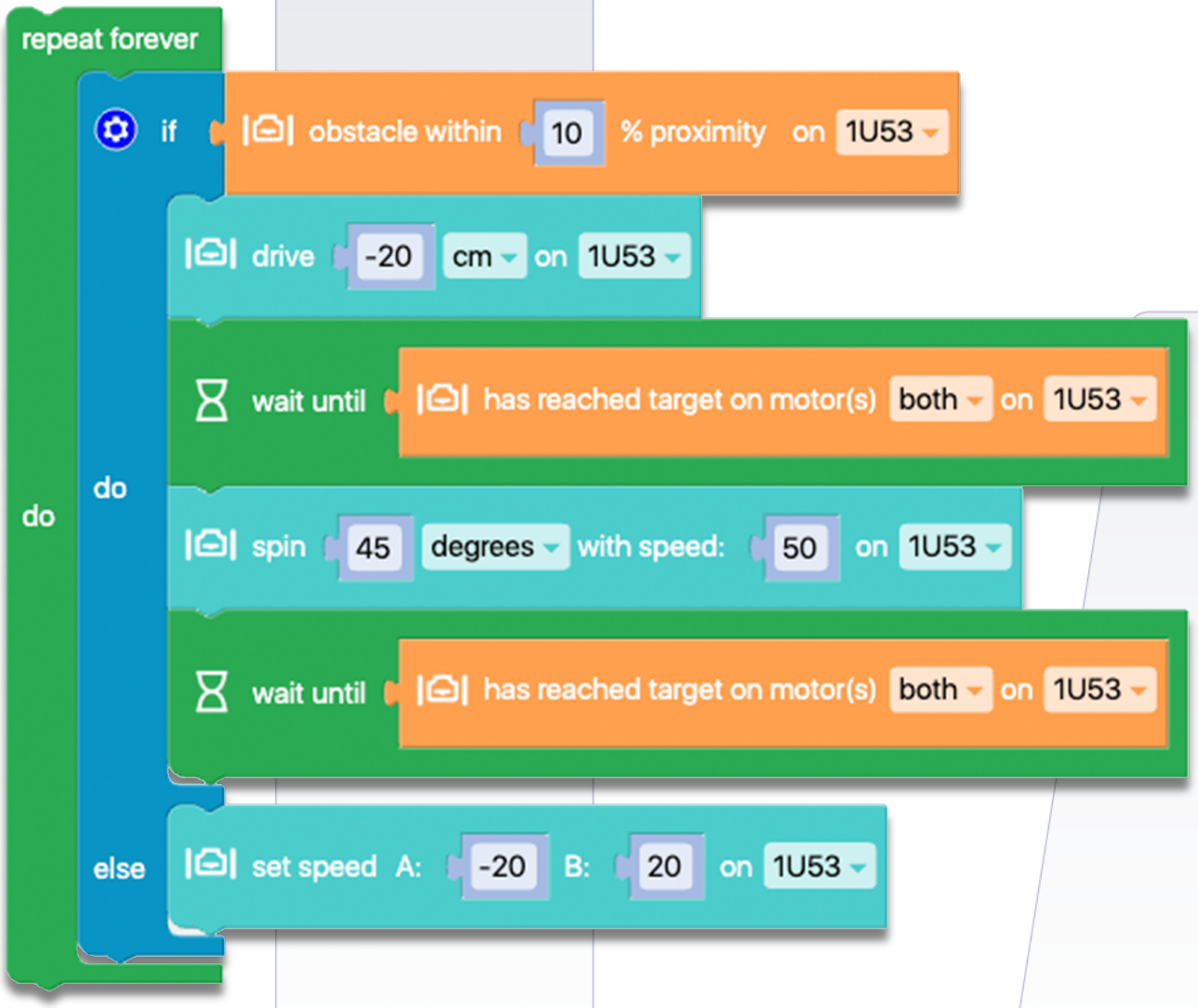

repeat forever
do
if
obstacle within 10 % proximity on 1HZ9
do
Avoid procedure
else
move forward on 1HZ9
light Hub
print get proximity from sensor 1 on 1HZ9
to Avoid procedure
light Hub
stop moving on 1HZ9
drive -30 cm with speed: 50 on 1HZ9
wait until has reached target on motor(s) both on 1HZ9
set speed A: -30 B: 60 on 1HZ9
wait in sec. 1
drive 10 cm with speed: 50 on 1HZ9
wait until has reached target on motor(s) both on 1HZ9
set speed A: -60 B: 30 on 1HZ9
wait in sec. 2
set speed A: -30 B: 60 on 1HZ9
wait in sec. 1

Code 26:
Avoid Obstacle – method 2

- The Spin module is designed to move forward, but only if there is no obstacle in its path. Whenever an obstacle is detected, the module reverses 20cm and rotates 45 degrees to avoid the obstacle before continuing its forward movement. To ensure precise execution of commands, two “Has reach target” motors commands were used, which prevent the next command from executing until the previous one has been completed.

WATCH VIDEO



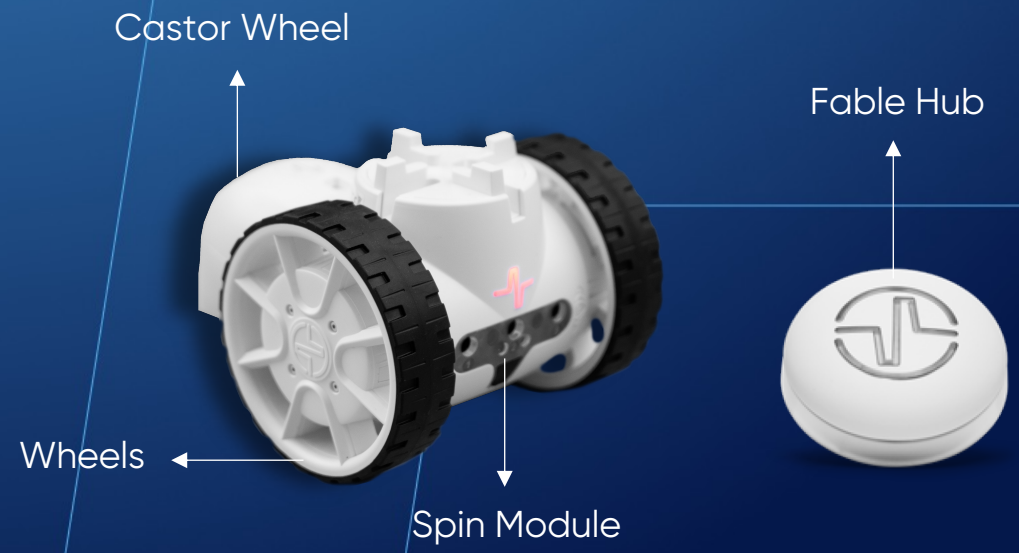

repeat forever
do
if
obstacle within
10
% proximity
on
1U53
drive
-20
cm
on
1U53
wait until
has reached target on motor(s)
both
on
1U53
do
spin
45
degrees
with speed:
50
on
1U53
wait until
has reached target on motor(s)
both
on
1U53
else
set speed
A:
-20
B:
20
on
1U53

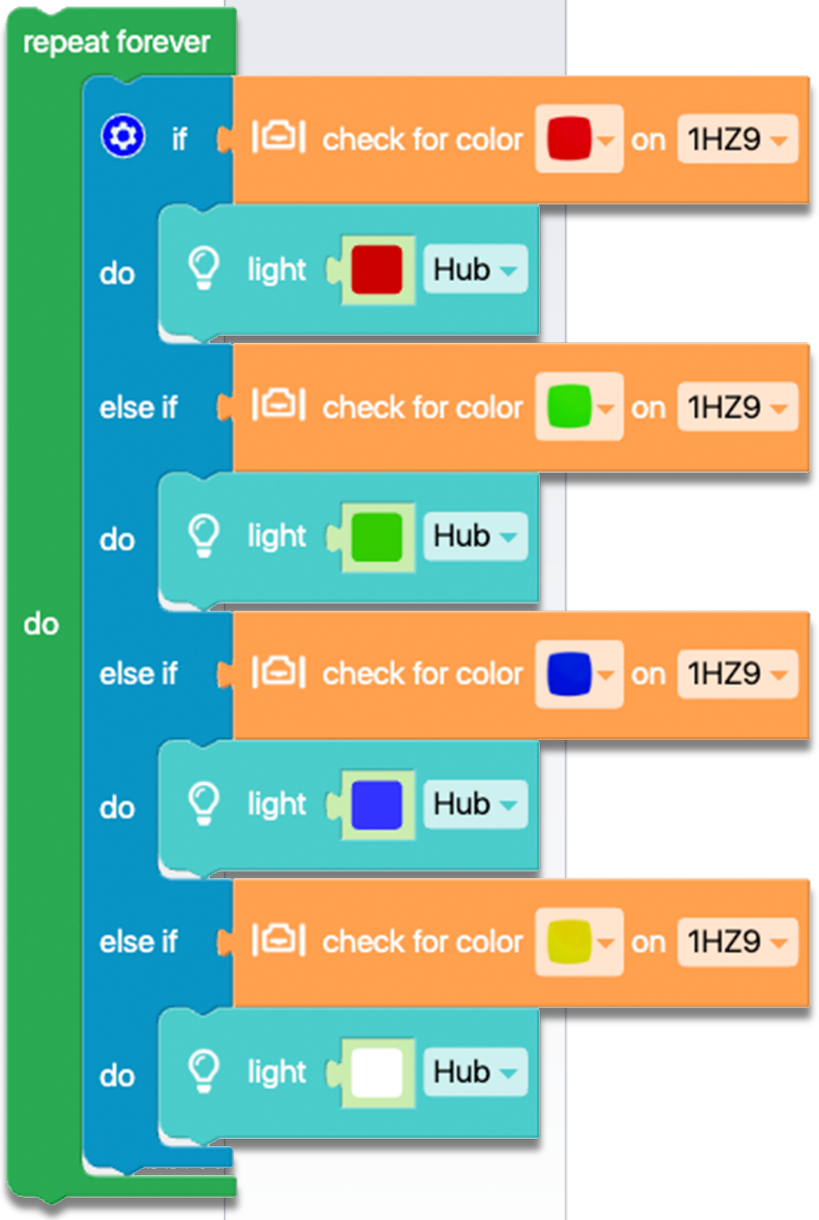
shape robotics
SMART LEARNING

Code 27:
Color Detection

- The program activates the color sensors by placing the Spin robotic module into detection mode. If the robot detects a red object in front of it, the program will command the Hub to display the red color.
- The same procedure is followed for detecting green, blue, and yellow objects. If none of these colors are detected in front of the robot, the Hub will be illuminated in white.

WATCH VIDEO





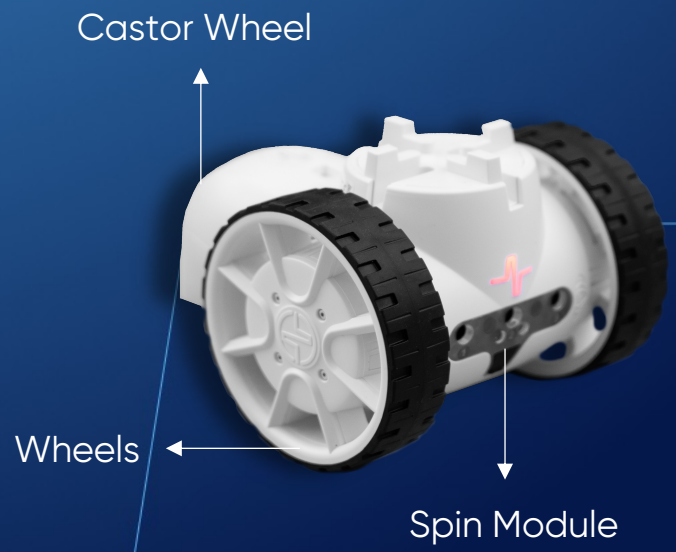
shape robotics
SMART LEARNING

shape robotics
SMART LEARNING

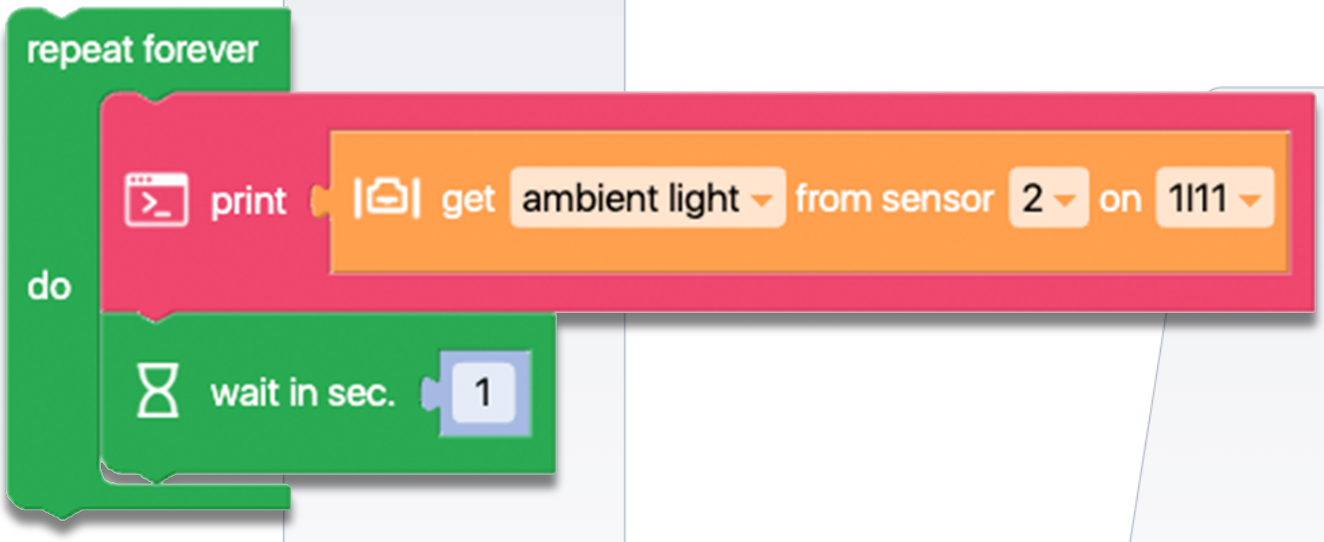
Code 28:
Measuring ambiental light

- The program utilizes sensor number 2 to measure ambient light levels. Once per second, the output console displays the value read by the sensor. It is also possible to use the other two sensors.

WATCH VIDEO



shape robotics
SMART LEARNING

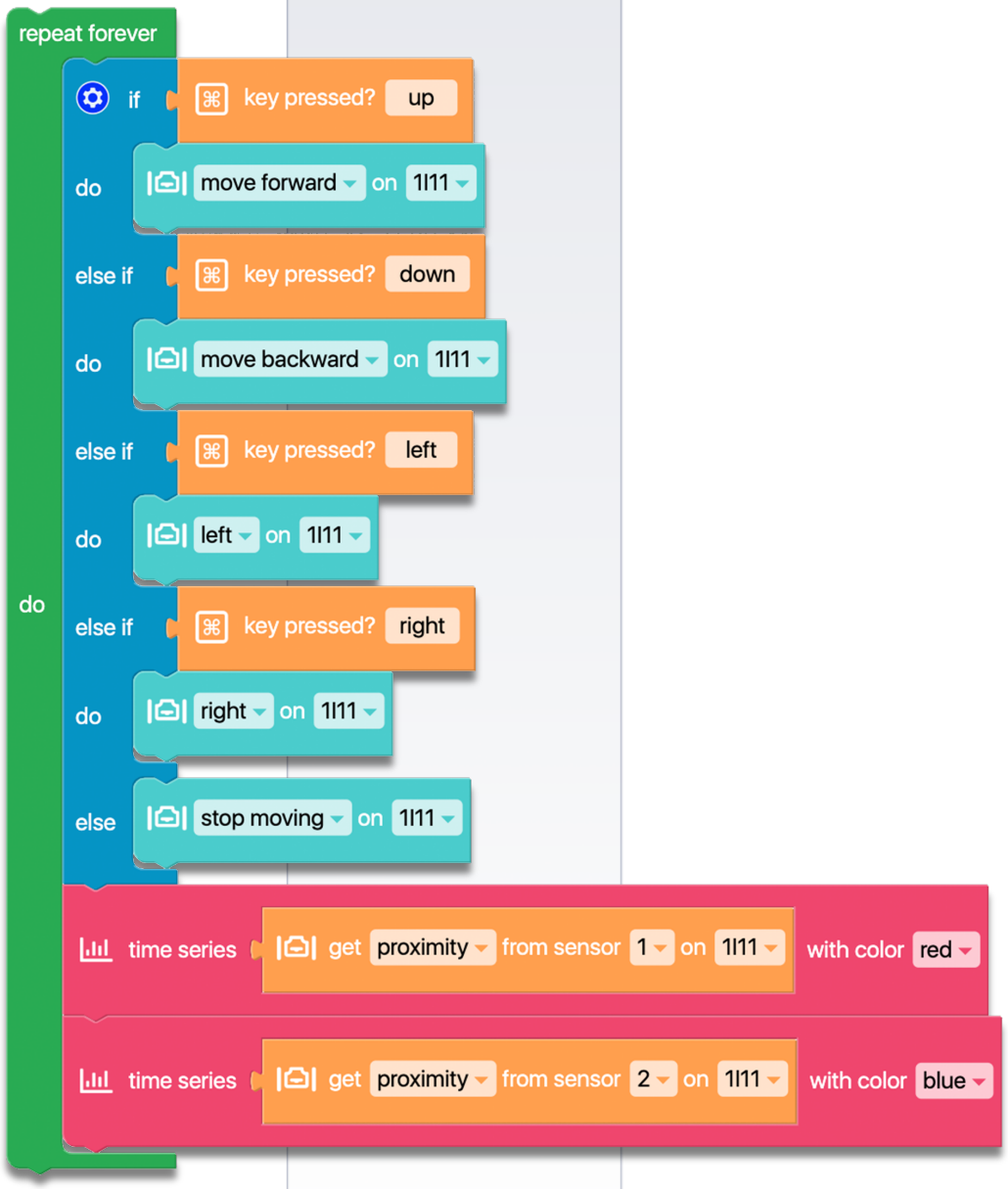


shape robotics
SMART LEARNING

Code 29:
Measuring Distances in Proximity

- The program utilizes “Spin Key Control”. Two new commands have been added that enable the display of a time-dependent graph depicting the proximity percentage detected by the first and second sensors on the robot. The graph is created in two colors to enable real-time observation and post-program analysis of the detection.

WATCH VIDEO

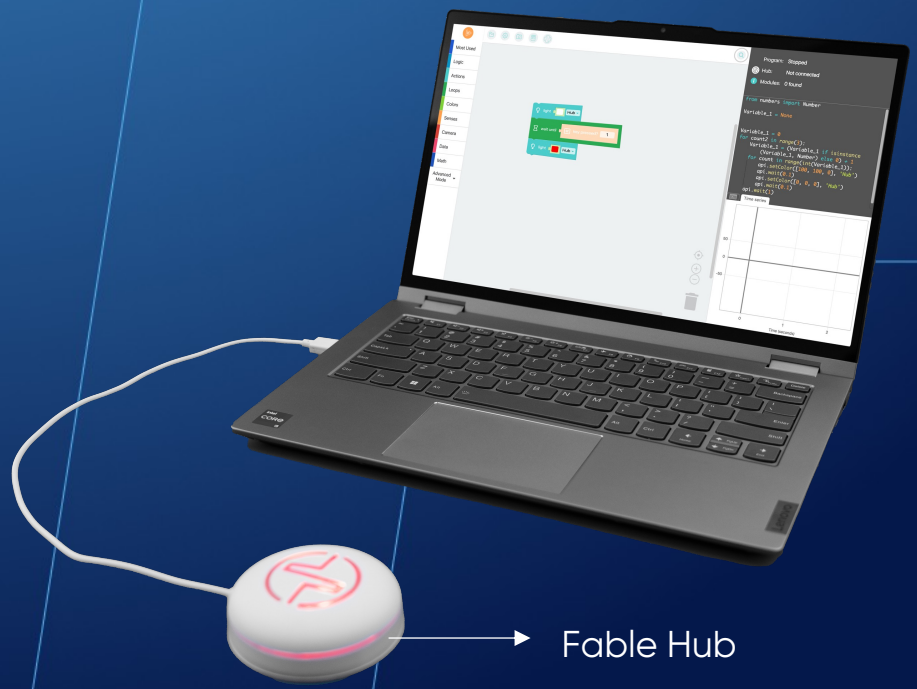

repeat forever
if
key pressed? up
do
move forward on 1I11
else if
key pressed? down
do
move backward on 1I11
else if
key pressed? left
do
left on 1I11
do
else if
key pressed? right
do
right on 1I11
else
stop moving on 1I11
time series
get proximity from sensor 1 on 1I11
with color red
time series
get proximity from sensor 2 on 1I11
with color blue

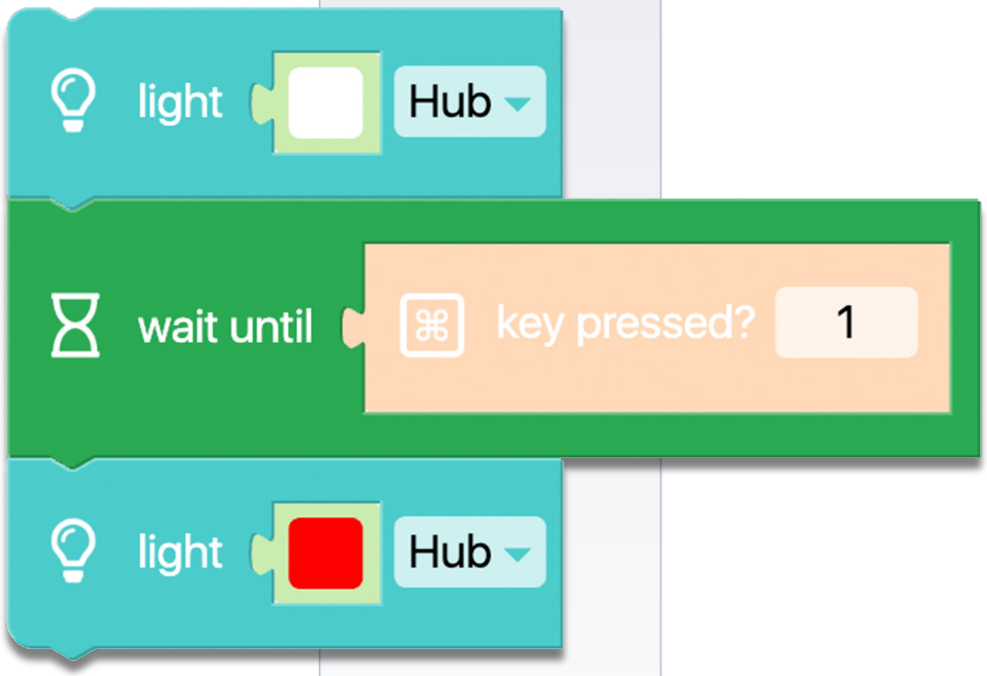
shape robotics
SMART LEARNING

Code 30:
Wait for Keys Press

- The program starts the Hub on white light and remains in this state until key 1 is pressed. Any other key does not affect the color, but after pressing key 1, the Hub will change color to red.

WATCH VIDEO



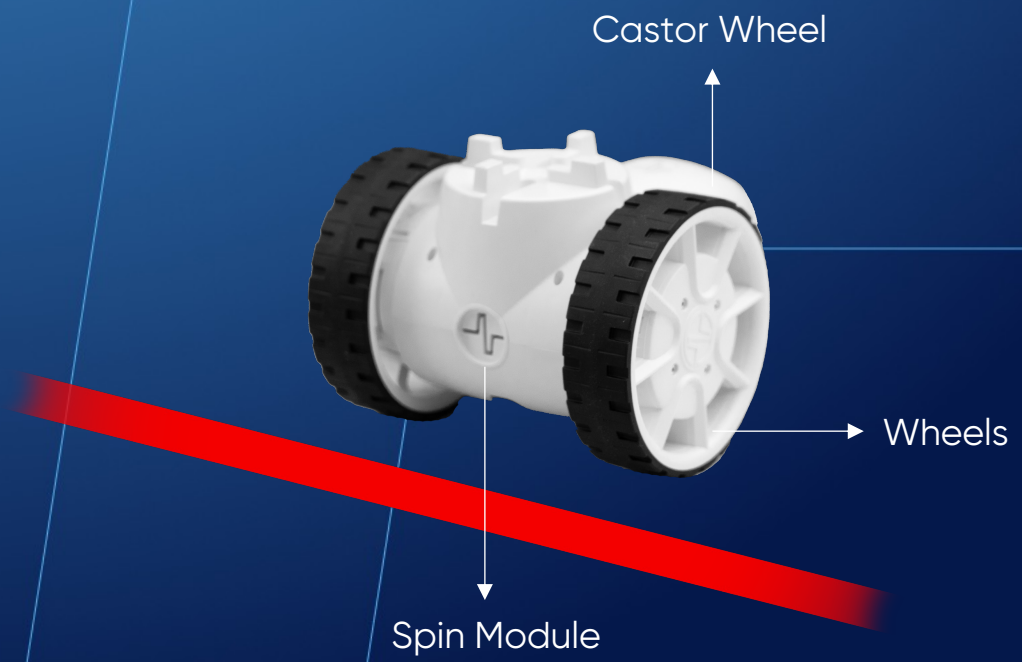

light
Hub
wait until
key pressed?
1
light
Hub

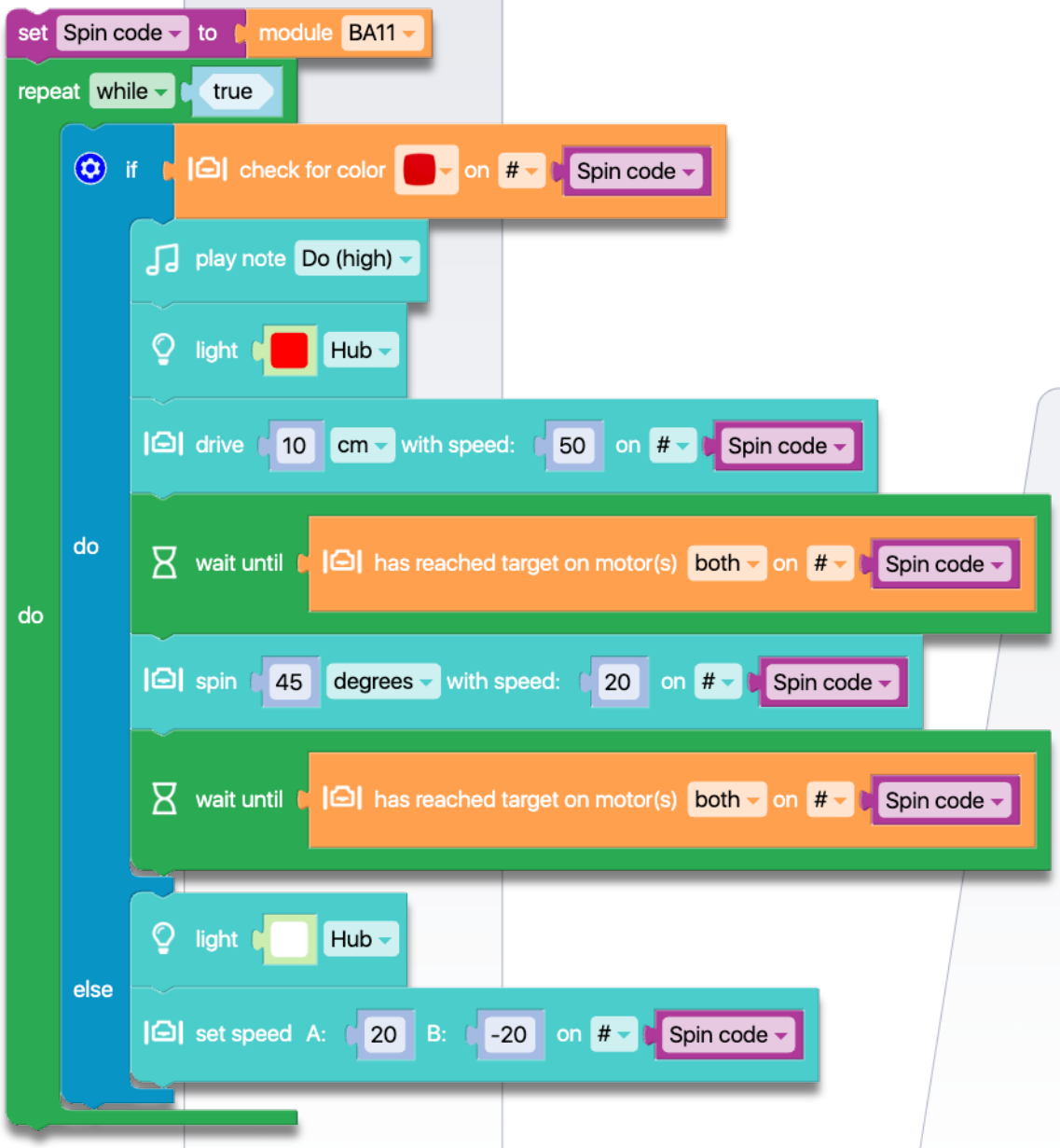
shape robotics
SMART LEARNING

Code 31:
Color Line Detect

- The Spin moves ahead at a steady pace until it comes across a red line underneath it. When the red color is detected, the program sounds a beep, lights up the Hub in red, and initiates an avoidance procedure by reversing and turning at a particular angle before resuming its forward motion. We have the flexibility to adjust the values to achieve different speeds, identify different colors, and execute other avoidance maneuvers.

WATCH VIDEO



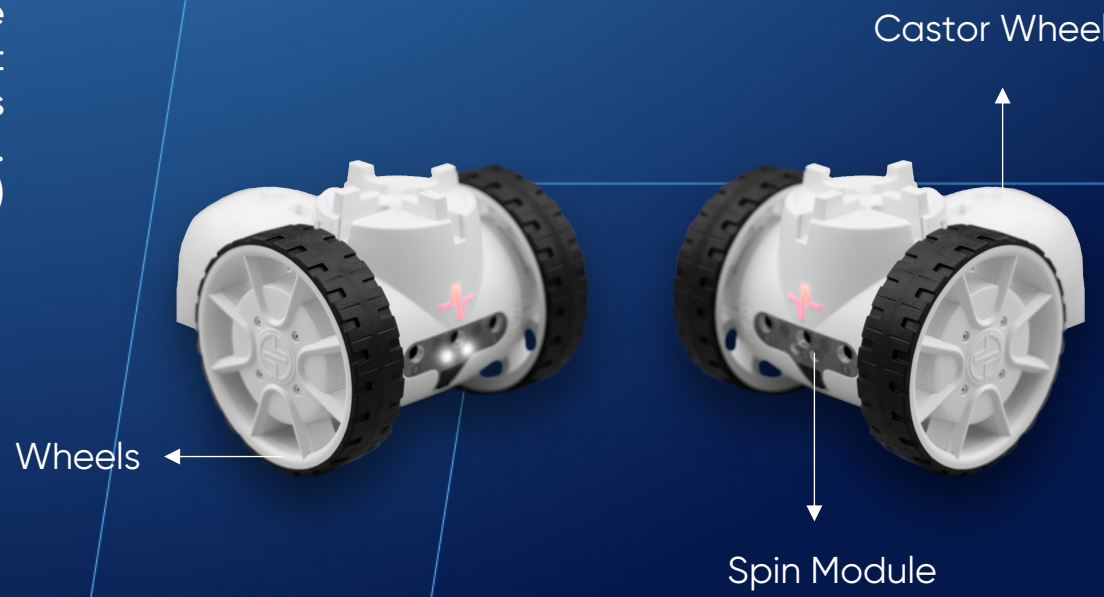

set Spin code to module BA11
repeat while true
do
if check for color on # Spin code
play note Do (high)
light Hub
drive 10 cm with speed: 50 on # Spin code
do
wait until has reached target on motor(s) both on # Spin code
spin 45 degrees with speed: 20 on # Spin code
wait until has reached target on motor(s) both on # Spin code
light Hub
else
set speed A: 20 B: -20 on # Spin code

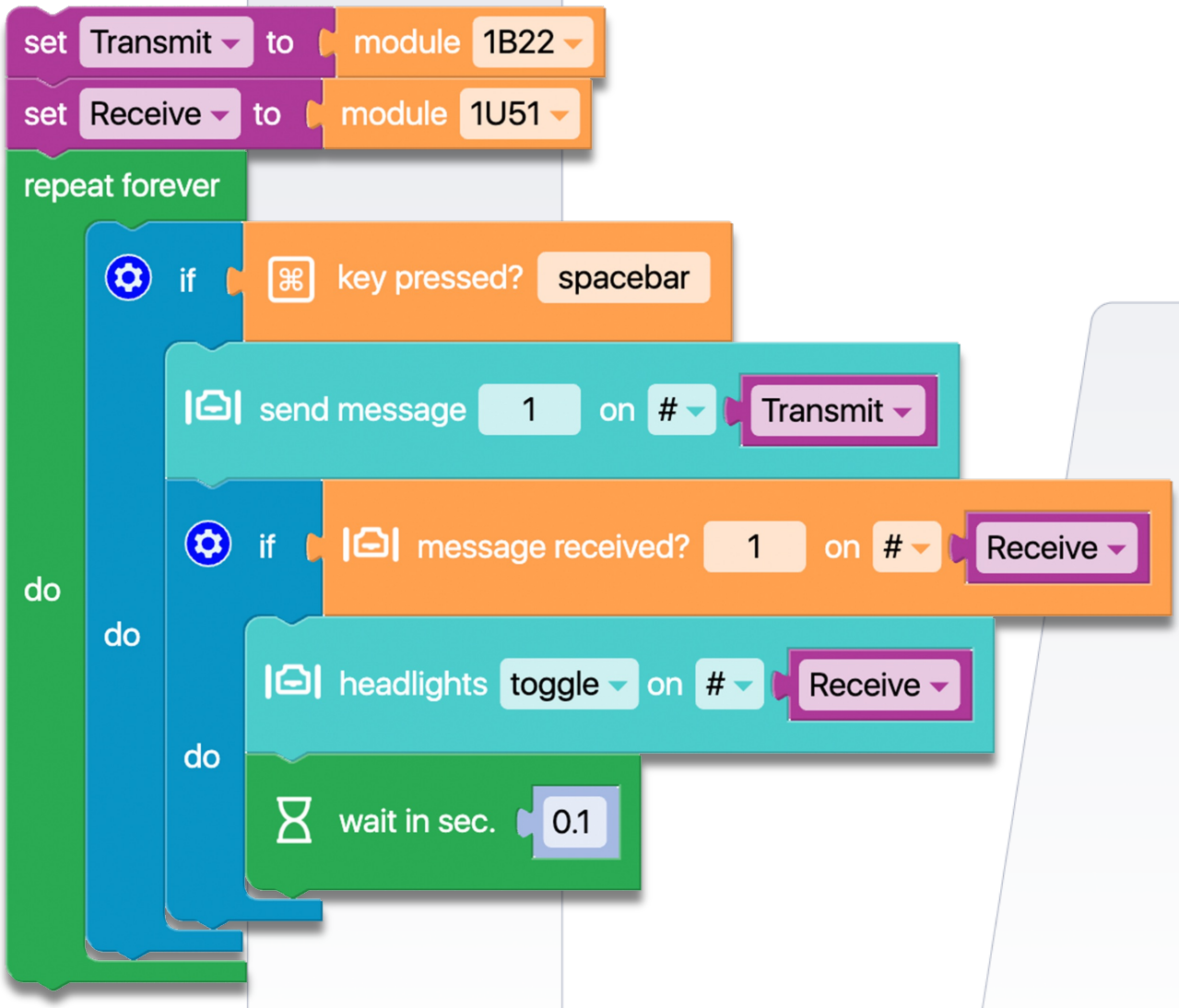
shape robotics
SMART LEARNING

Code 32:
Sends/Receive messages with Spin

- The Spin modules are capable of transmitting and receiving IR messages. In this program, the transmitter module detects key presses, specifically the space key, and when activated, sends a message (key 1) to the receiver Spin module.
- The Receiver module, upon receiving the message, will change the state of the lights from off to on or vice versa. It is important to correctly identify which module is transmitting and which is receiving, and to use the appropriate identification codes. Both the detecting key and the transmitted element (key 1) can be modified to suit your needs.
- Spin modules must be aligned face to face.

WATCH VIDEO



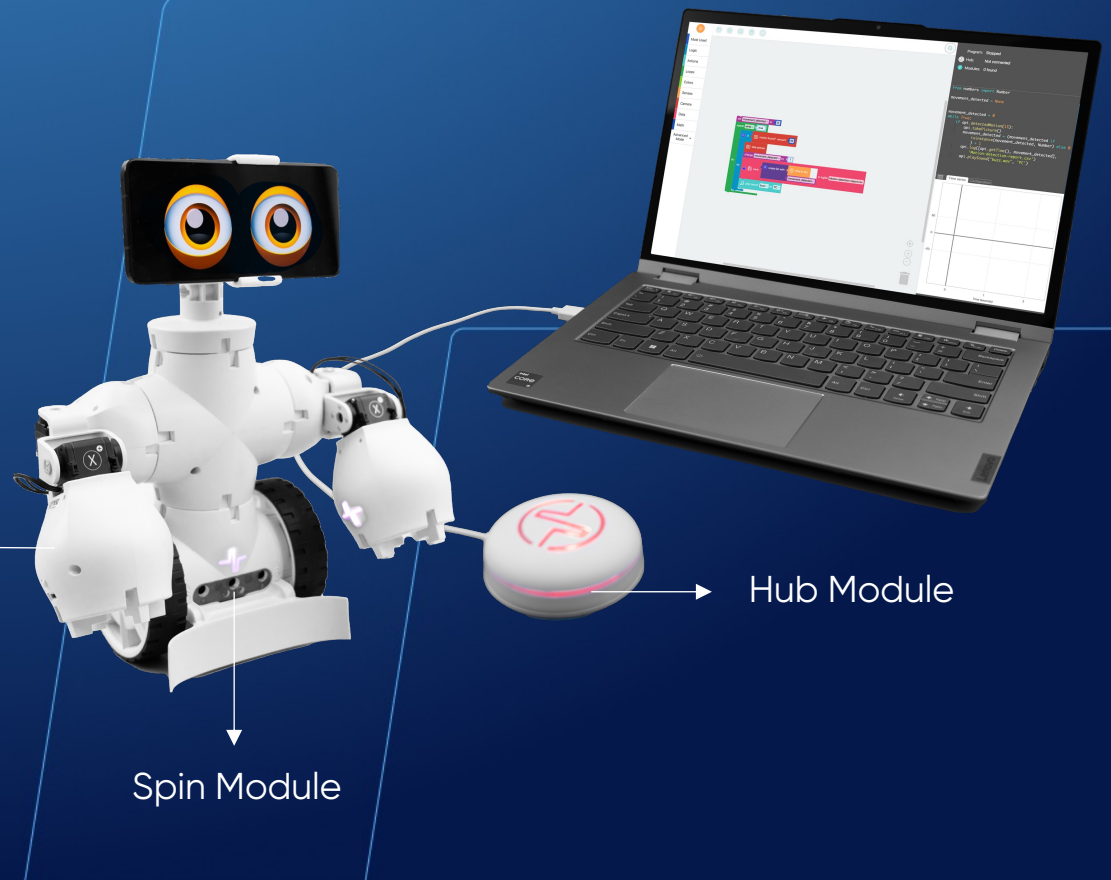

set Transmit to module 1B22
set Receive to module 1U51
repeat forever
do
if key pressed? spacebar
do
send message 1 on # Transmit
if message received? 1 on # Receive
do
headlights toggle on # Receive
wait in sec. 0.1

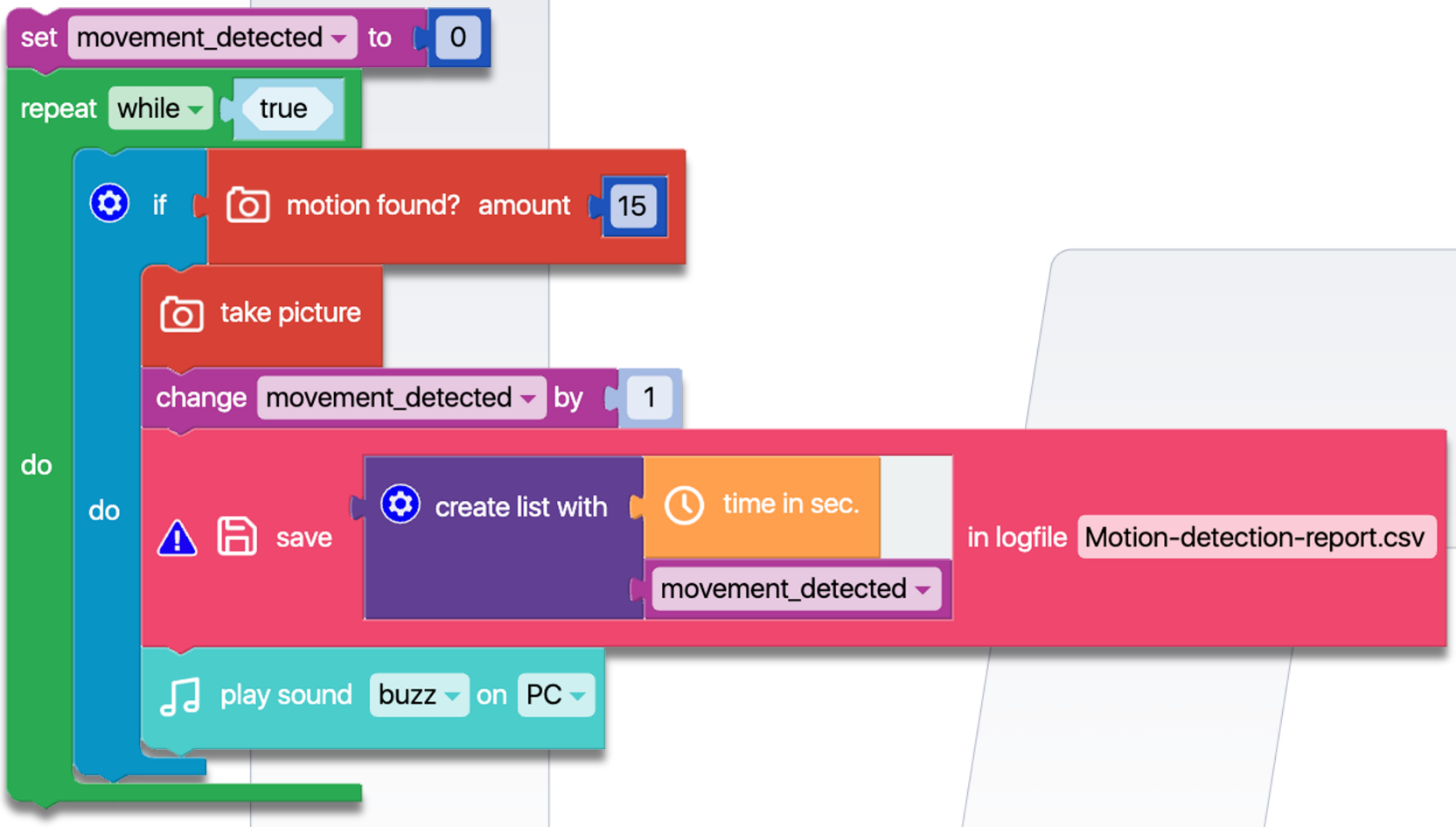
shape robotics
SMART LEARNING

Code 33:
WebCam reacting at Motion

- The program utilises the built-in camera of the computer on which Fable Blockly it is installed. It continuously analyses the images captured by the camera and triggers an audible alarm when a movement above a certain threshold (set at 15) is detected.
- Additionally, the program creates a list with two markers that store the time and the number of detections made up to that moment (which increments by 1 with each detection).

WATCH VIDEO



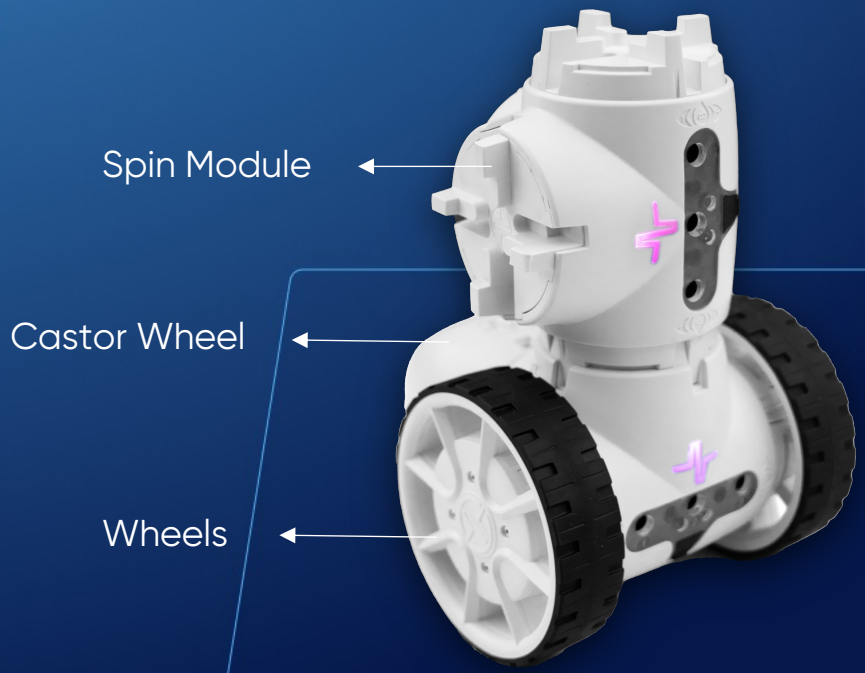

set movement_detected to 0
repeat while true
do
if motion found? amount 15
take picture
change movement_detected by 1
do
save create list with time in sec. movement_detected in logfile Motion-detection-report.csv
play sound buzz on PC

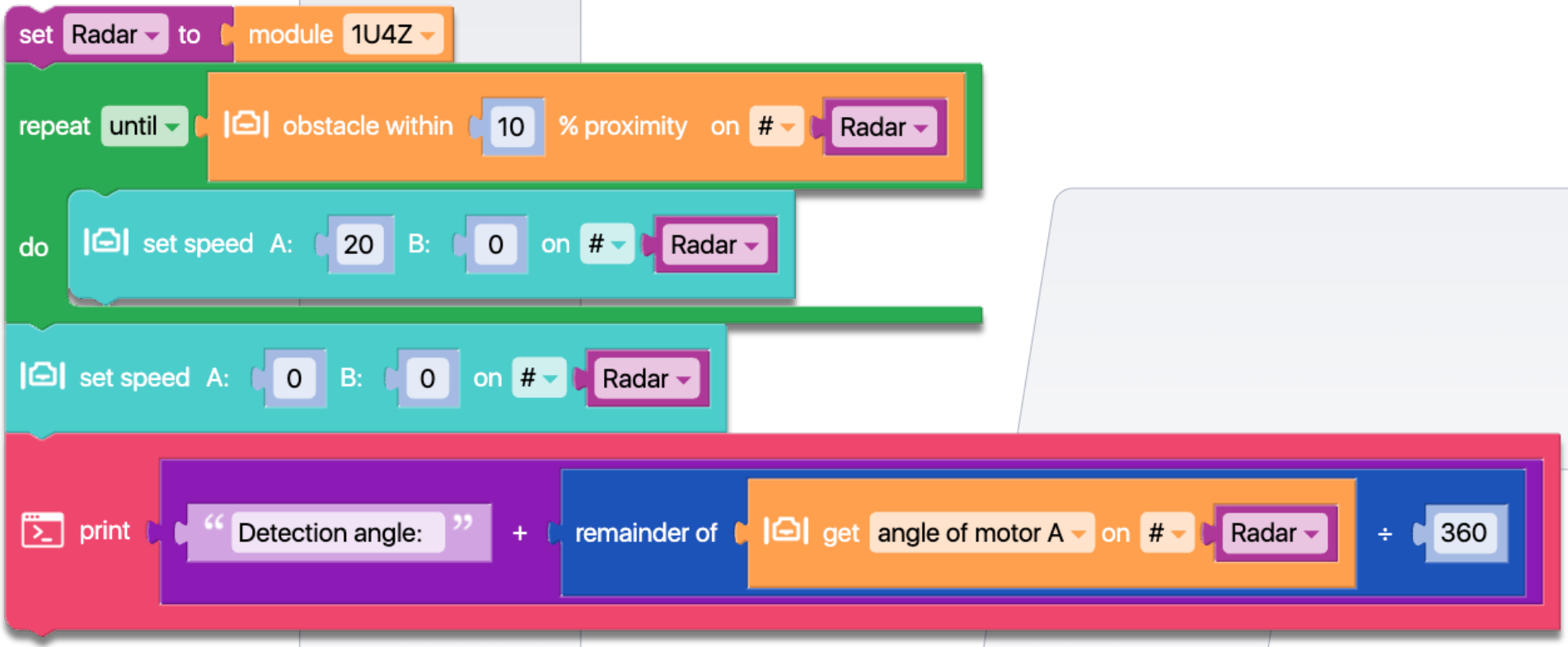
shape robotics
SMART LEARNING

Code 34:
Radar functioning model for Spin

- The program is crafted to simulate the rotary motion of a radar system, denoted by the variable "Radar." The spinner rotates continuously while the proximity sensor remains active. Upon detecting an object nearby, the radar system triggers a visual and auditory alert, simultaneously printing the precise angle of the detected object on the Output Console.

WATCH VIDEO



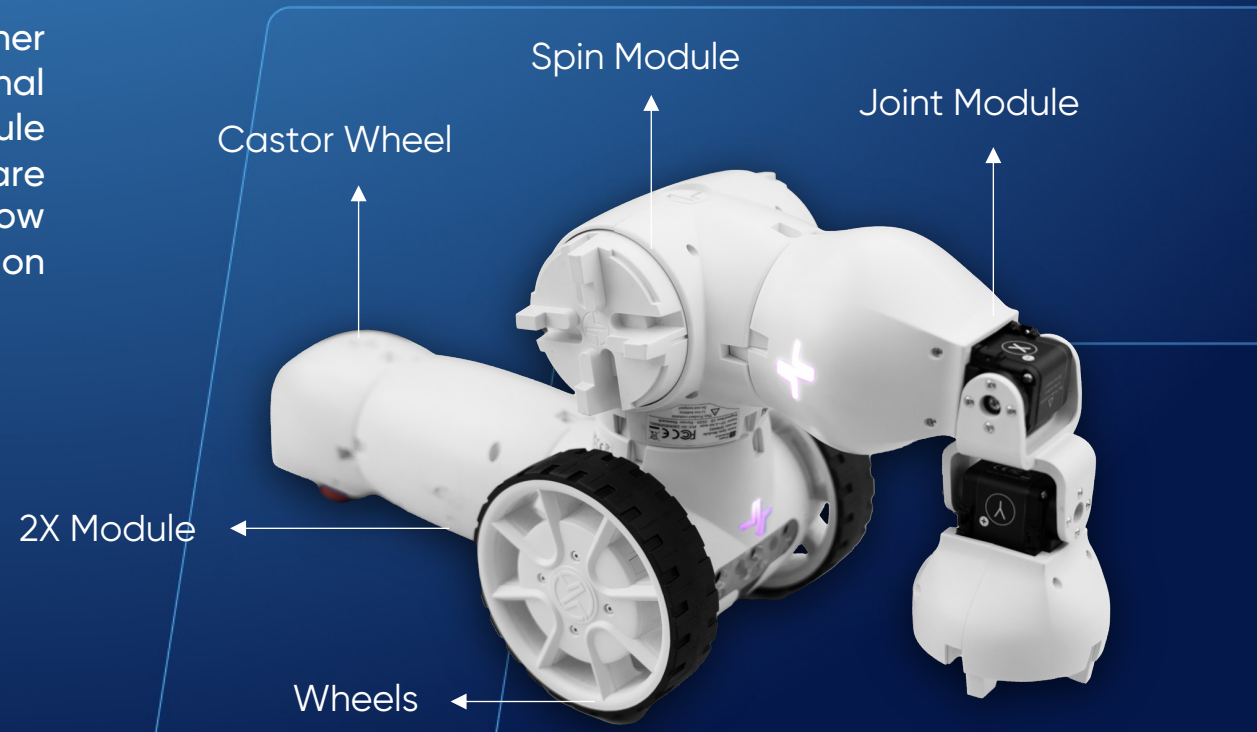

set Radar to module 1U4Z
repeat until obstacle within 10 % proximity on # Radar
do set speed A: 20 B: 0 on # Radar
set speed A: 0 B: 0 on # Radar
print " Detection angle: " + remainder of get angle of motor A on # Radar ÷ 360

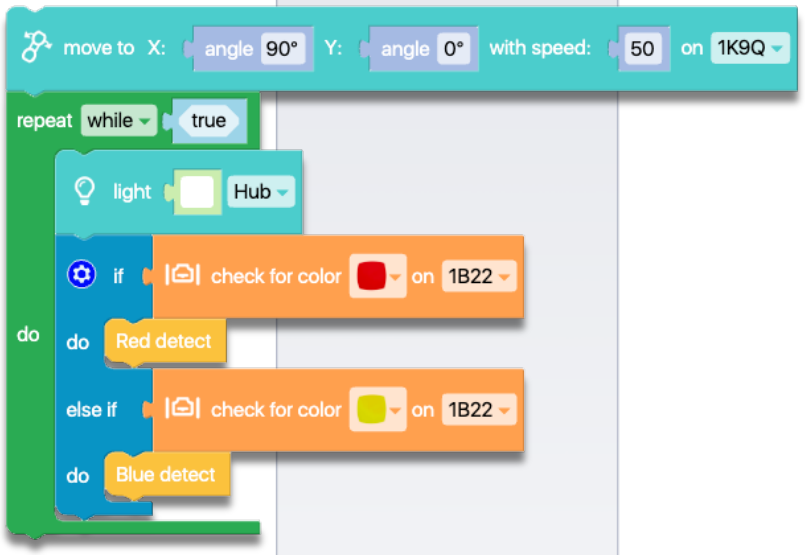
shape robotics
SMART LEARNING

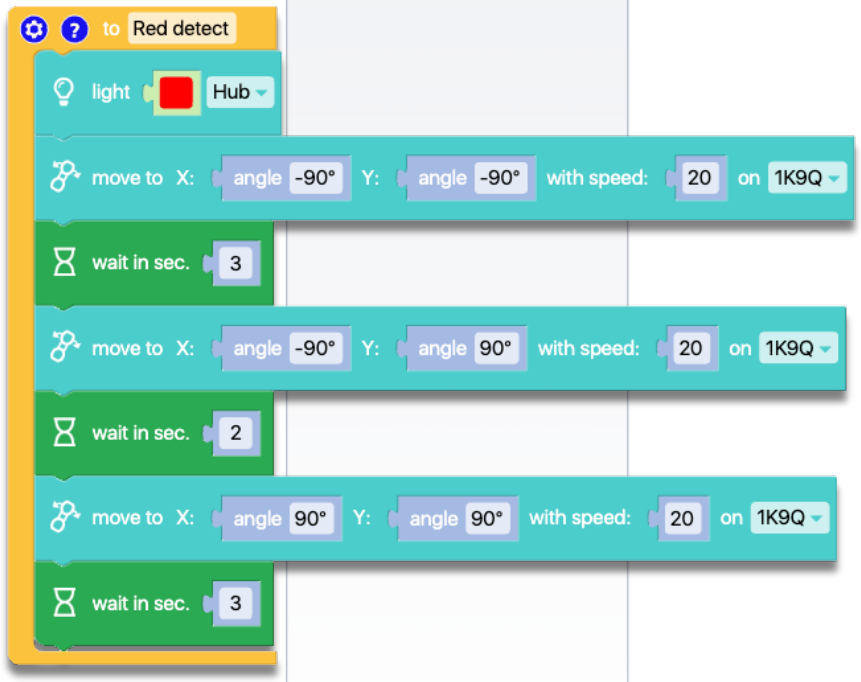
Code 35:
Color Sorting Machine Method 1

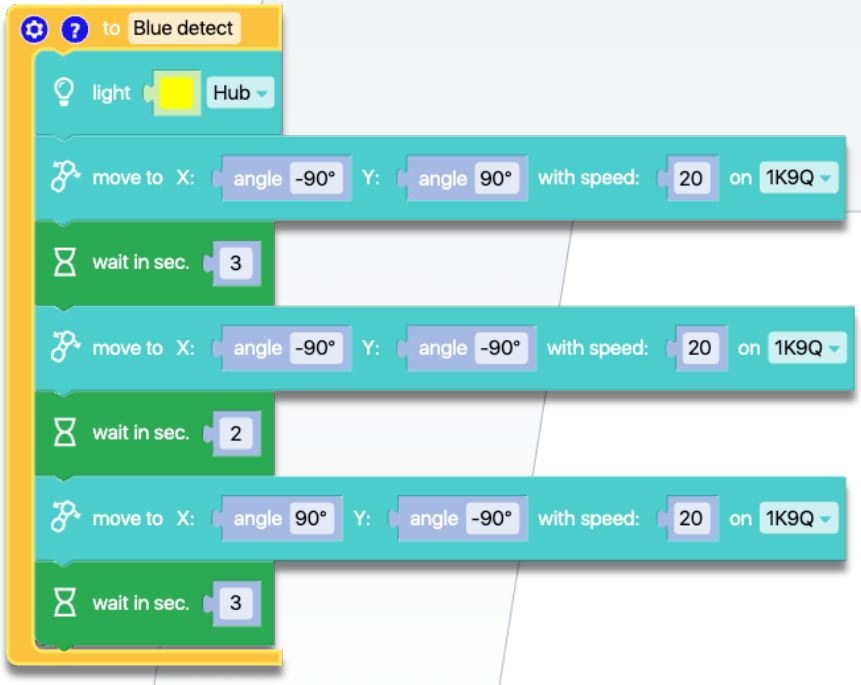
- The program employs a Spin module for color detection, another Spin module to assist in Joint mode orientation, and additional accessories for balance maintenance. The first Spin module identifies colors and initiates a sorting procedure: objects are directed to the right if red is detected and to the left if yellow is detected. Furthermore, the hub adjusts its light based on the detected color.

WATCH VIDEO




move to X: angle 90° Y: angle 0° with speed: 50 on 1K9Q
repeat while true
light Hub
if check for color on 1B22
do Red detect
else if check for color on 1B22
do Blue detect
do


to Red detect
light Hub
move to X: angle -90° Y: angle -90° with speed: 20 on 1K9Q
wait in sec. 3
move to X: angle -90° Y: angle 90° with speed: 20 on 1K9Q
wait in sec. 2
move to X: angle 90° Y: angle 90° with speed: 20 on 1K9Q
wait in sec. 3

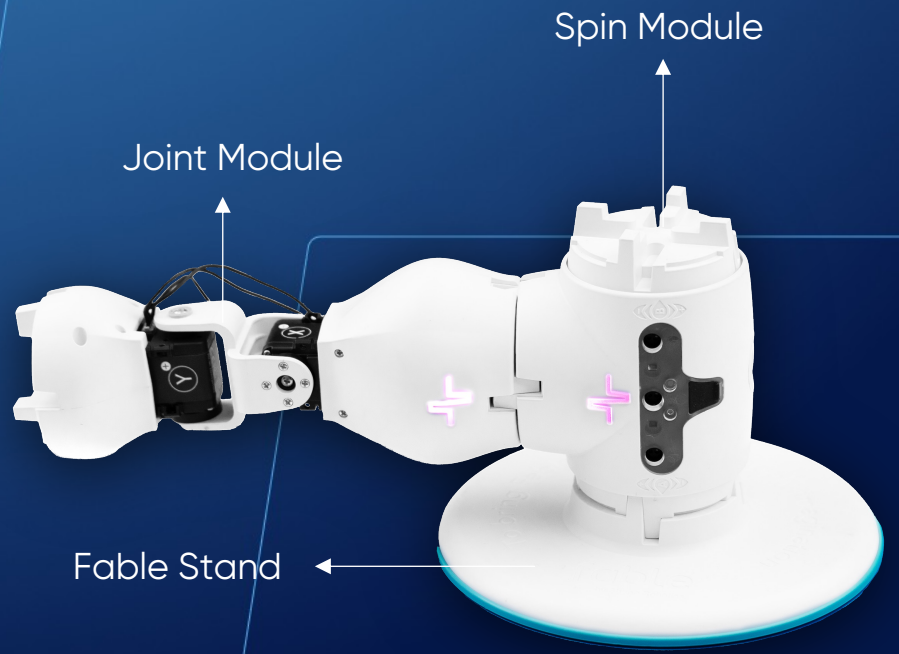

to Blue detect
light Hub
move to X: angle -90° Y: angle 90° with speed: 20 on 1K9Q
wait in sec. 3
move to X: angle -90° Y: angle -90° with speed: 20 on 1K9Q
wait in sec. 2
move to X: angle 90° Y: angle -90° with speed: 20 on 1K9Q
wait in sec. 3

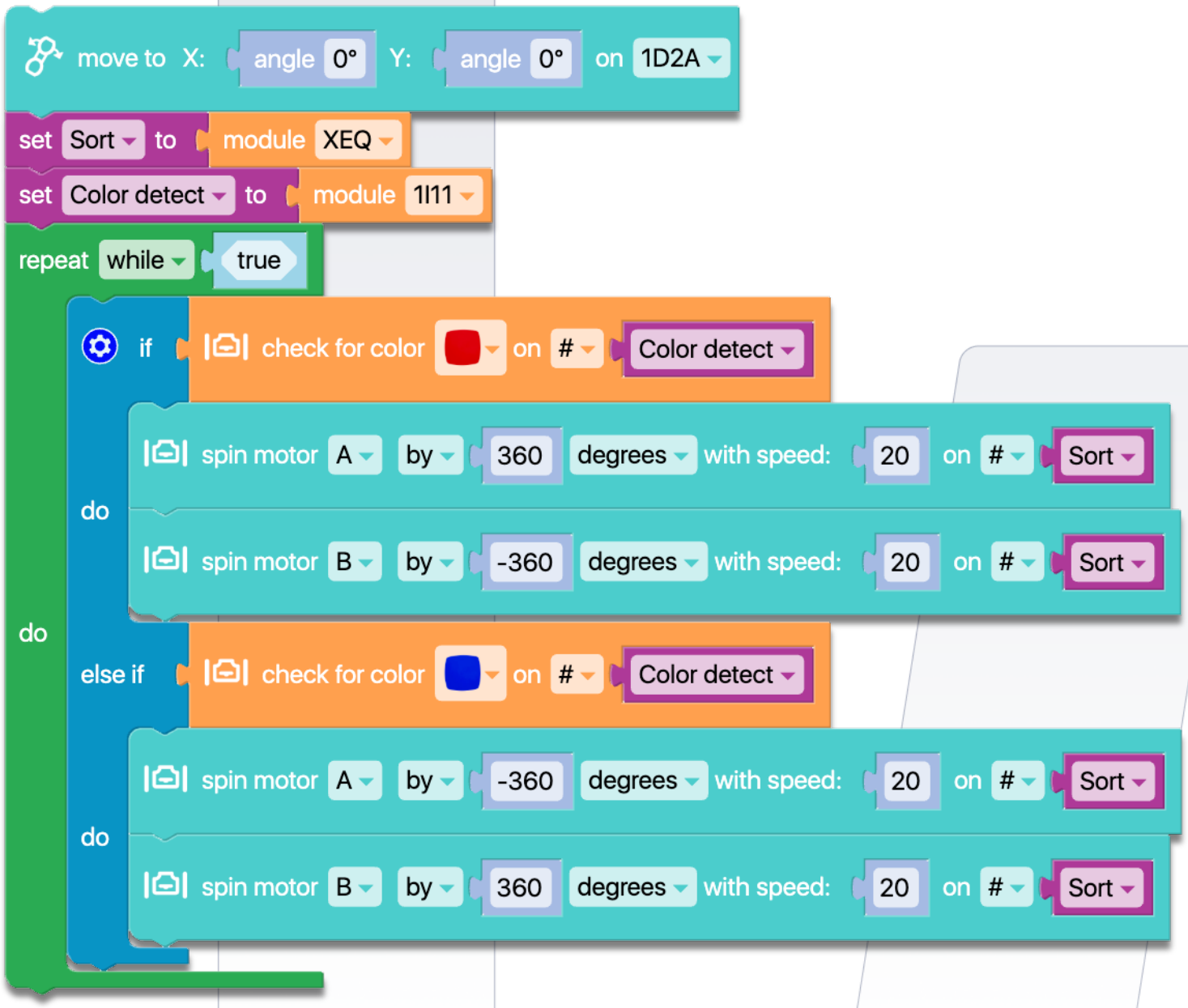
shape robotics
SMART LEARNING

Code 36:
Color Sorting Machine Method 2

- The program is designed to sort objects based on two colors: red and blue. This is accomplished by employing a longer arm positioned at a height of approximately 4cm. Object detection and rotation are performed with the Spin module.

WATCH VIDEO



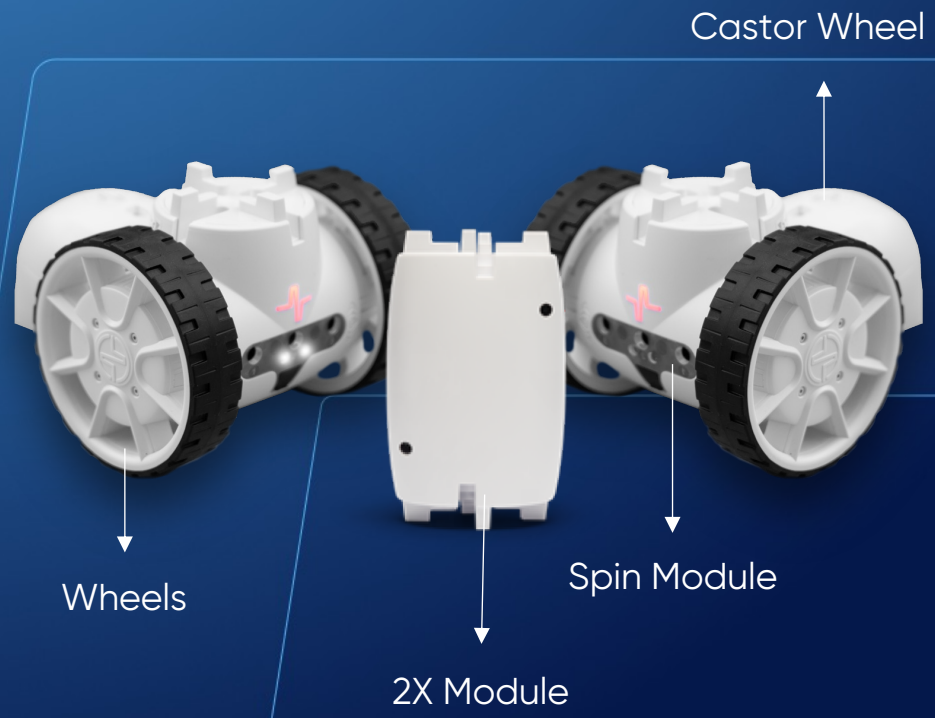

move to X: angle 0° Y: angle 0° on 1D2A
set Sort to module XEQ
set Color detect to module 1I11
repeat while true
if check for color on # Color detect
do
spin motor A by 360 degrees with speed: 20 on # Sort
spin motor B by -360 degrees with speed: 20 on # Sort
else if check for color on # Color detect
do
spin motor A by -360 degrees with speed: 20 on # Sort
spin motor B by 360 degrees with speed: 20 on # Sort

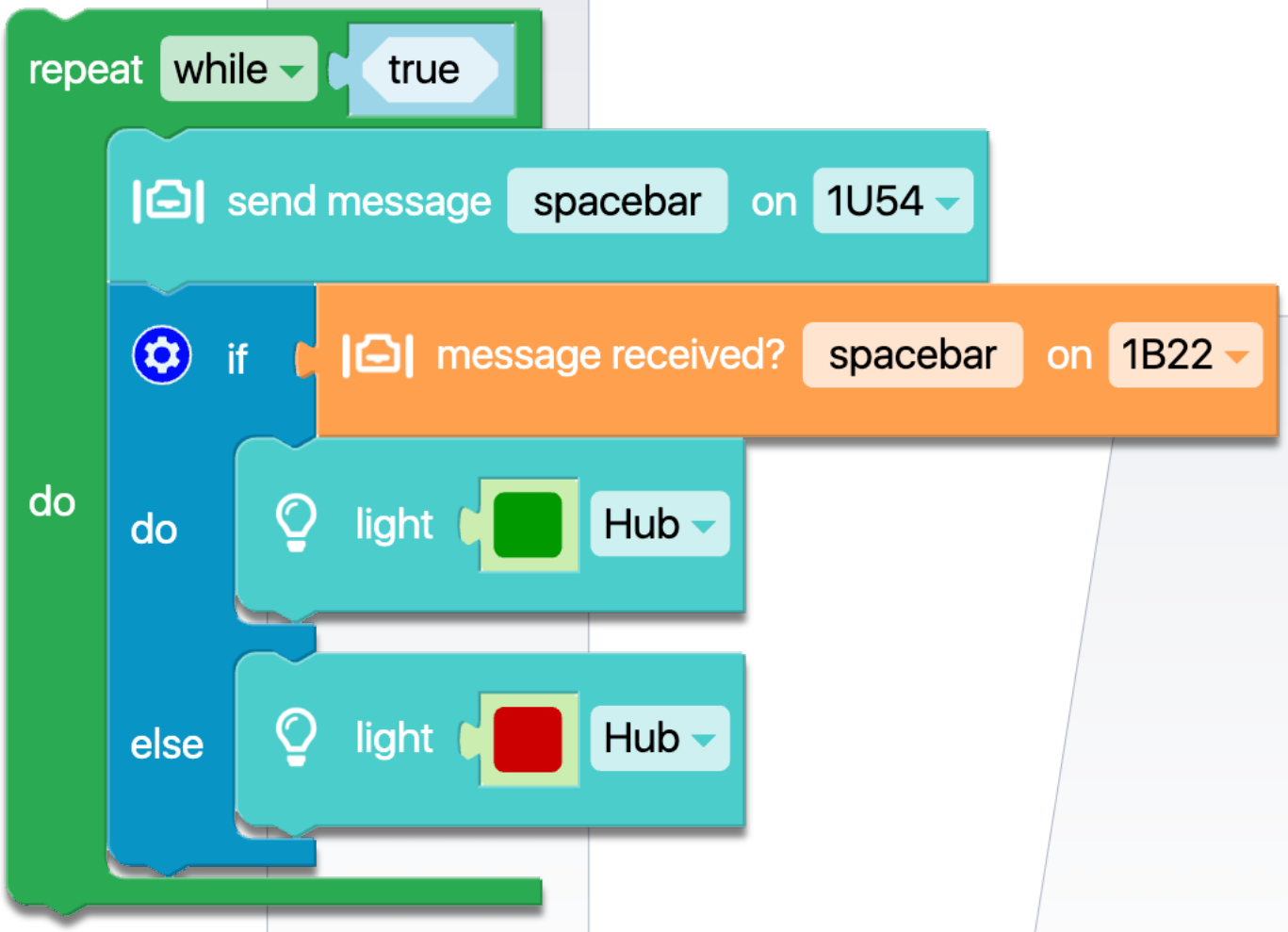
shape robotics
SMART LEARNING

Code 37:
InfraRed Detection

- Two Spin modules are positioned to face each other. One of them emits a continuous message through an Infrared channel. As long as the message is received by the other Spin module, the Hub remains illuminated in green. However, if an obstacle interrupts the Infrared flow, such as a Fable accessory, another robot, a ball, or a hand, the Hub will change to red. This program can be employed for the automatic detection of the ball during a football match.

WATCH VIDEO



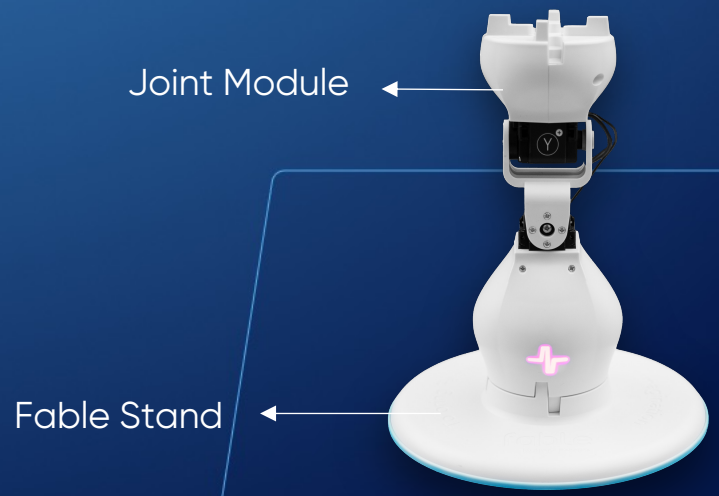

repeat while true
send message spacebar on 1U54
if message received? spacebar on 1B22
do
light Hub
do
else
light Hub
else

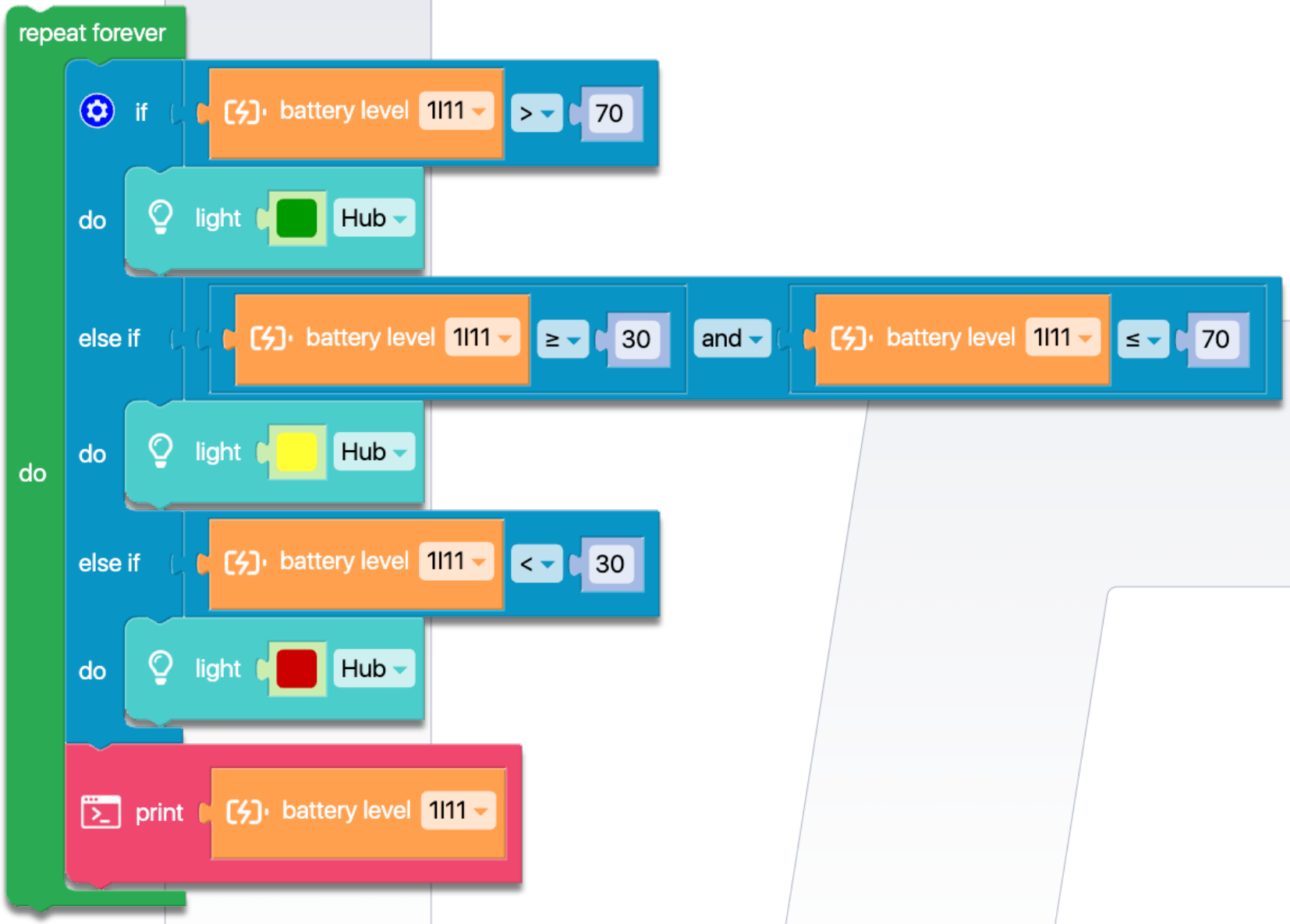
shape robotics
SMART LEARNING

Code 38:
Monitoring Battery Level

- The program continually monitors the battery level of a Joint or Spin mode and illuminates the Hub with colors corresponding to the level. When the battery charge is within the range of 70 to 100 percent, the displayed color is green. For the range of 30 to 70 percent, the color displayed is yellow, and for the range of 0 to 30 percent, the color displayed is red.

WATCH VIDEO

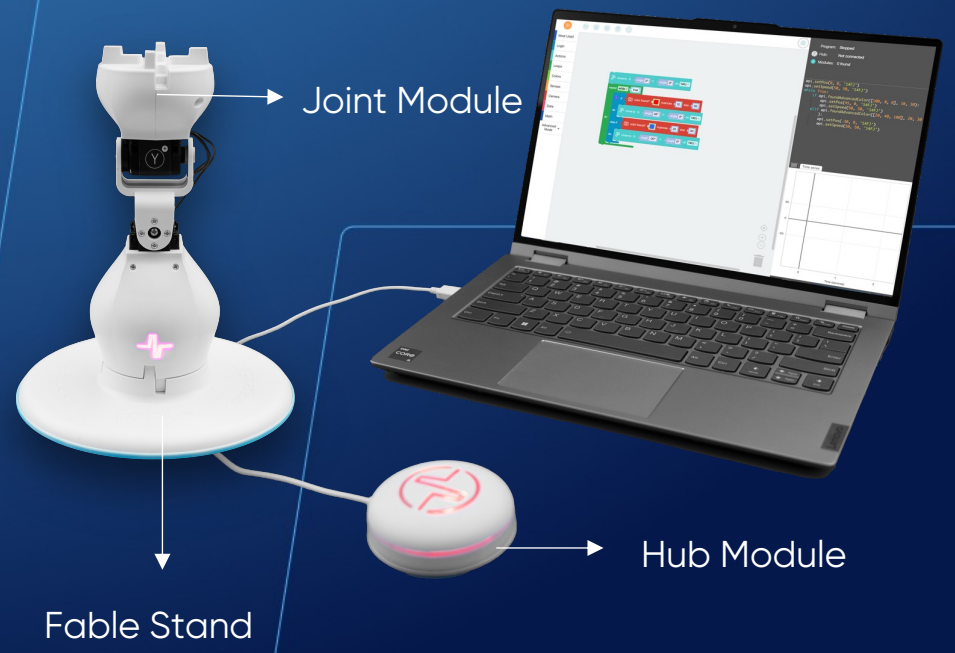


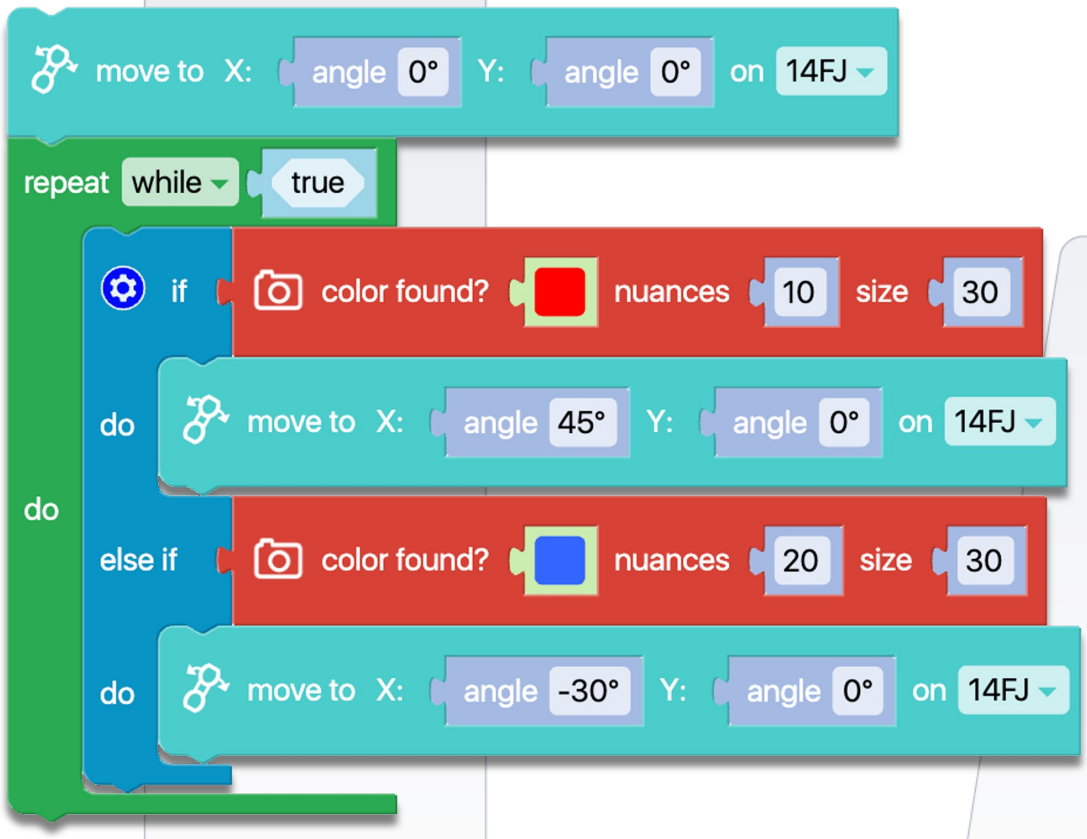

repeat forever
do
if
battery level 1l11
>
70
do
light
Hub
else if
battery level 1l11
≥
30
and
battery level 1l11
≤
70
do
light
Hub
else if
battery level 1l11
<
30
do
light
Hub
print
battery level 1l11

Code 39:
Joint controlled by Webcam Color Detection

- The following sequence utilizes the camera of the device where the programming is being carried out. It sets the color, nuance, and area(percent) to be detected. Whenever a color is detected, the sequence commands a Joint mode. Upon detecting blue, the Joint moves to -30 degrees for engine X, and upon detecting red, it moves to 45 degrees for engine X.

WATCH VIDEO



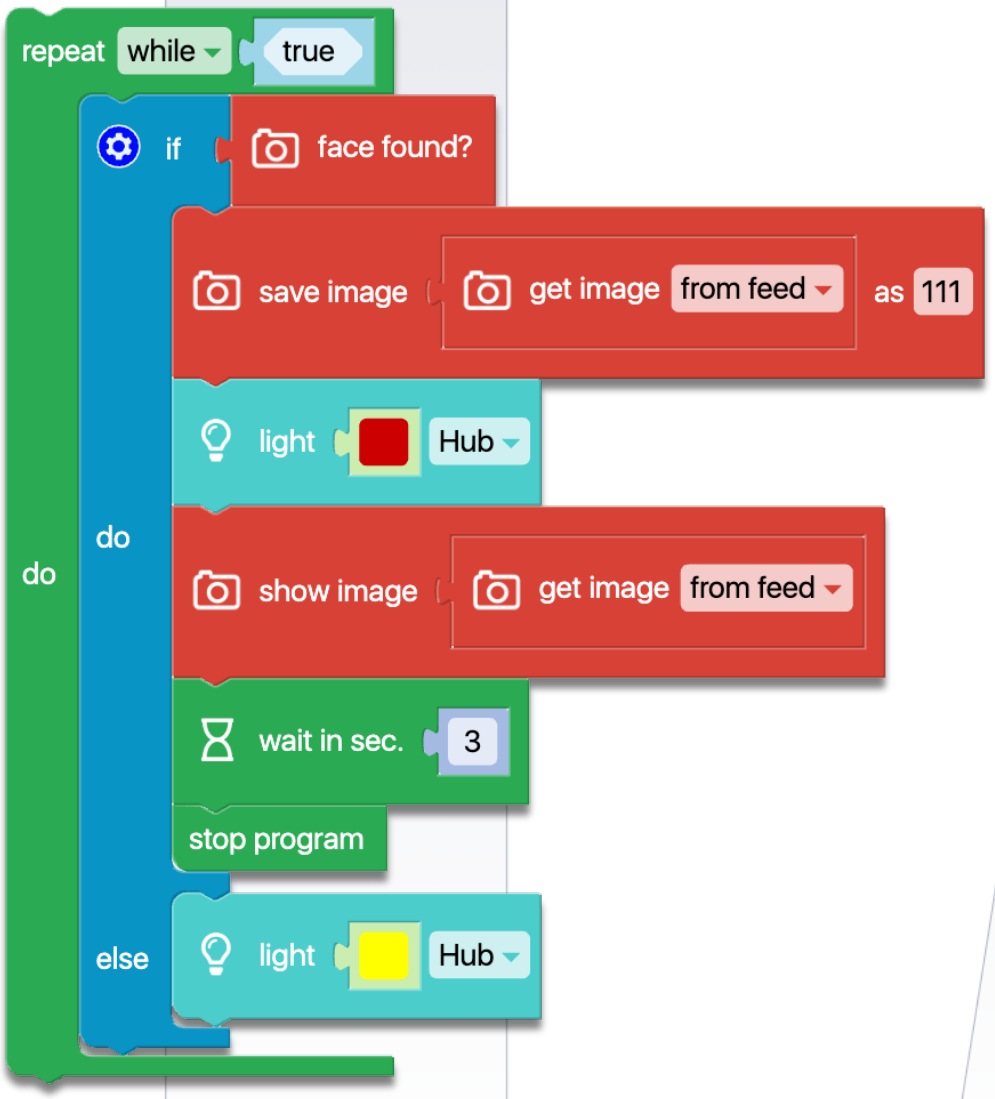

move to X: angle 0° Y: angle 0° on 14FJ
repeat while true
if color found? nuances 10 size 30
do move to X: angle 45° Y: angle 0° on 14FJ
else if color found? nuances 20 size 30
do move to X: angle -30° Y: angle 0° on 14FJ

shape robotics
SMART LEARNING

Code 40:
Webcam detecting Face

- This sequence activates the webcam of the device it is running on and continually monitors the images it captures for the presence of a human face. Upon detecting a face, the Hub changes color from yellow to red. Simultaneously, the program captures a screenshot of the detection and displays it for three seconds before shutting down completely.

WATCH VIDEO

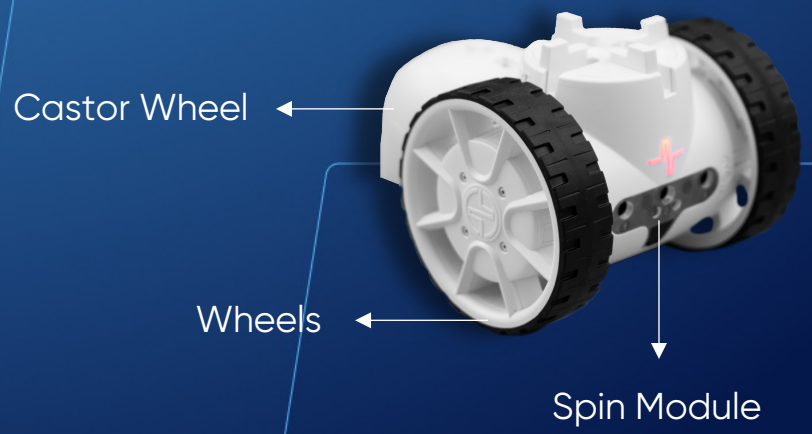

repeat while true
if face found?
save image get image from feed as 111
light Hub
do
show image get image from feed
wait in sec. 3
stop program
do
else
light Hub

shape robotics
SMART LEARNING

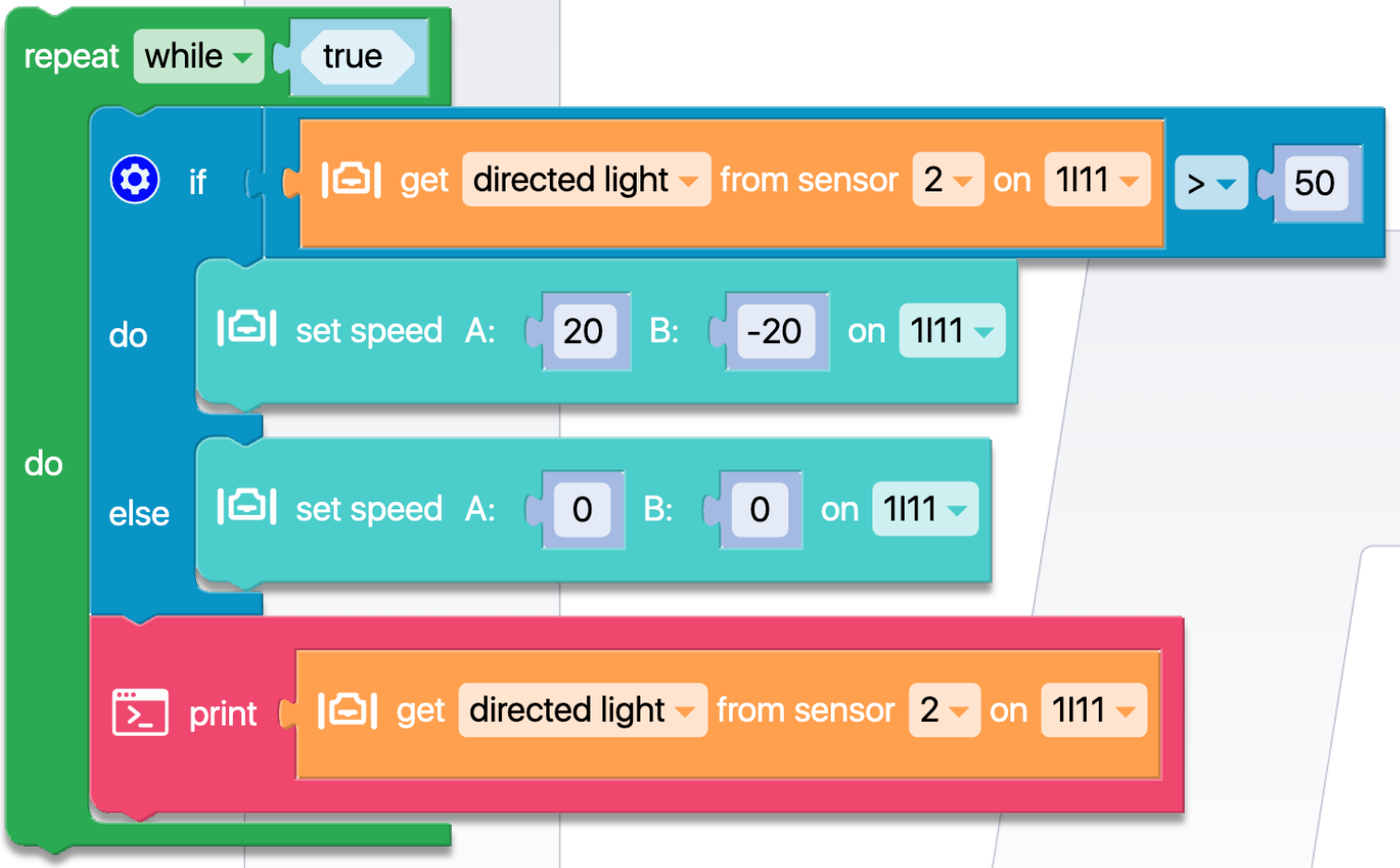
Code 41:
React to changes in light intensity

- The program is intended to maintain the Spin module in an area with reduced direct light exposure on the sensors. When the amount of light exceeds a predetermined value set in the code, the module reverses its movement until it returns to an area with illumination below the threshold value. This sequence can be applied in various projects, such as one involving a Spin module used to operate a sunshade for plants, pulling or pushing it as needed.

WATCH VIDEO



shape robotics
SMART LEARNING

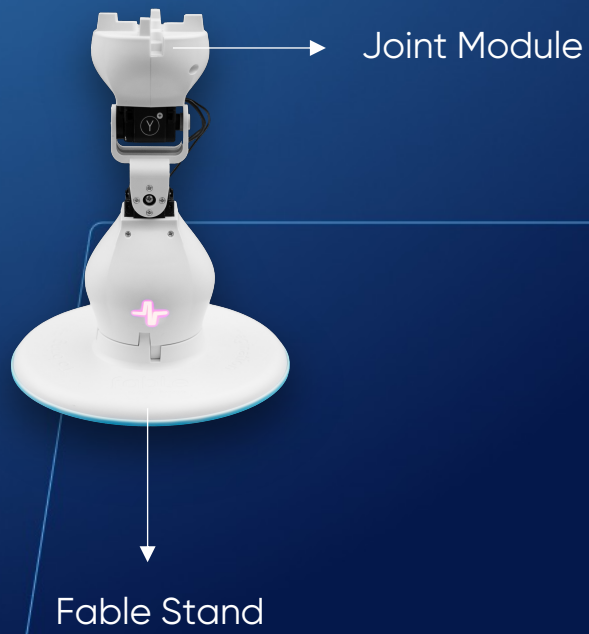


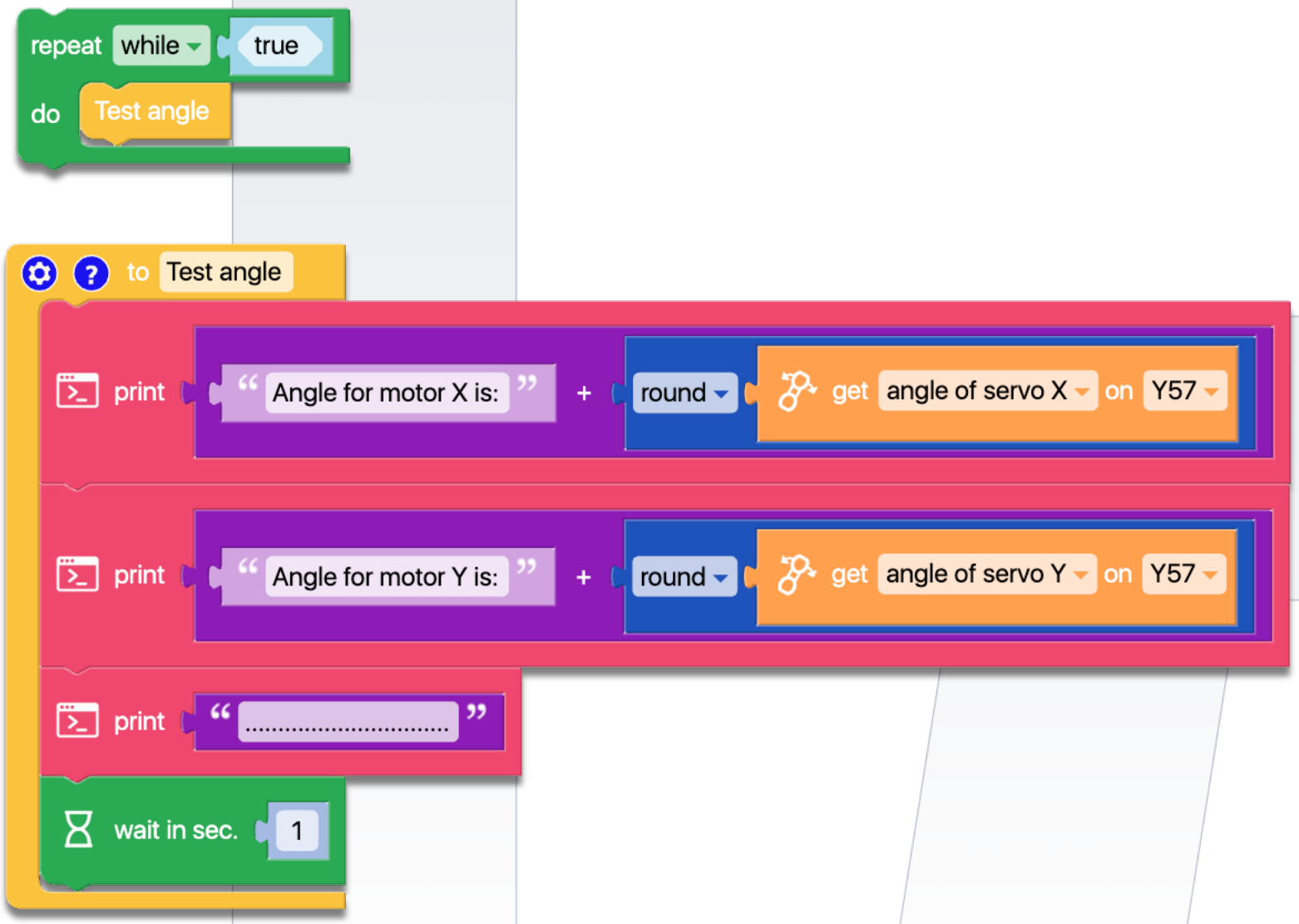
shape robotics
SMART LEARNING

Code 42:
Detecting Joint Position Angles

- The code sequence is useful for situations where we need to position the Joint mode servo motors under an angle that is detectable but not known in advance. This sequence consistently displays the servo motor angles

WATCH VIDEO



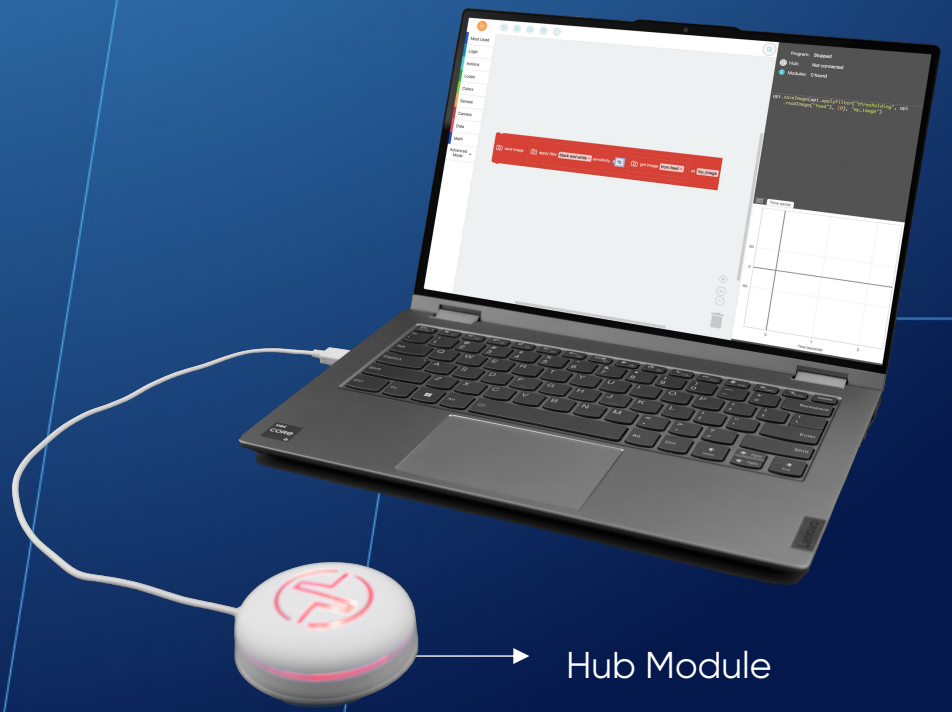

repeat while true
do Test angle
to Test angle
print "Angle for motor X is:" + round get angle of servo X on Y57
print "Angle for motor Y is:" + round get angle of servo Y on Y57
print "....."
wait in sec. 1

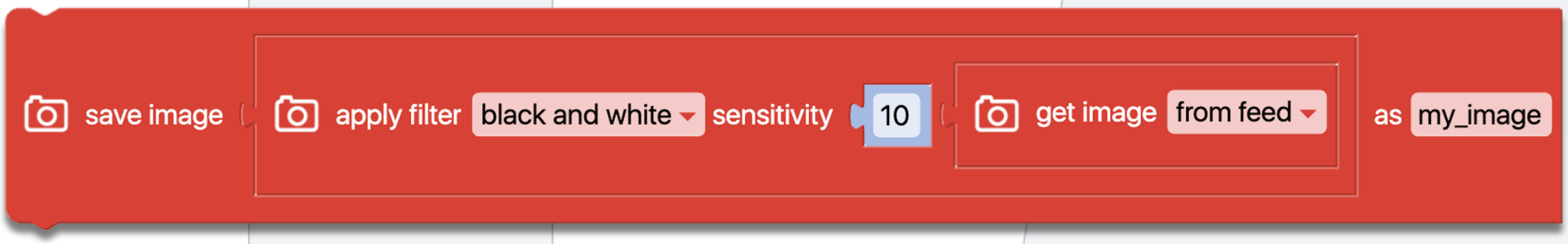
shape robotics
SMART LEARNING

Code 43:
Applying filter on Webcam Screenshot

- The sequence combines three commands: Save image, Apply filter, and Get image from feed. Together, they capture an image from the webcam of the device where the programming is executed, saving the picture as 'my_image'. The type of filter applied, the sensitivity (for some filters), and the name of the capture can be customized.

WATCH VIDEO



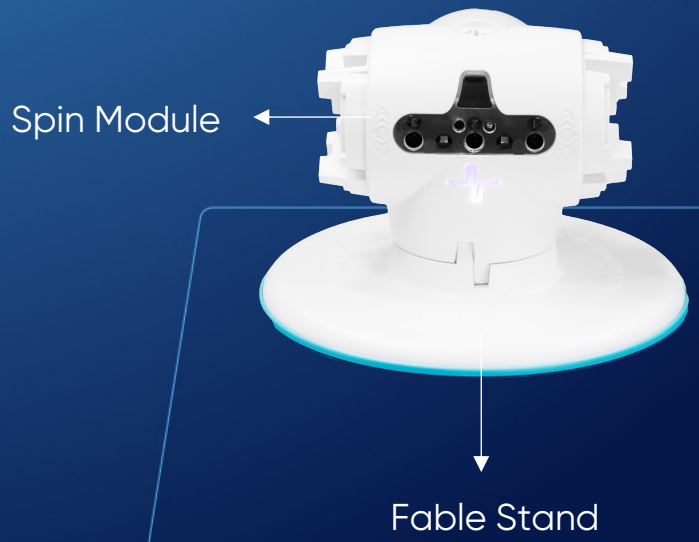

save image
apply filter
black and white
sensitivity
10
get image
from feed
as
my_image

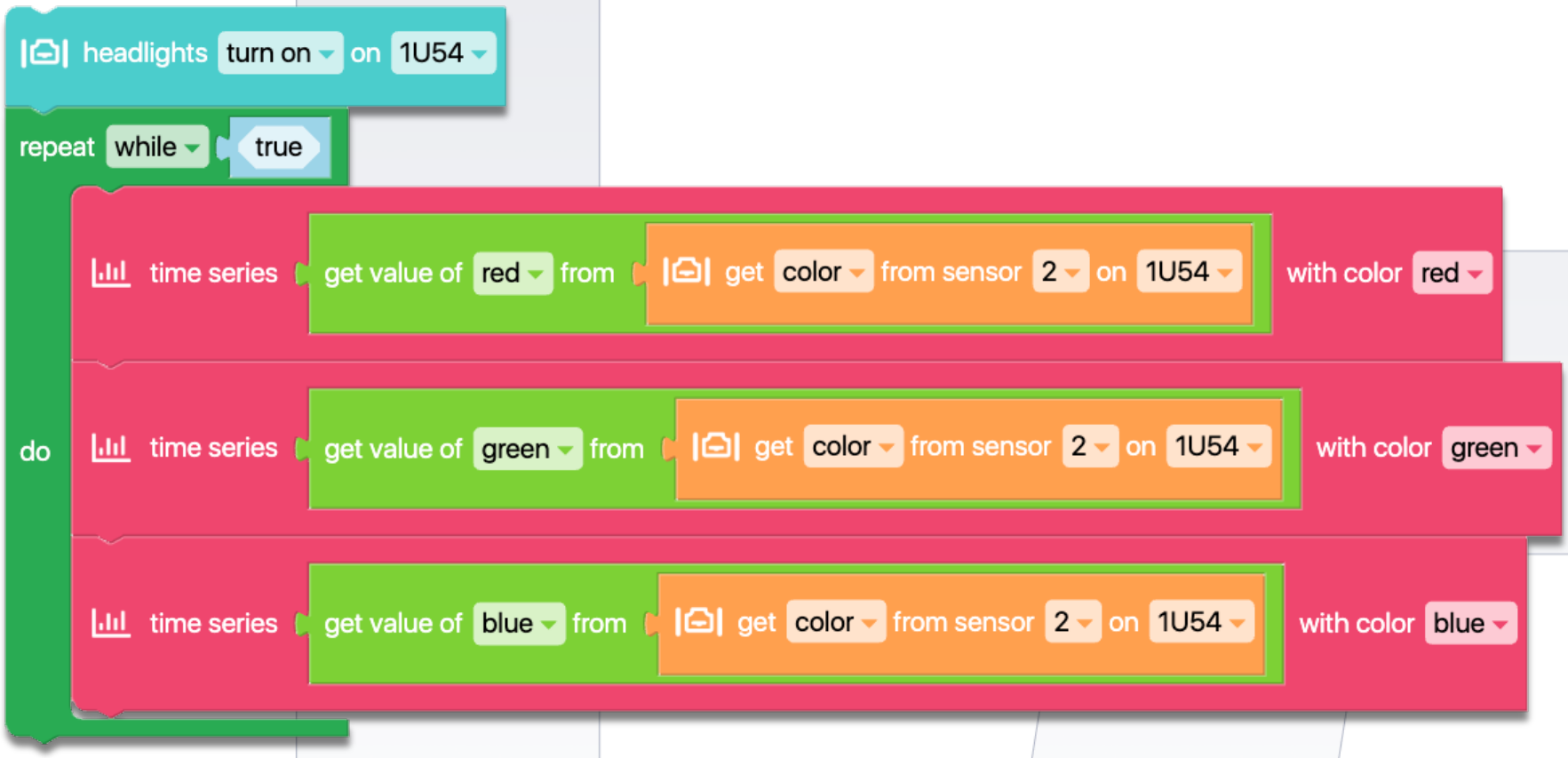
shape robotics
SMART LEARNING

Code 44:
Extracting RGB Levels

- The code sequence "extracts" R, G, B values from a color detected in front of a Spin module. The color is analyzed, and then the R,G,B, are displayed on graphs based on these values.

WATCH VIDEO



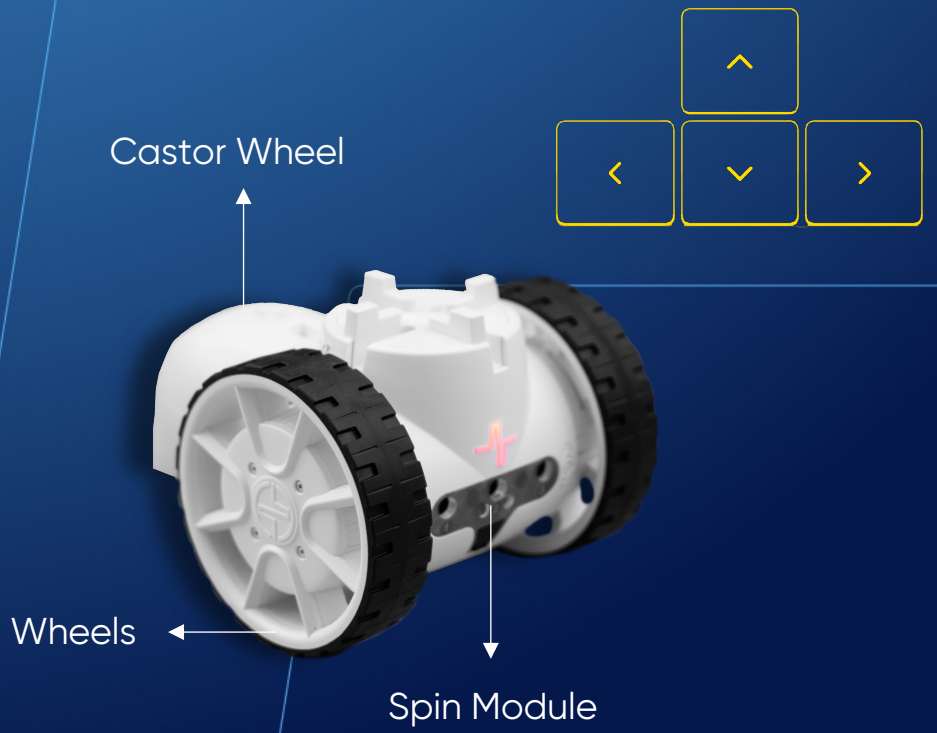

headlights turn on on 1U54
repeat while true
do
time series get value of red from get color from sensor 2 on 1U54 with color red
time series get value of green from get color from sensor 2 on 1U54 with color green
time series get value of blue from get color from sensor 2 on 1U54 with color blue

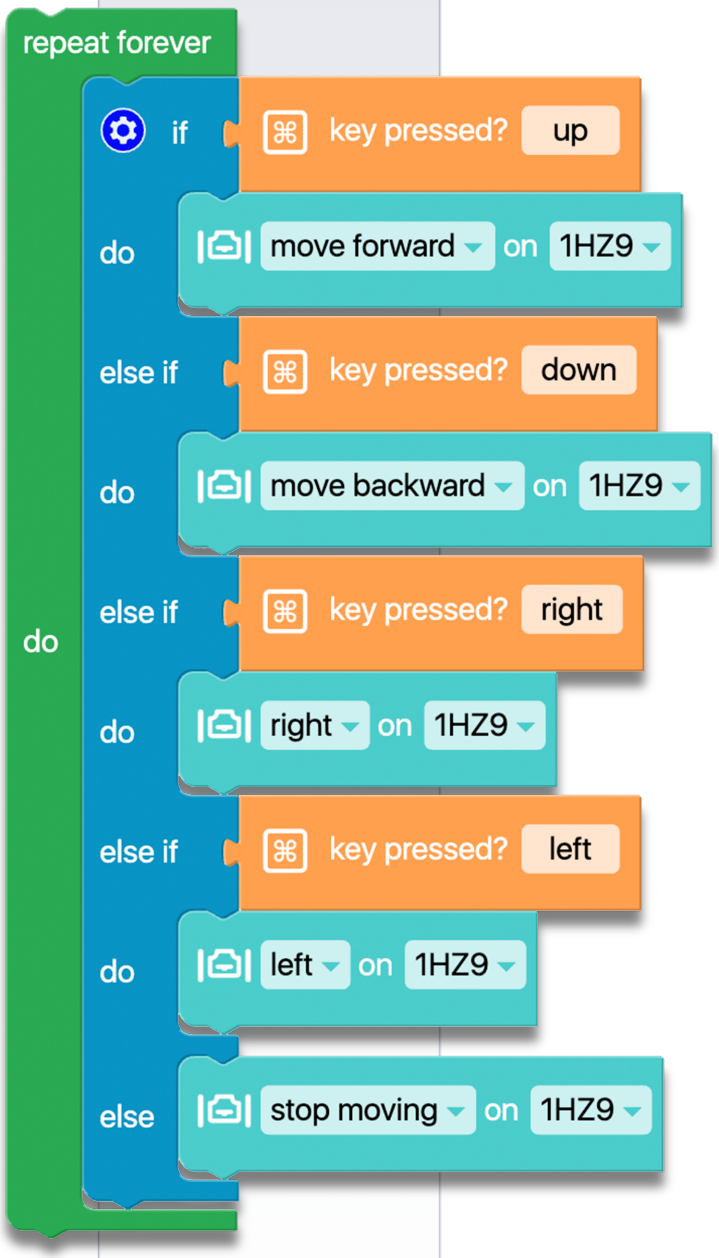
shape robotics
SMART LEARNING

Code 45:
Spin Key Control

- The program is designed to control the Spin module, using key inputs. The program runs in a continuous loop that allows for the detection of key presses at all times. If no key is pressed, the Spin robotic module will stop because its command will be to Stop moving.

WATCH VIDEO





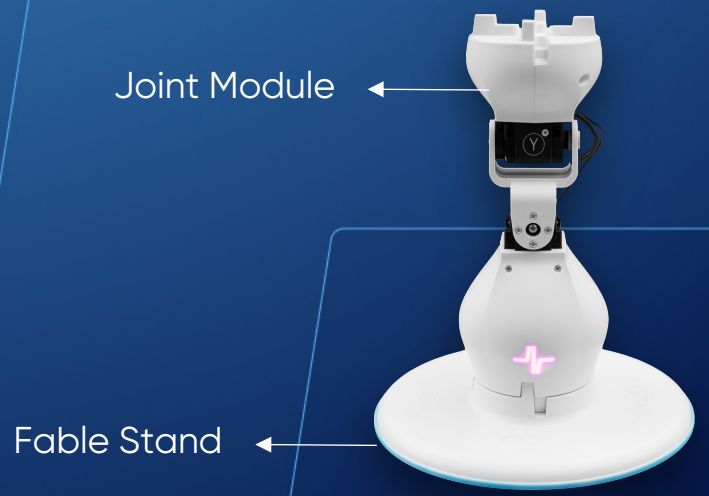
shape robotics
SMART LEARNING

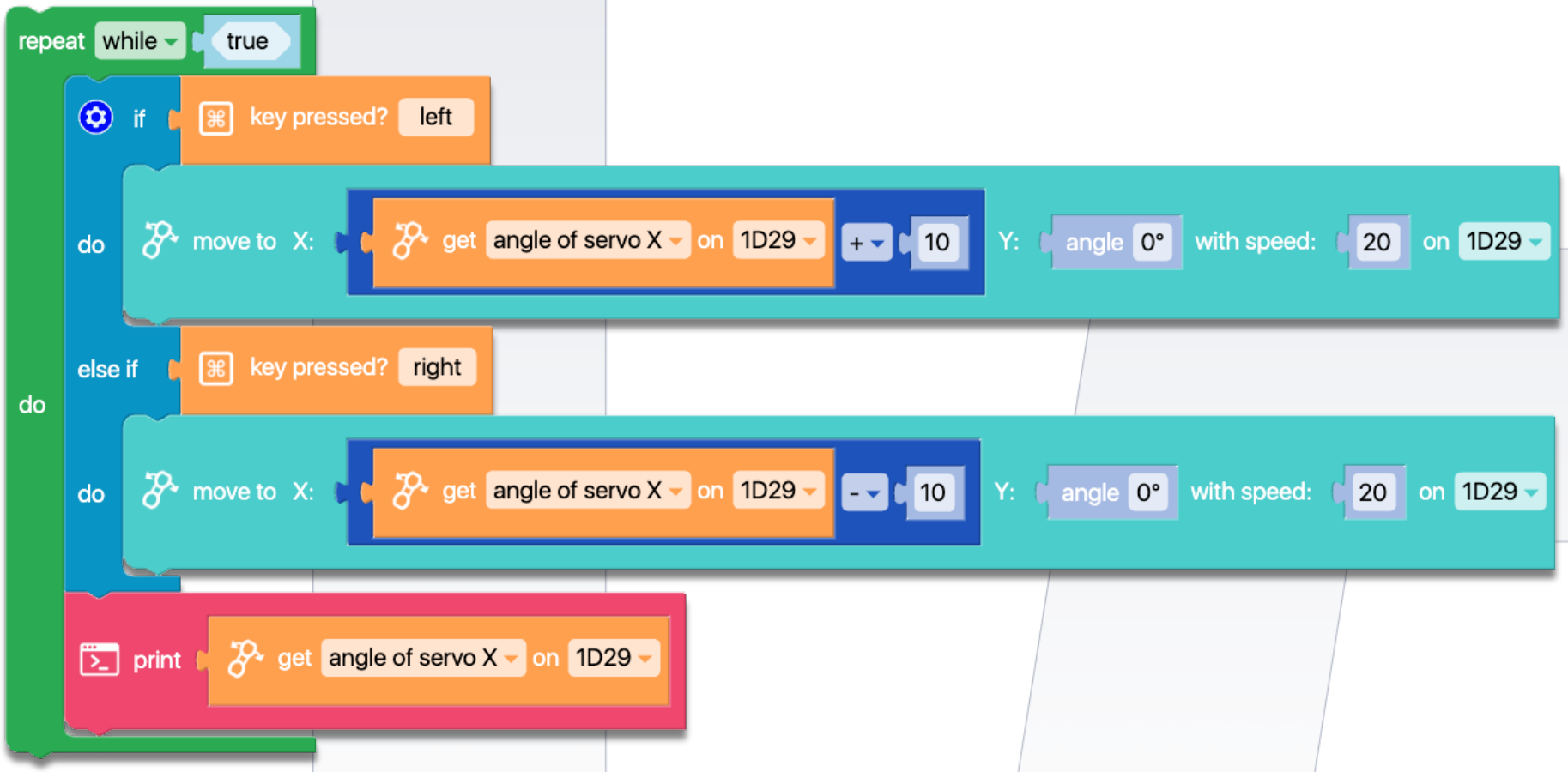
shape robotics
SMART LEARNING

Code 46:
Joint Key Control

- The Joint module is positioned vertically with both motors set at a zero angle. While the Y motor will retain this angle, the angle of the X motor can be adjusted by +10 or -10 units by pressing the left or right key, respectively. The angle value is displayed in the console output. This program sequence enables more precise control of the Joint module, particularly useful when operating a tool connected to it.

WATCH VIDEO



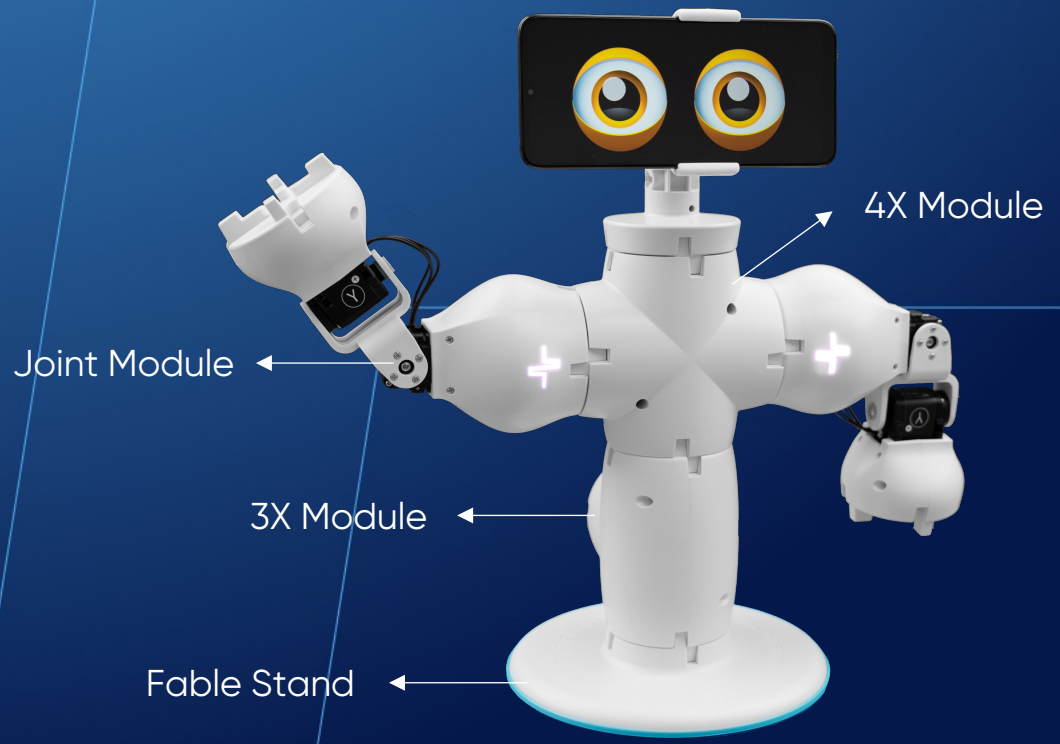

repeat while true
if key pressed? left
do move to X: get angle of servo X on 1D29 + 10 Y: angle 0° with speed: 20 on 1D29
else if key pressed? right
do move to X: get angle of servo X on 1D29 - 10 Y: angle 0° with speed: 20 on 1D29
do
print get angle of servo X on 1D29

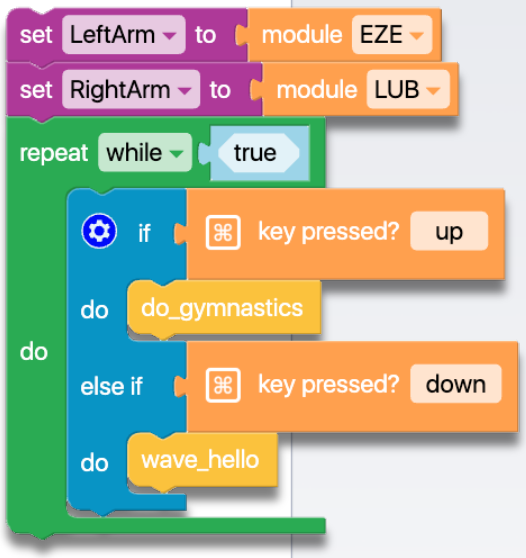
shape robotics
SMART LEARNING

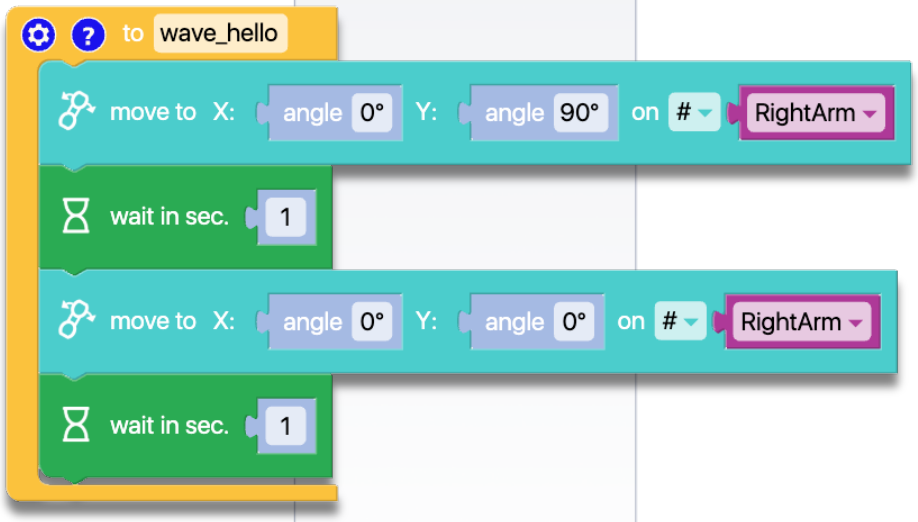
Code 47:
Fable Hello

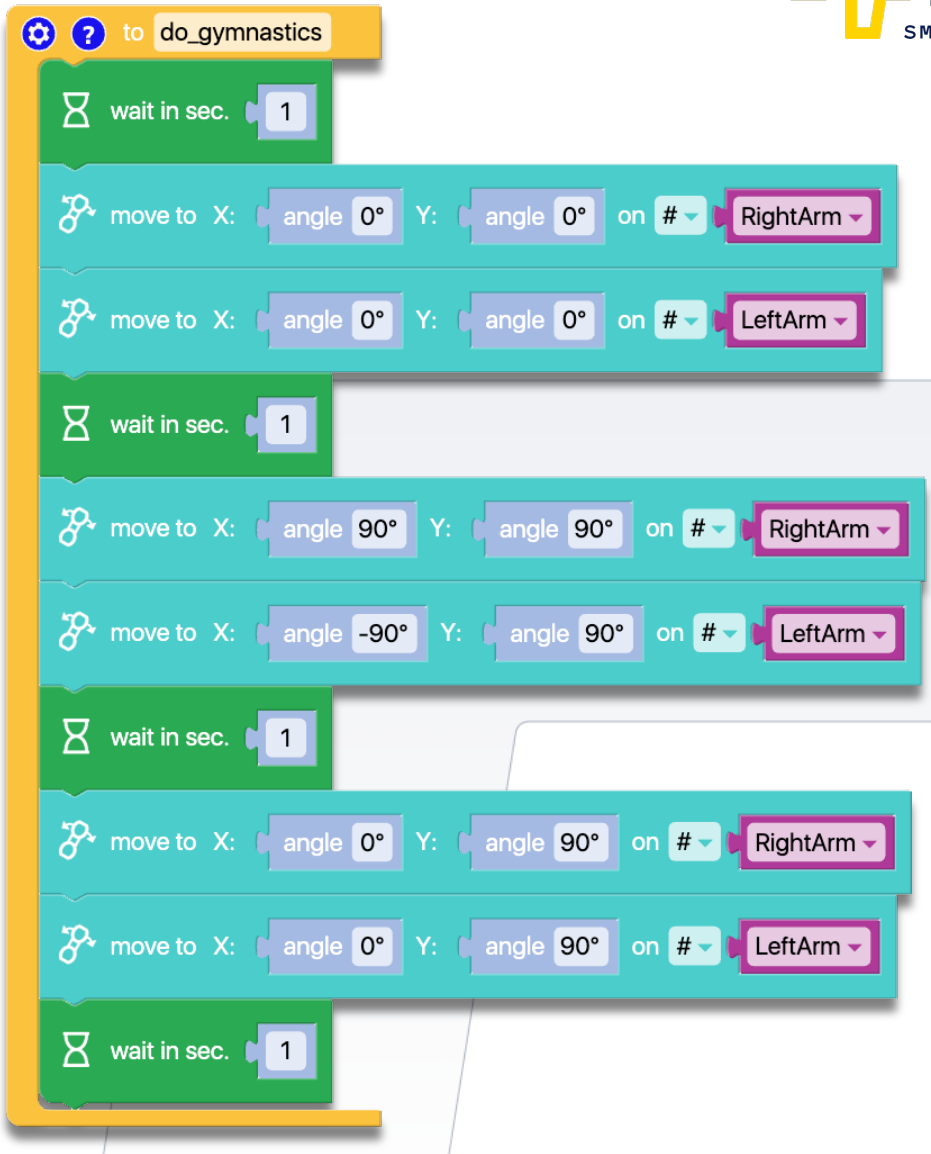
- The program consists of two functions, namely "do_gymnastic" and "wave_hello". These functions are activated by pressing the Up-arrow key and the Down-arrow key, respectively. The movements in these functions are already defined. Moreover, variables have also been defined so that the robots can be changed quickly. This means that if you have a new robot, all you need to do is fill in thecorrect code in the variable at the beginning of the program.

WATCH VIDEO




set LeftArm to module EZE
set RightArm to module LUB
repeat while true
do
if key pressed? up
do do_gymnastics
else if key pressed? down
do wave_hello


to wave_hello
move to X: angle 0° Y: angle 90° on # RightArm
wait in sec. 1
move to X: angle 0° Y: angle 0° on # RightArm
wait in sec. 1

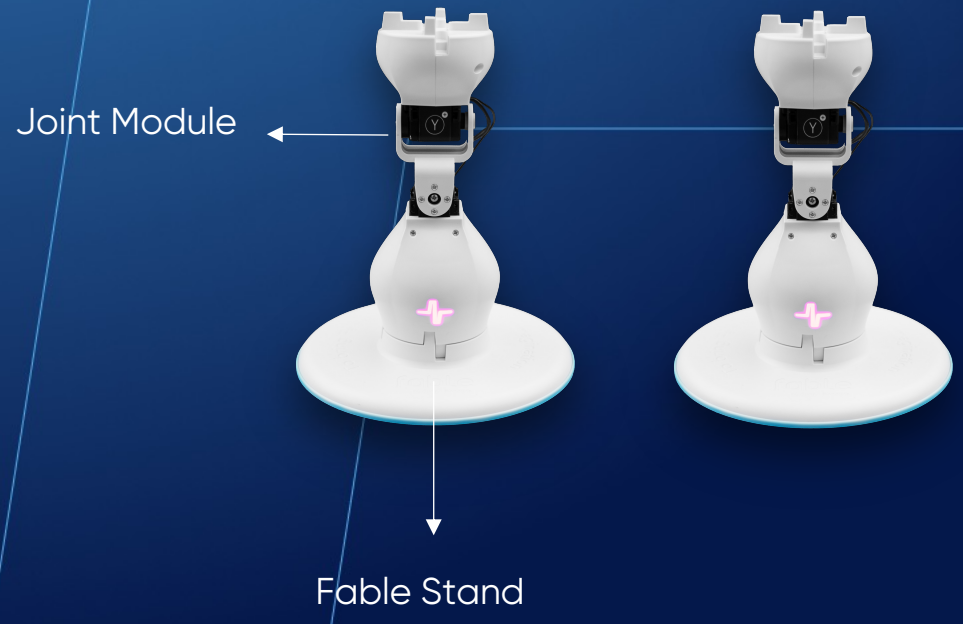

to do_gymnastics
wait in sec. 1
move to X: angle 0° Y: angle 0° on # RightArm
move to X: angle 0° Y: angle 0° on # LeftArm
wait in sec. 1
move to X: angle 90° Y: angle 90° on # RightArm
move to X: angle -90° Y: angle 90° on # LeftArm
wait in sec. 1
move to X: angle 0° Y: angle 90° on # RightArm
move to X: angle 0° Y: angle 90° on # LeftArm
wait in sec. 1

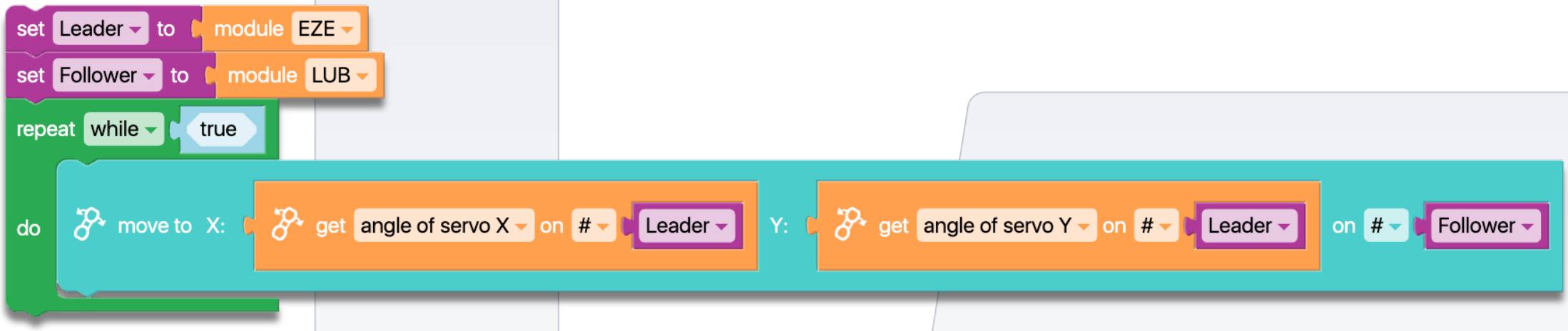
shape robotics
SMART LEARNING

Code 48:
Follow the Leader

- The program assigns roles to two robots - Leader and Follower - via two variables. During the run, the program reads the angles of the X and Y motors of the Leader robot and utilizes them to control the motion of the Follower robot. In other words, the second robot mirrors the movements of the first.

WATCH VIDEO

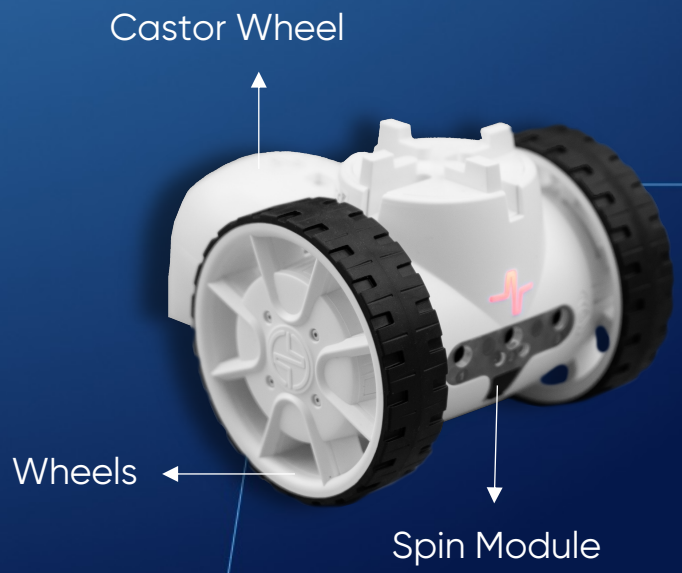


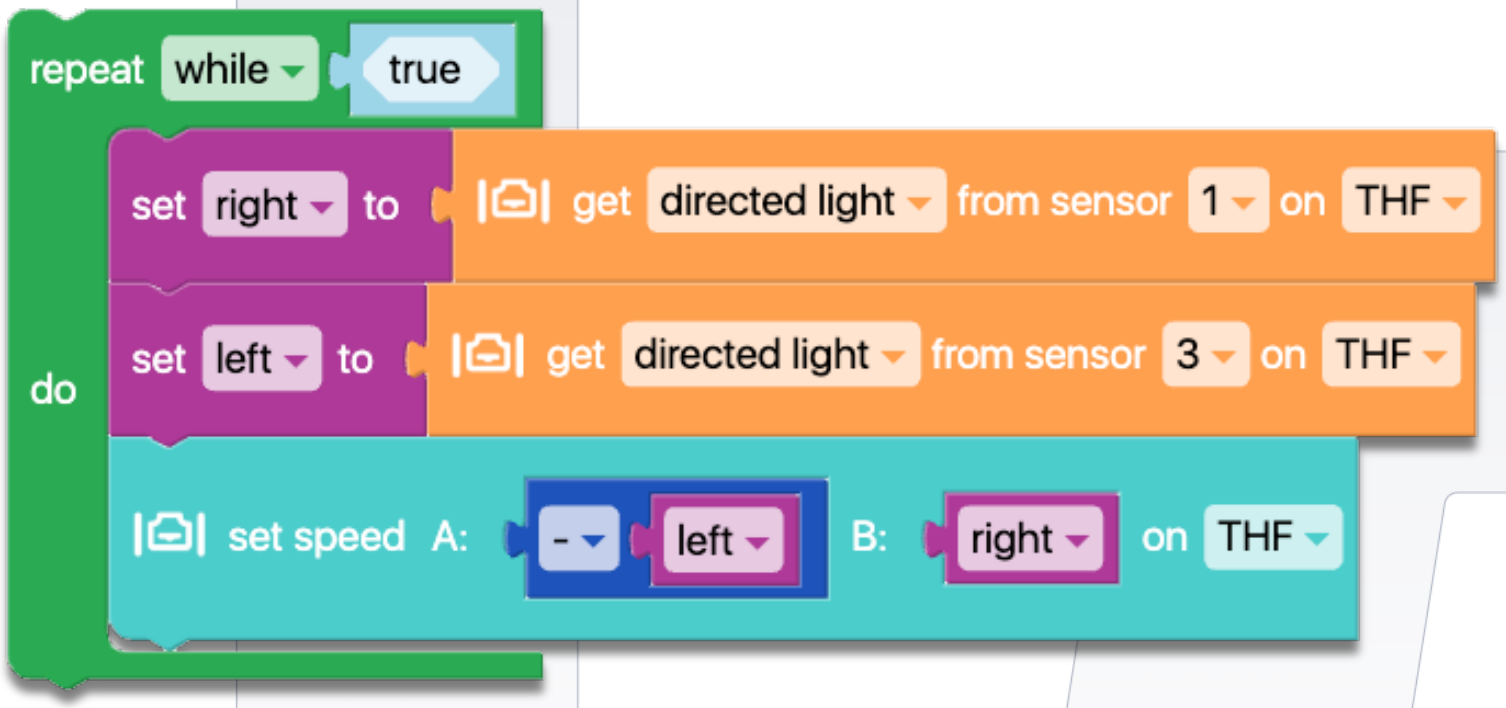

set Leader to module EZE
set Follower to module LUB
repeat while true
do
move to X:
get angle of servo X on # Leader
Y:
get angle of servo Y on # Leader
on # Follower

Code 49:
Into the Light

- The program utilizes two light sensors to measure the values recorded by sensor 1 and sensor 3. These values are then used to control the speed of the motors in a Spin module. It's important to note that motor A uses a negative value to enable forward motion of the Spin.

WATCH VIDEO



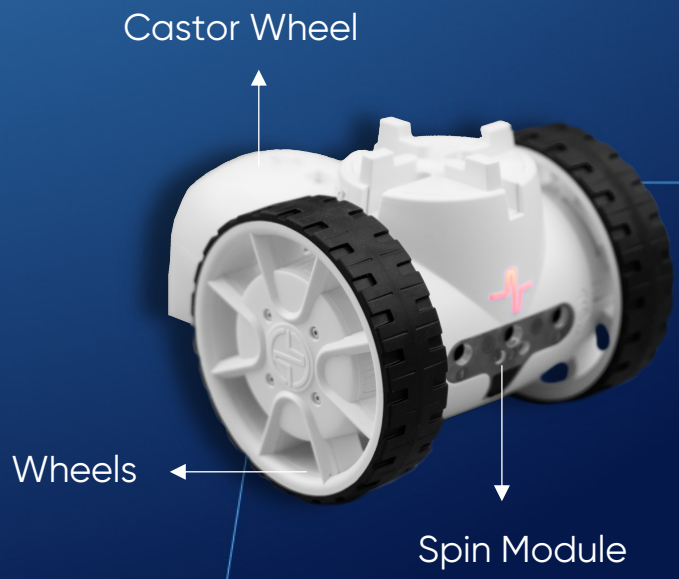

repeat while true
do
set right to get directed light from sensor 1 on THF
set left to get directed light from sensor 3 on THF
set speed A: - left B: right on THF

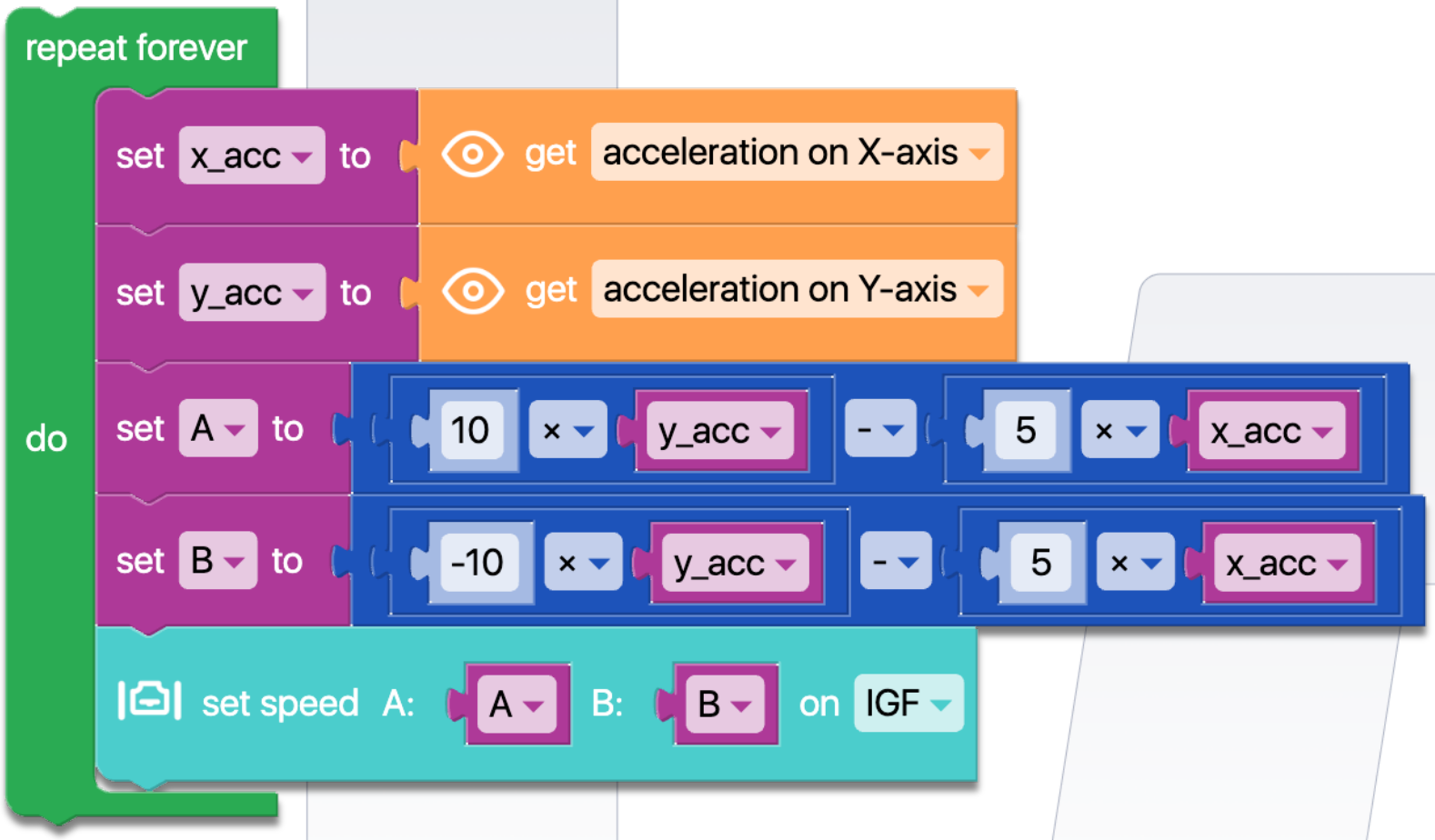
shape robotics
SMART LEARNING

Code 50:
Spin Remote Control

- The program utilizes a Spin module equipped with wheels and a caster wheel, paired with a phone connected via Fable Face and Hub. The Spin module receives acceleration data from the phone's axles and translates it into commands for its motors.
- By incorporating a plug-in accessory, you can engage in football using the ball provided in the kit, while remotely controlling the robot from your phone.

WATCH VIDEO



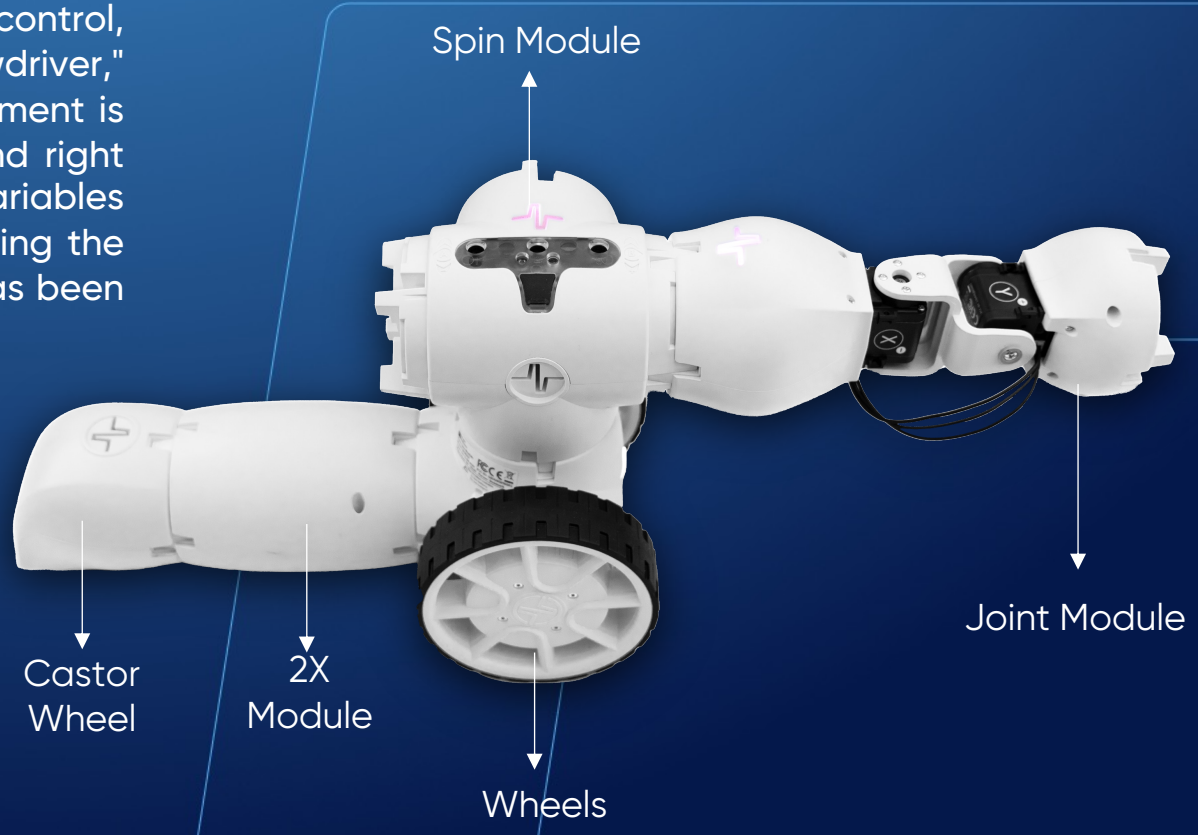

repeat forever
do
set x_acc to
get acceleration on X-axis
set y_acc to
get acceleration on Y-axis
set A to
10 × y_acc - 5 × x_acc
set B to
-10 × y_acc - 5 × x_acc
set speed A: A B: B on IGF

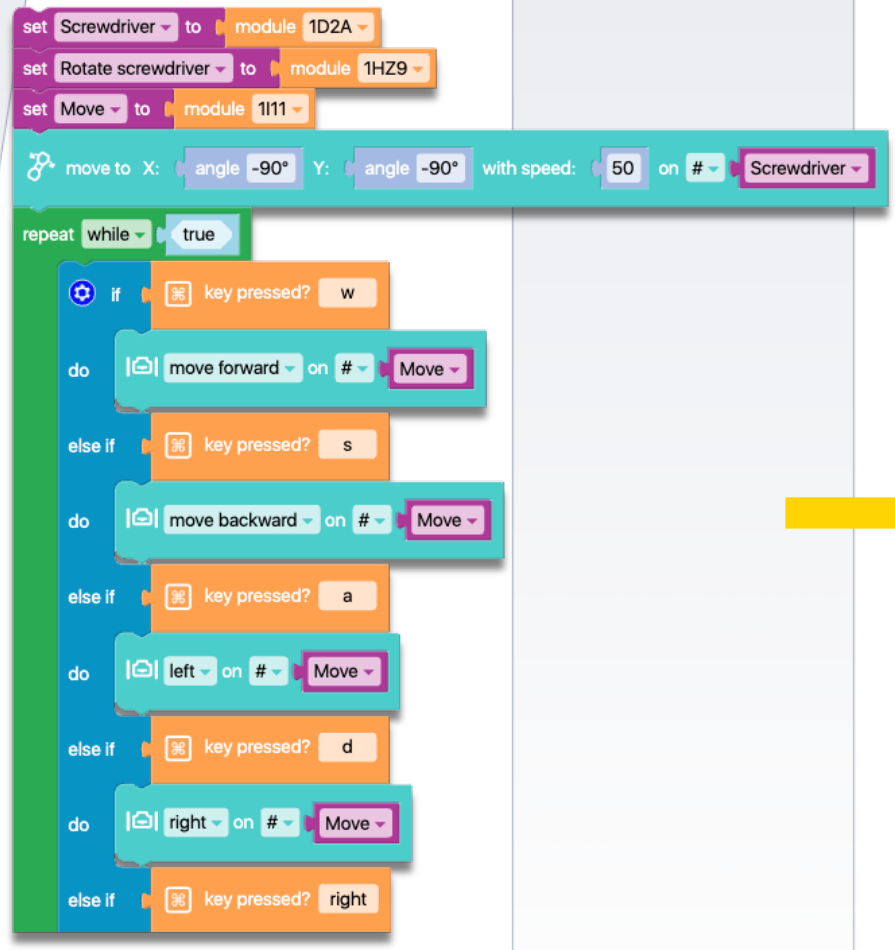
shape robotics
SMART LEARNING

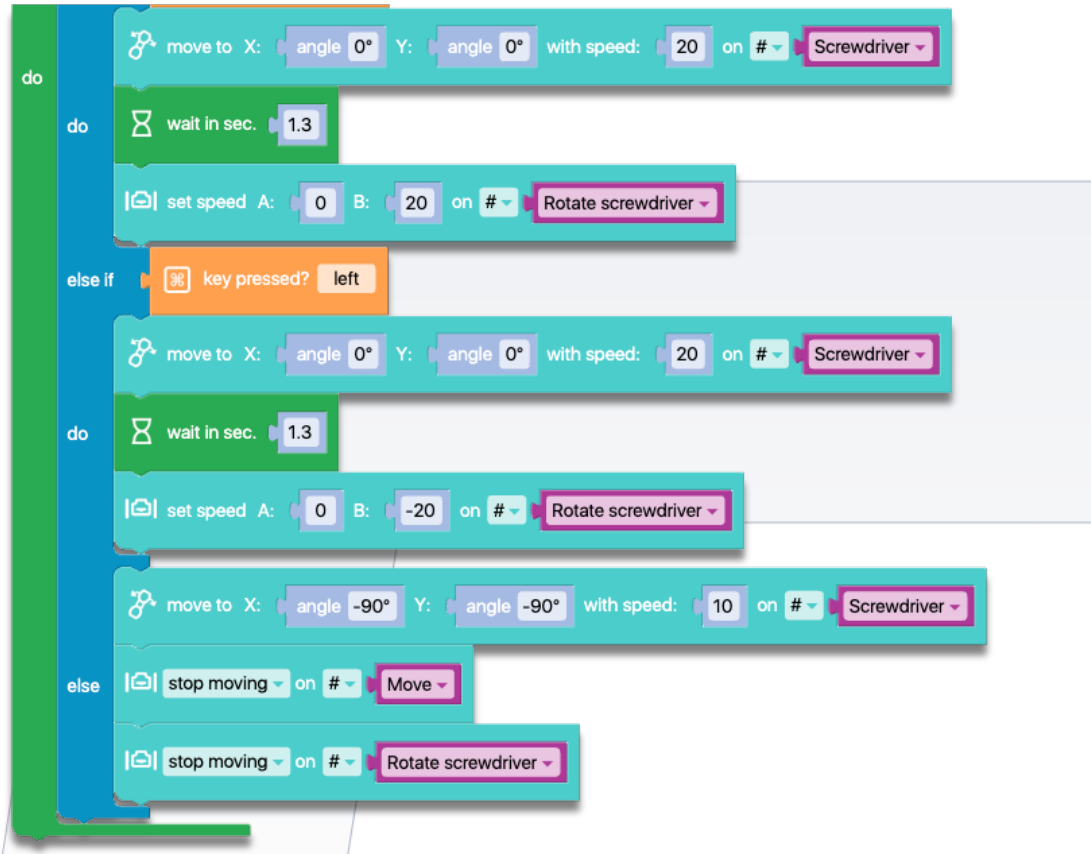
Code 51:
Joint as a Screwdriver

- The program employs a Spin module for movement control, another Spin module to regulate the rotation of the "screwdriver," and a "screwdriver" represented by a Joint module. Movement is controlled using the keys w, a, s, and d, while the left and right arrow keys control the screwdriver's rotation. Declared variables facilitate easy replacement of the robot by simply modifying the code at the program's outset. Additionally, a 2X module has been incorporated for balance.

WATCH VIDEO




set Screwdriver to module 1D2A
set Rotate screwdriver to module 1HZ9
set Move to module 1I11
move to X: angle -90° Y: angle -90° with speed: 50 on # Screwdriver
repeat while true
if key pressed? w
do move forward on # Move
else if key pressed? s
do move backward on # Move
else if key pressed? a
do left on # Move
else if key pressed? d
do right on # Move
else if key pressed? right

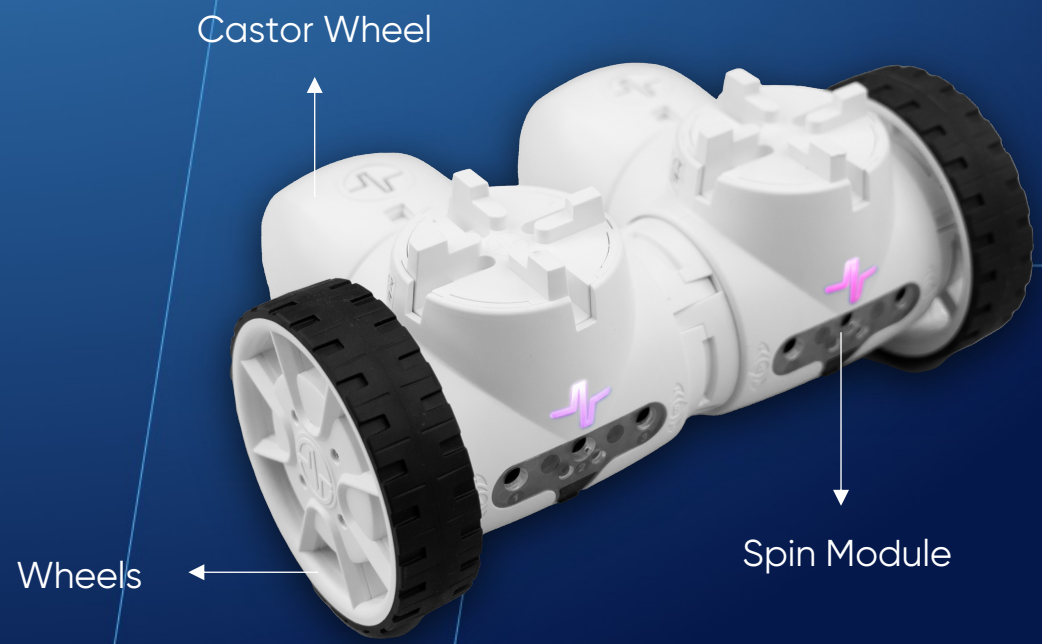

do
move to X: angle 0° Y: angle 0° with speed: 20 on # Screwdriver
do
wait in sec. 1.3
set speed A: 0 B: 20 on # Rotate screwdriver
else if key pressed? left
move to X: angle 0° Y: angle 0° with speed: 20 on # Screwdriver
do
wait in sec. 1.3
set speed A: 0 B: -20 on # Rotate screwdriver
move to X: angle -90° Y: angle -90° with speed: 10 on # Screwdriver
else
stop moving on # Move
stop moving on # Rotate screwdriver

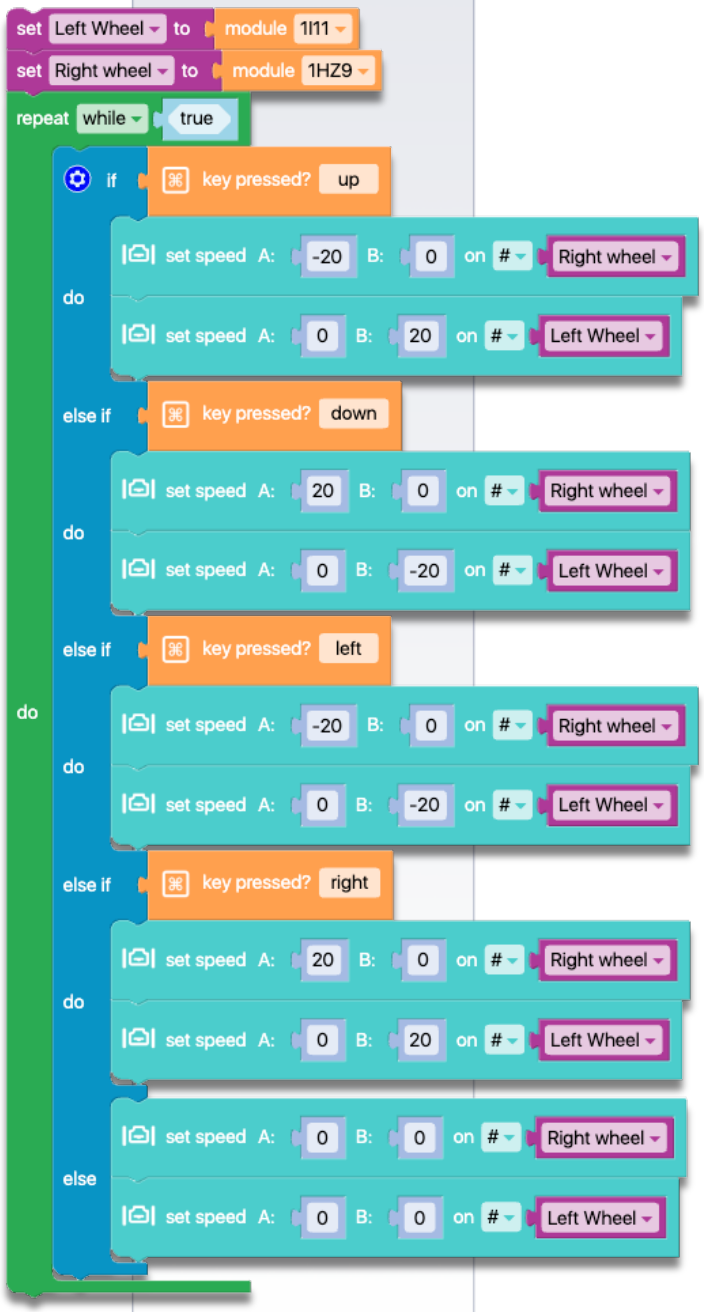
shape robotics
SMART LEARNING

Code 52:
Siamese structure for Spins

- The project utilizes two Spin modules and only two wheels. To facilitate wheel movement, two motors are interconnected and maintained at zero speed. When turning, one wheel from one Spin module and one wheel from the other Spin module are utilized.

WATCH VIDEO





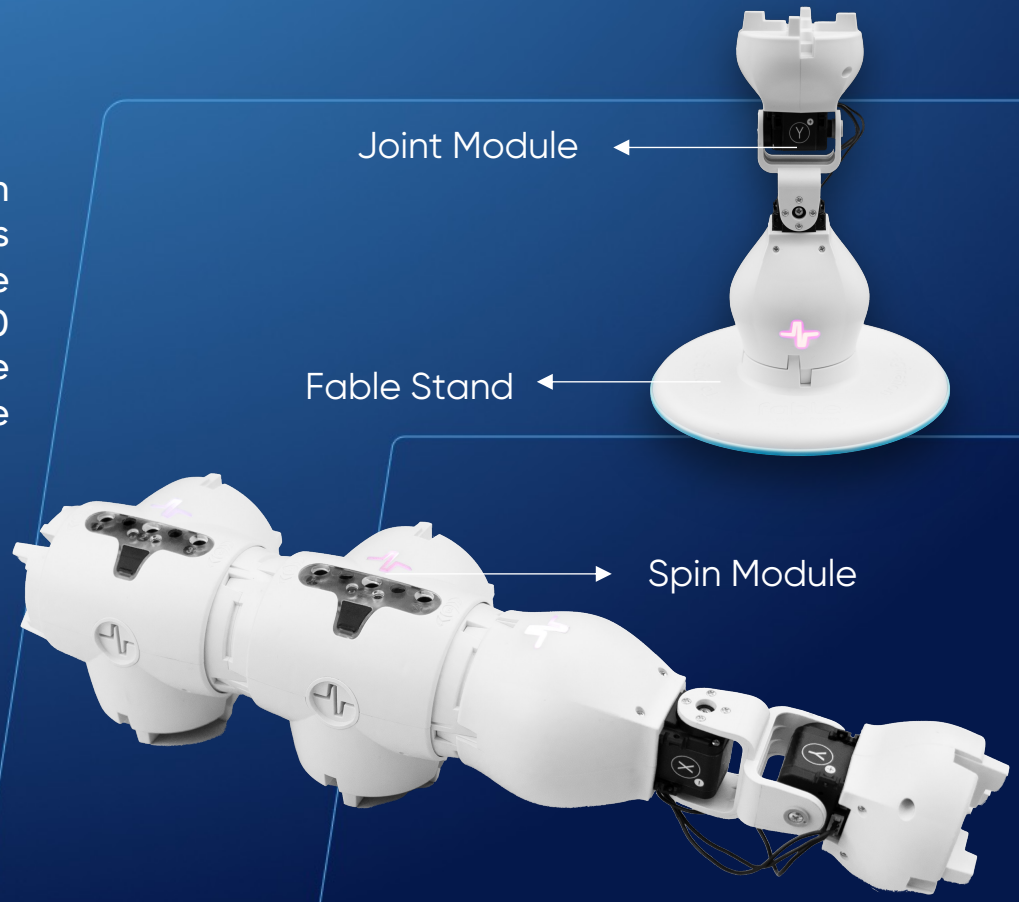
shape robotics
SMART LEARNING

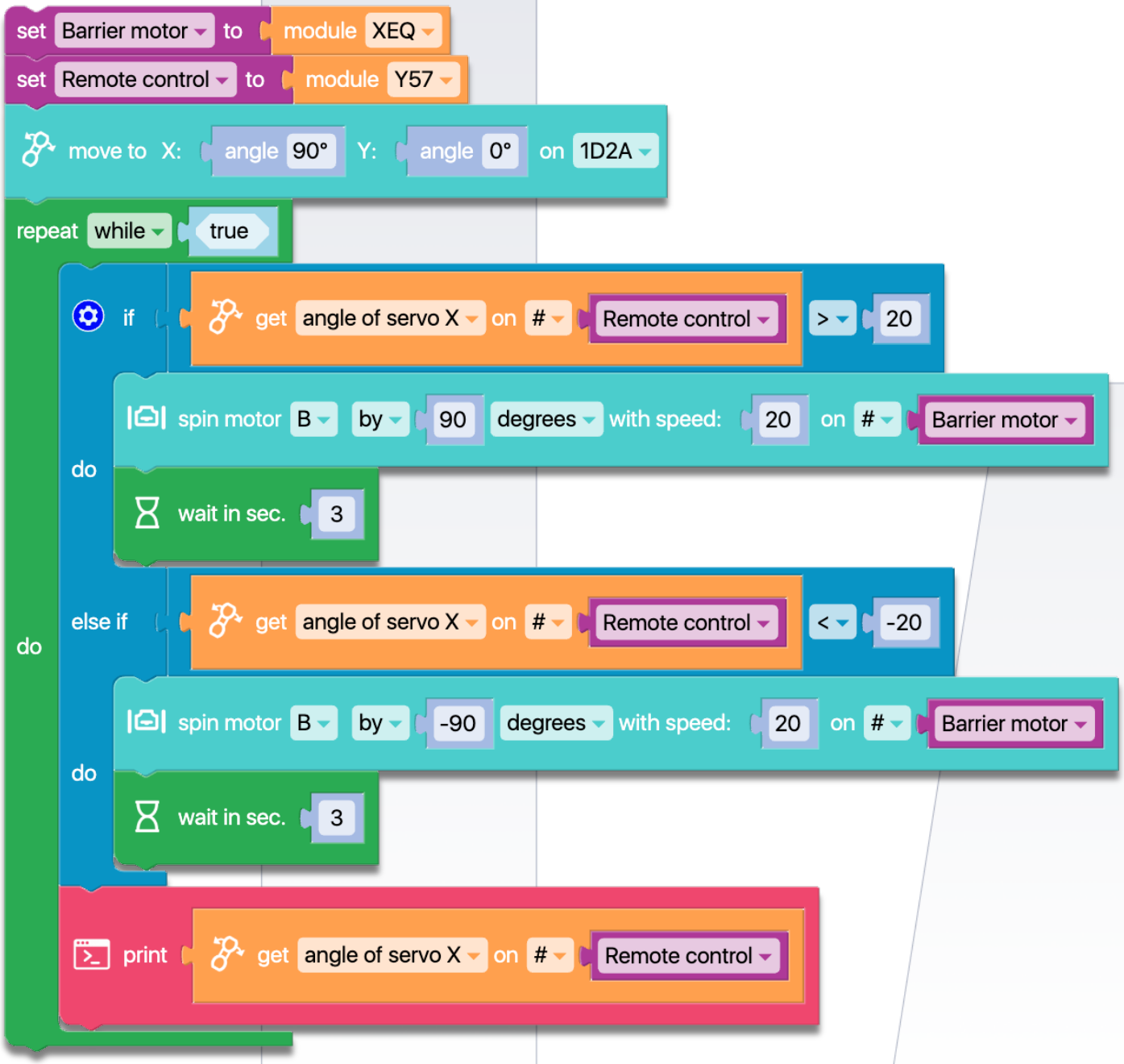
shape robotics
SMART LEARNING

Code 53:
Remote Barrier Control

- The barrier is controlled by a Joint module linked to one of the Spin engines. Another Joint communicates the angle at which the X motor is rotated. When the angle exceeds 20 degrees, the motor controlling the barrier rotates by 90 degrees. Conversely, if the angle falls below -20 degrees, the rotation is reversed. The values of 20 and -20 degrees were selected to allow a range of movement for the remote Joint. The angle of the X motor is displayed in the output console.

WATCH VIDEO





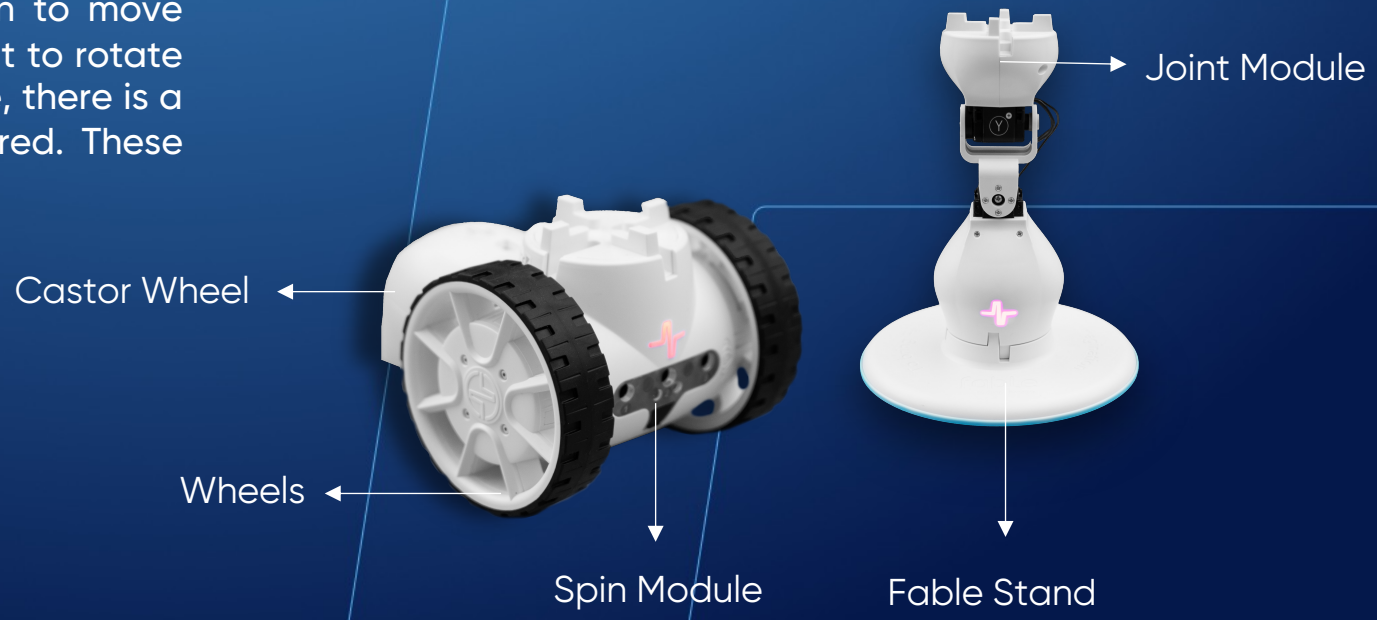
shape robotics
SMART LEARNING

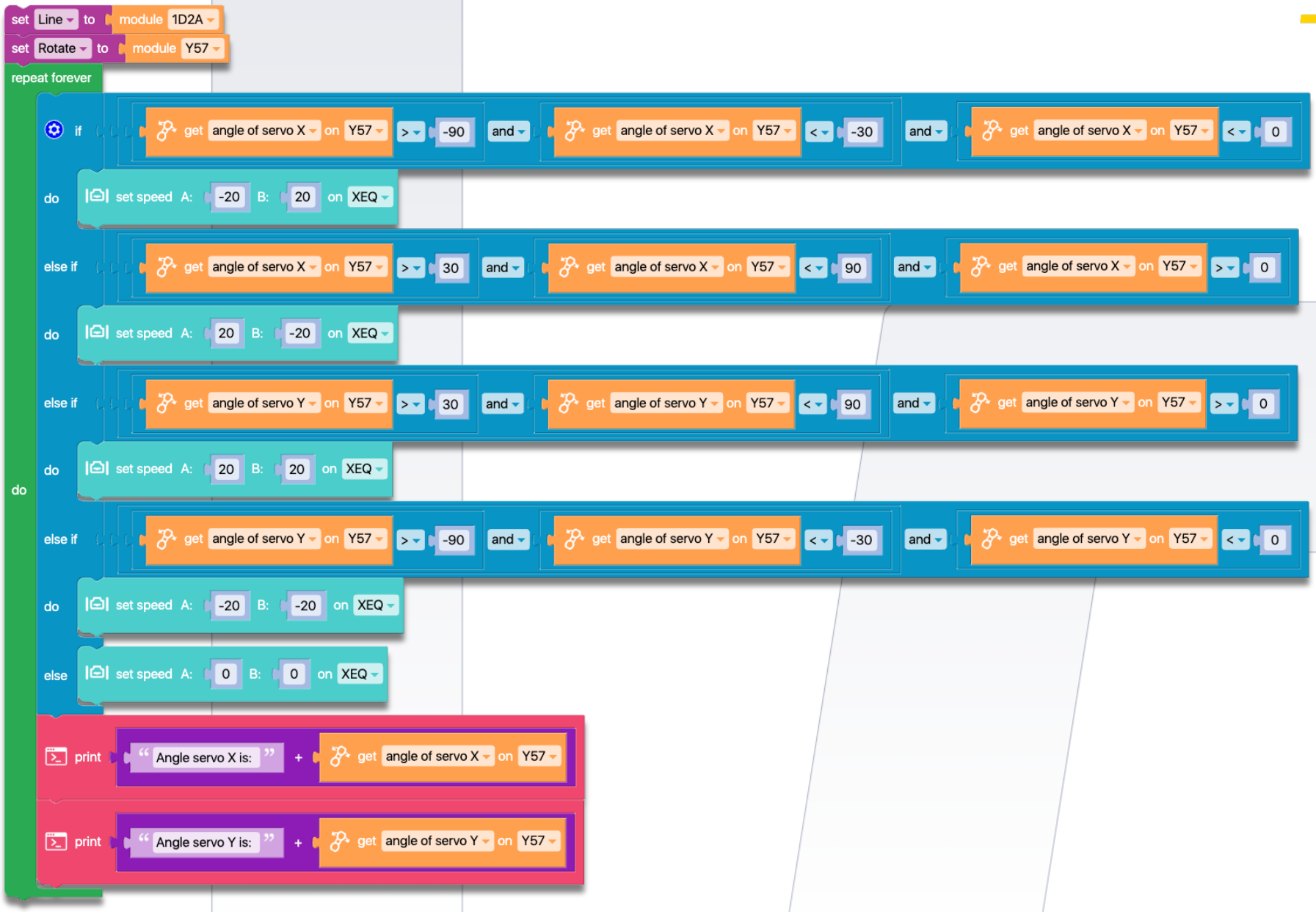
shape robotics
SMART LEARNING

Code 54:
Joint as a Joystick for Spin

- The program enables a Joint module to remotely control the movement of a Spin module sequentially, without mixing the movements. The Joint's X motor directs the Spin to move forward or backward, while the Y motor instructs it to rotate left or right. To refine control over the Spin module, there is a range of angles where no movement is considered. These angles will be displayed in the console output.

WATCH VIDEO



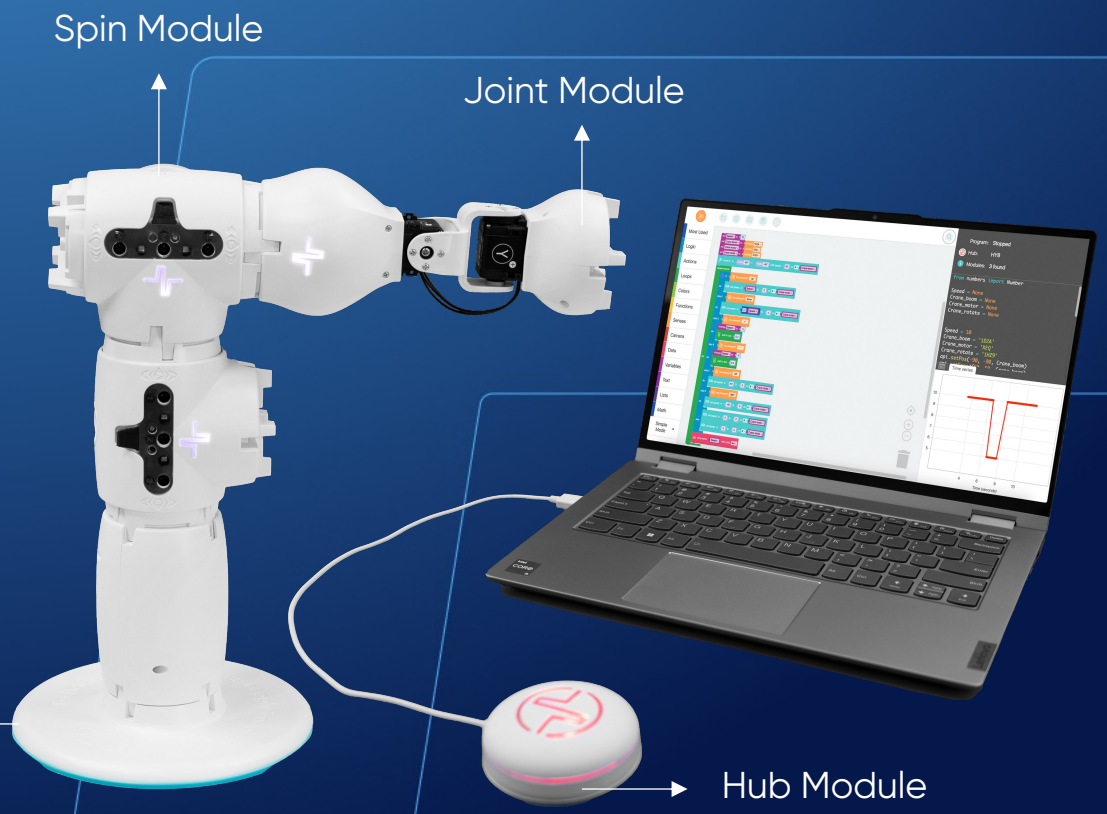

set Line to module 1D2A
set Rotate to module Y57
repeat forever
do
if get angle of servo X on Y57 > -90 and get angle of servo X on Y57 < -30 and get angle of servo X on Y57 < 0
do set speed A: -20 B: 20 on XEQ
else if get angle of servo X on Y57 > 30 and get angle of servo X on Y57 < 90 and get angle of servo X on Y57 > 0
do set speed A: 20 B: -20 on XEQ
else if get angle of servo Y on Y57 > 30 and get angle of servo Y on Y57 < 90 and get angle of servo Y on Y57 > 0
do set speed A: 20 B: 20 on XEQ
else if get angle of servo Y on Y57 > -90 and get angle of servo Y on Y57 < -30 and get angle of servo Y on Y57 < 0
do set speed A: -20 B: -20 on XEQ
else set speed A: 0 B: 0 on XEQ
print "Angle servo X is:" + get angle of servo X on Y57
print "Angle servo Y is:" + get angle of servo Y on Y57

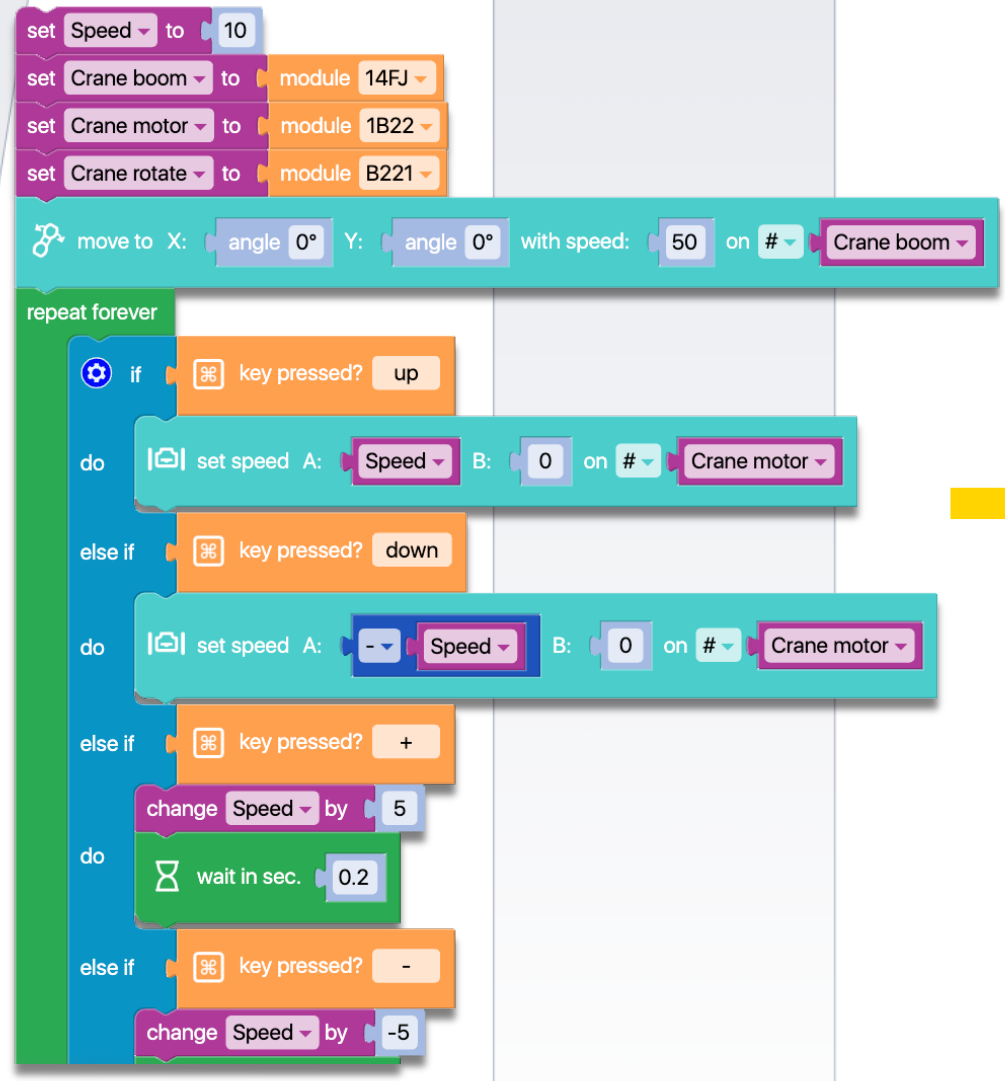
shape robotics
SMART LEARNING

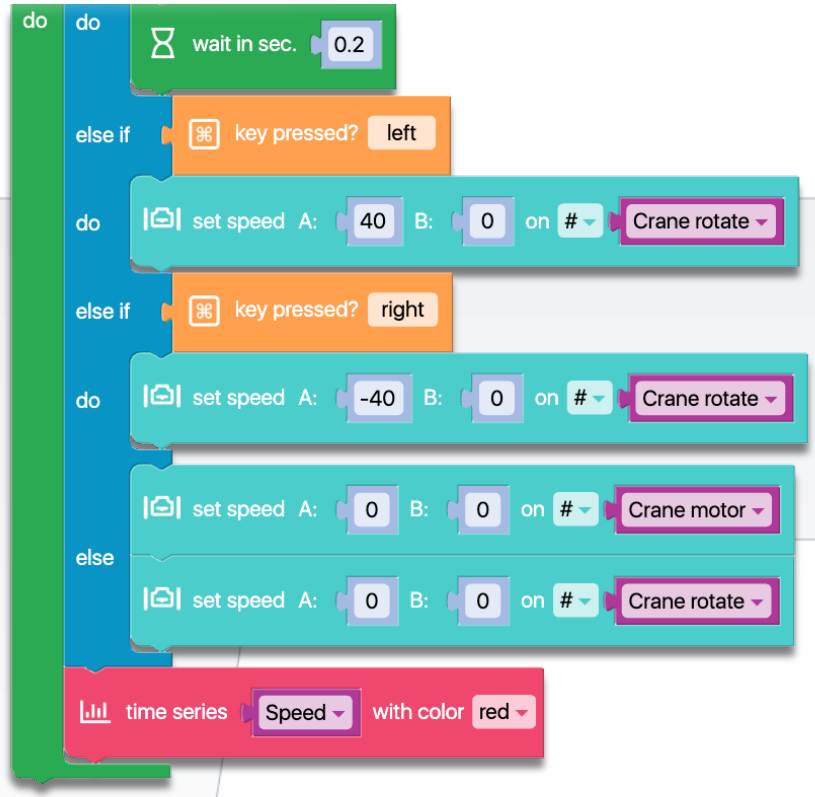
Code 55:
The Crane Simulator

- This program simulates the operation of a crane. You can use the Spin module to rotate the jib and raise or lower a weight using the up and down arrow keys, while the other Spin module rotates the entire crane using the left and right arrow keys. Additionally, you can adjust the speed of raising or lowering using the + and - keys. The graph will display the current speed value in real-time.

WATCH VIDEO




set Speed to 10
set Crane boom to module 14FJ
set Crane motor to module 1B22
set Crane rotate to module B221
move to X: angle 0° Y: angle 0° with speed: 50 on # Crane boom
repeat forever
if key pressed? up
do set speed A: Speed B: 0 on # Crane motor
else if key pressed? down
do set speed A: - Speed B: 0 on # Crane motor
else if key pressed? +
change Speed by 5
do wait in sec. 0.2
else if key pressed? -
change Speed by -5

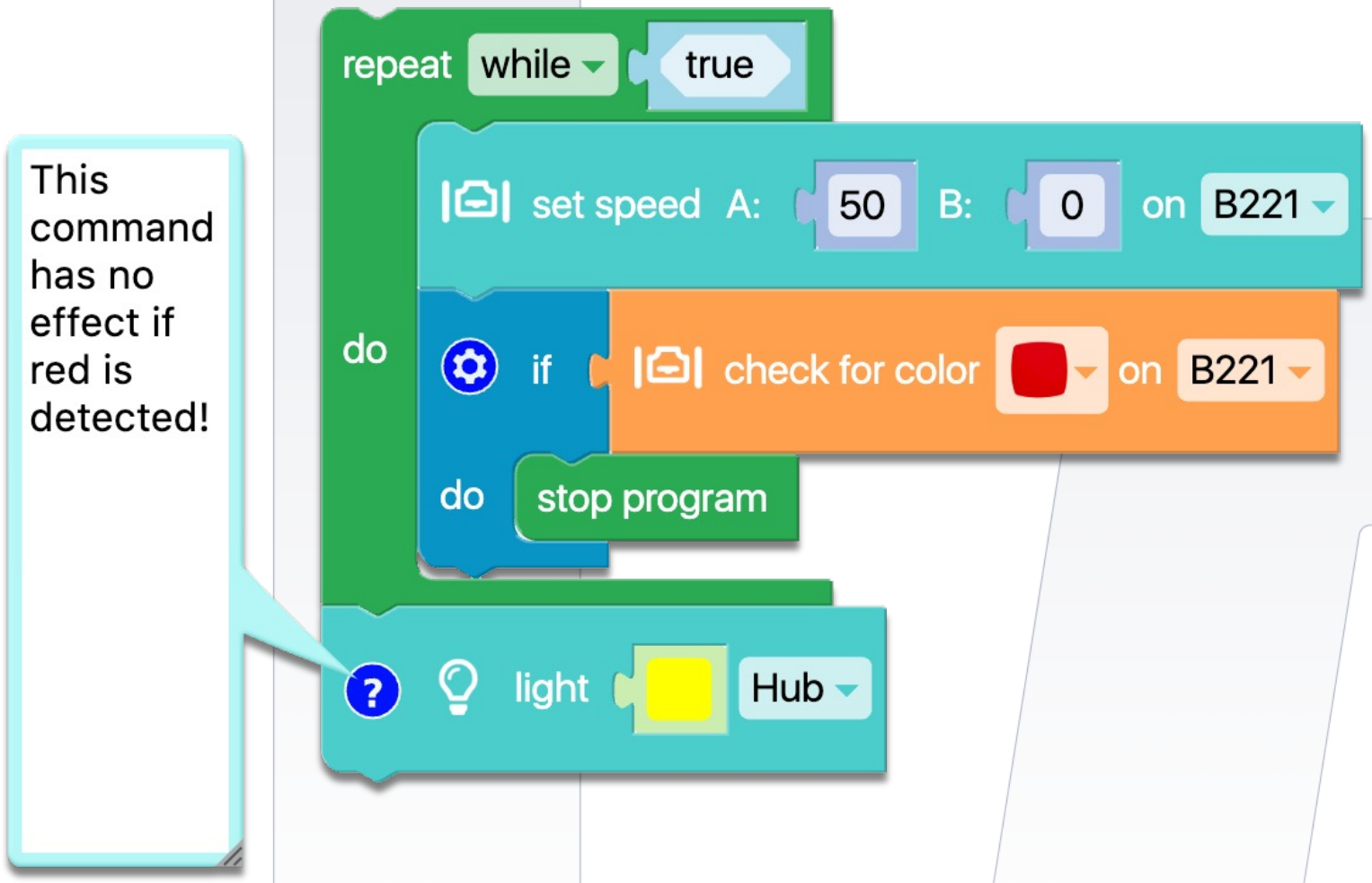

do
do wait in sec. 0.2
else if key pressed? left
do set speed A: 40 B: 0 on # Crane rotate
else if key pressed? right
do set speed A: -40 B: 0 on # Crane rotate
else
set speed A: 0 B: 0 on # Crane motor
set speed A: 0 B: 0 on # Crane rotate
time series Speed with color red

shape robotics
SMART LEARNING

Code 56:
Complete STOP of a Program

- The program sequence starts the Spin A motor, causing it to spin continuously. When Spin detects red in front of its sensors, the code triggers the Stop program command, halting the entire program. In this scenario, any subsequent Hub commands will be ignored and cannot be executed.

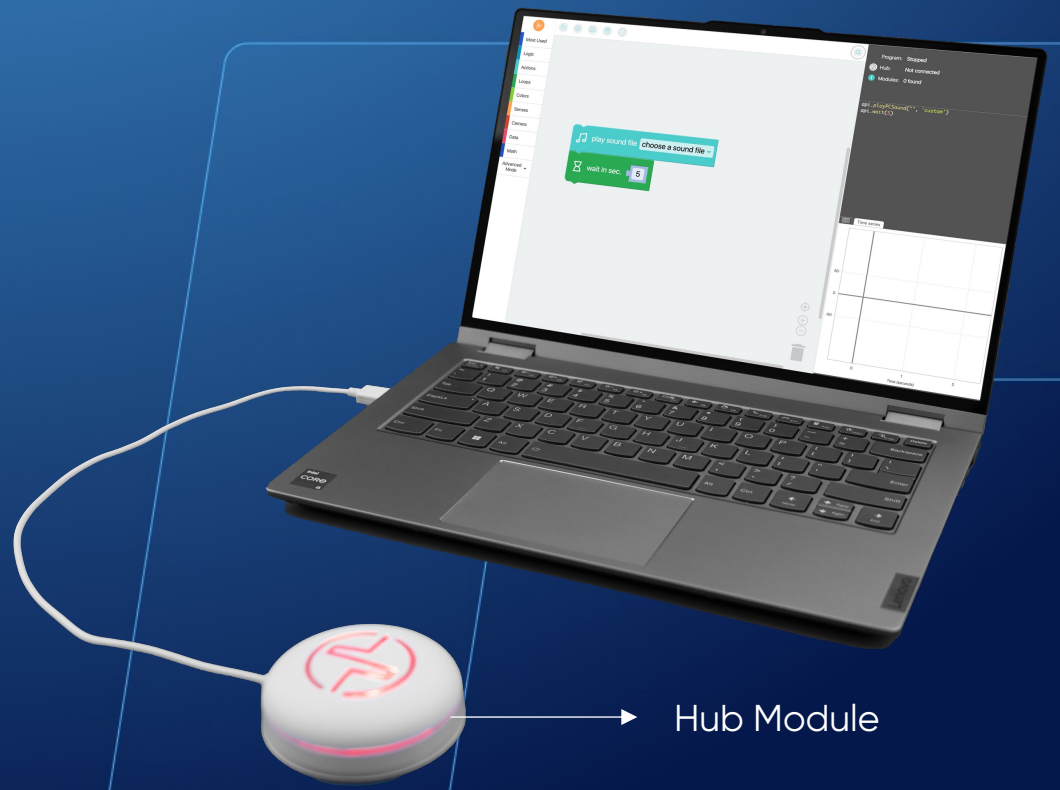
WATCH VIDEO

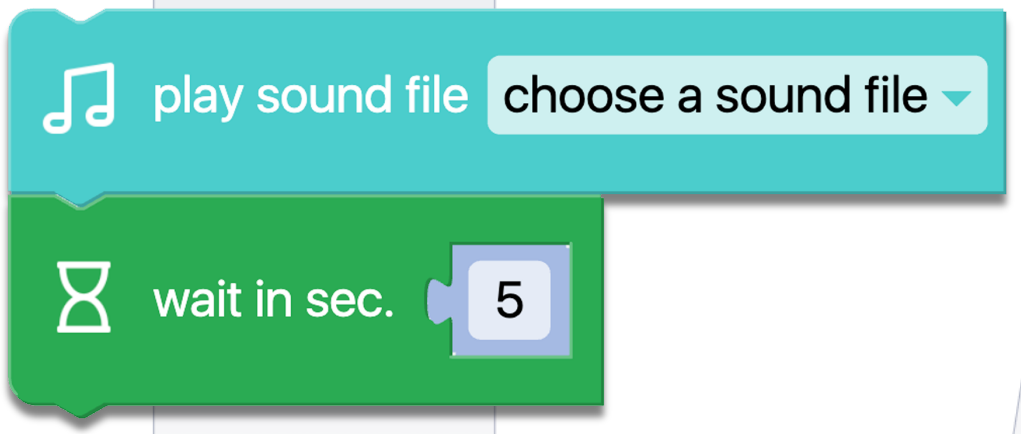

This command has no effect if red is detected!
repeat while true
set speed A: 50 B: 0 on B221
do
if check for color on B221
do
stop program
light Hub

Code 57:
Using MP3 File

- The code sequence calls an MP3 file from the computer. It's important to note that the sequence plays the audio file while the program continues with other commands. If another sound follows, a wait command is necessary to ensure the audio finishes playing before proceeding. The file must be located in Documents/Fable/My Fable Sounds.

WATCH VIDEO




play sound file
choose a sound file
wait in sec.
5

shape robotics
SMART LEARNING

Code 58:
Playing with Musical Notes

- The code sequence plays the first part of the song "Twinkle, Twinkle Little Star". The rhythm can be changed by changing the tempo in the pause controls after each note is played.

WATCH VIDEO

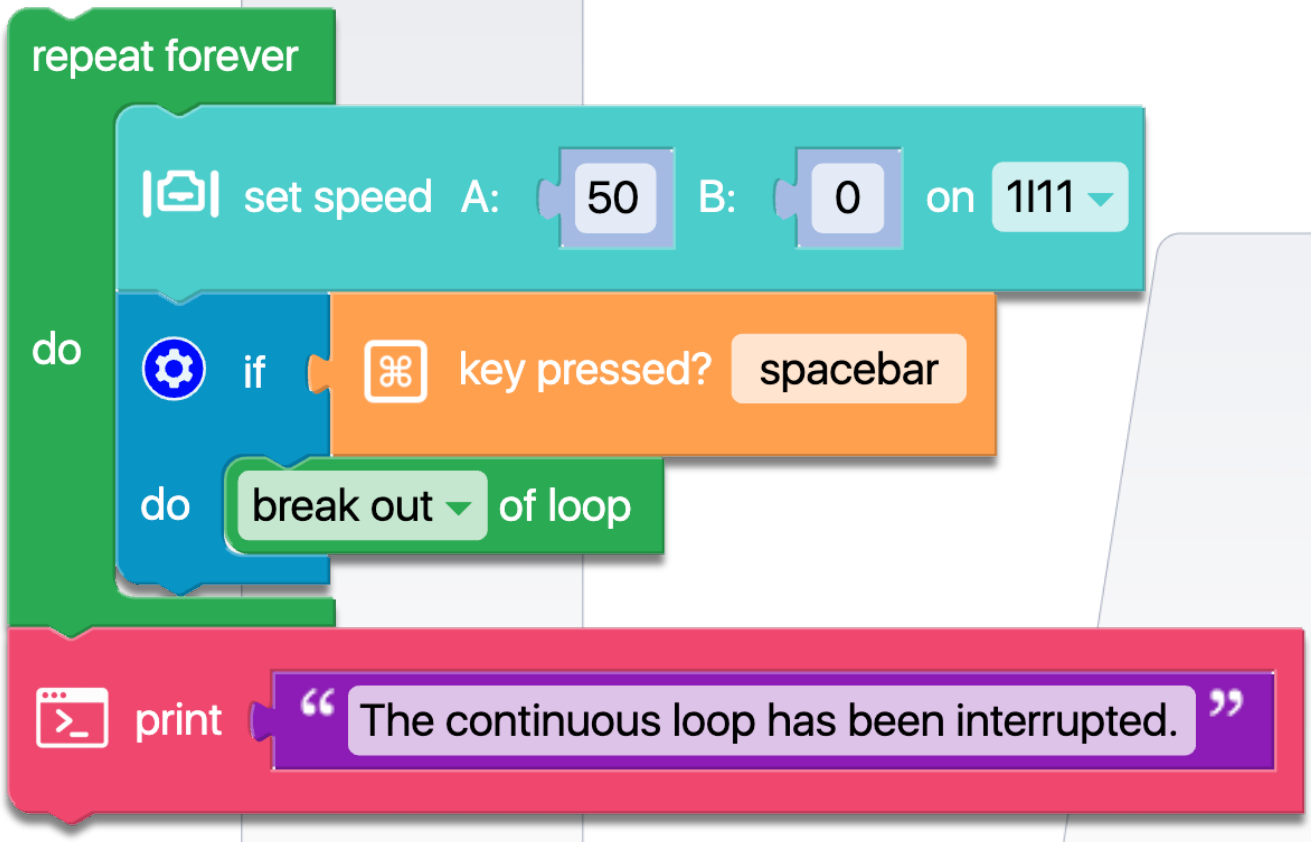
shape robotics
SMART LEARNING

Code 59:
Out of Endless Loop

- It is generally not recommended to use infinite loops (Repeat forever block) in coding. However, in certain situations, there may be a need to employ such a loop and break out of it when necessary. This can be accomplished by using the "Break out of loop" command. When this command is executed, the loop will terminate, and the subsequent commands will continue to execute in sequence after the loop.
- For instance, consider a scenario where the Spin A motor is configured to rotate continuously until the Space key is pressed. Upon pressing the Space key, the Print command is executed in the Output Console, indicating that the loop has been terminated.

WATCH VIDEO

shape robotics
SMART LEARNING

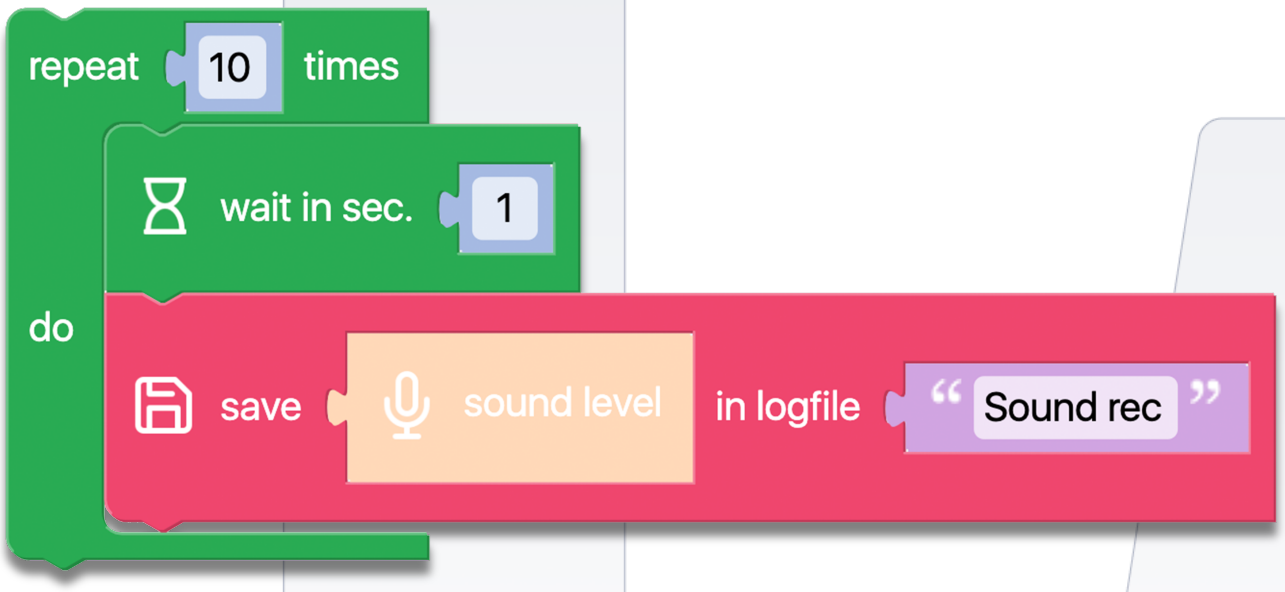


shape robotics
SMART LEARNING

Code 60:
Record in a .csv File

- The program records the noise level from the computer's microphone in a loop of 10 repetitions every second. This data is stored numerically in a CSV file with a file name specified by the programmer. The file is saved in the 'Documents' folder under the 'Fable' folder. The data stored in the CSV file can be utilized in other programs as required.

WATCH VIDEO


repeat
10
times
wait in sec.
1
do
save
sound level
in logfile
“
Sound rec
”

shape robotics
SMART LEARNING

Code 61:
Read from a .csv File

- The program reads data from a .csv file named "Sound rec". It then displays this data in the Output Console. The data can be processed and used for other purposes as well. To provide an example, we have used the file saved from programming “Record in a .csv File”.

WATCH VIDEO

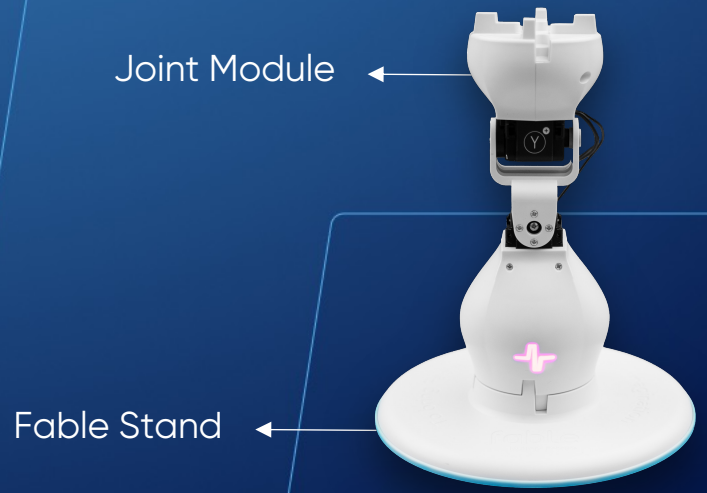

print
read from csv
Sound rec

shape robotics
SMART LEARNING

Code 62:
Record multiple data in a .csv file

- Upon launching the program, a file with the .csv extension will be automatically generated(you can set the file name). This file stores various values such as the X and Y motor angles of the Joint mode that is connected to the Hub, the battery level of the Joint mode, and the time elapsed since the program was started.

WATCH VIDEO



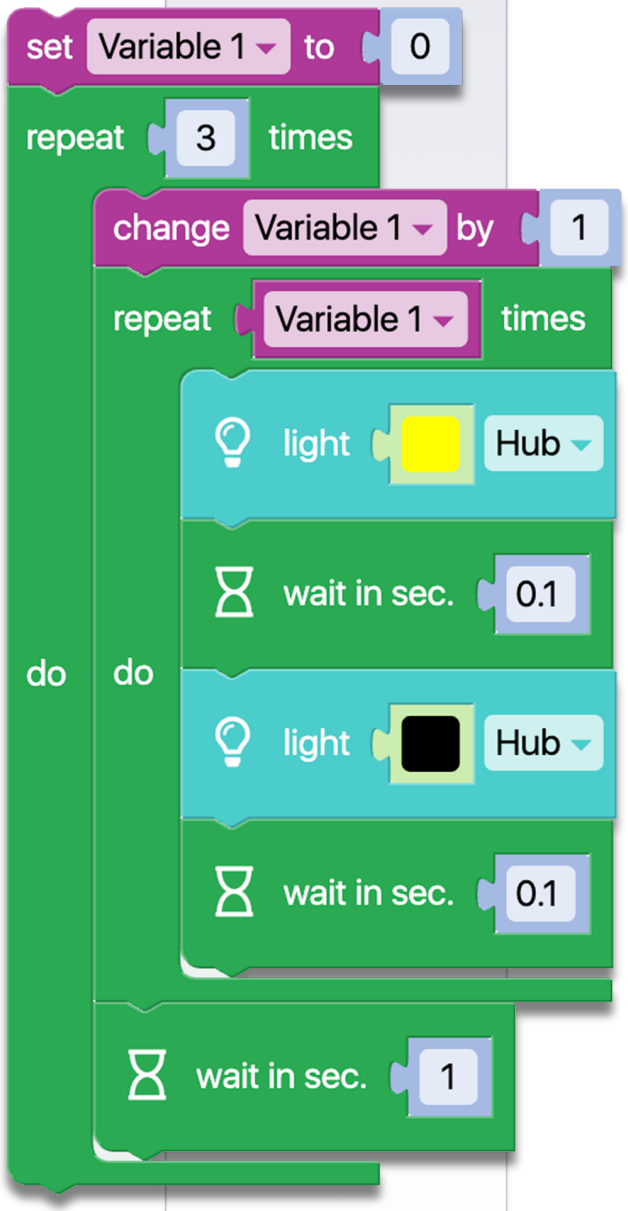

repeat while true
move to X: angle -15° Y: angle 15° on #
save create list with time in sec.
get angle of servo X on #
get angle of servo Y on #
battery level #
in logfile Fable-log.csv
wait in sec. 1
do
move to X: angle 15° Y: angle -15° on #
save create list with time in sec.
get angle of servo X on #
get angle of servo Y on #
battery level #
in logfile Fable-log.csv
wait in sec. 1

shape robotics
SMART LEARNING

Code 63:
Create a Variable

- To create a variable in Advance Mode, we can follow these steps: Click on Variables and then on Create Variables. Once we have done this, we can name our variable. For example, we can name it Variable 1.
- At the beginning of the program, Variable 1 has a numeric value of zero. However, during a loop of three repetitions, the variable's value will change by adding a unit. Moreover, the Hub will light up yellow with the updated value.
- After the variable's value has been updated three times, the loop will be exited due to the three repetitions.

WATCH VIDEO

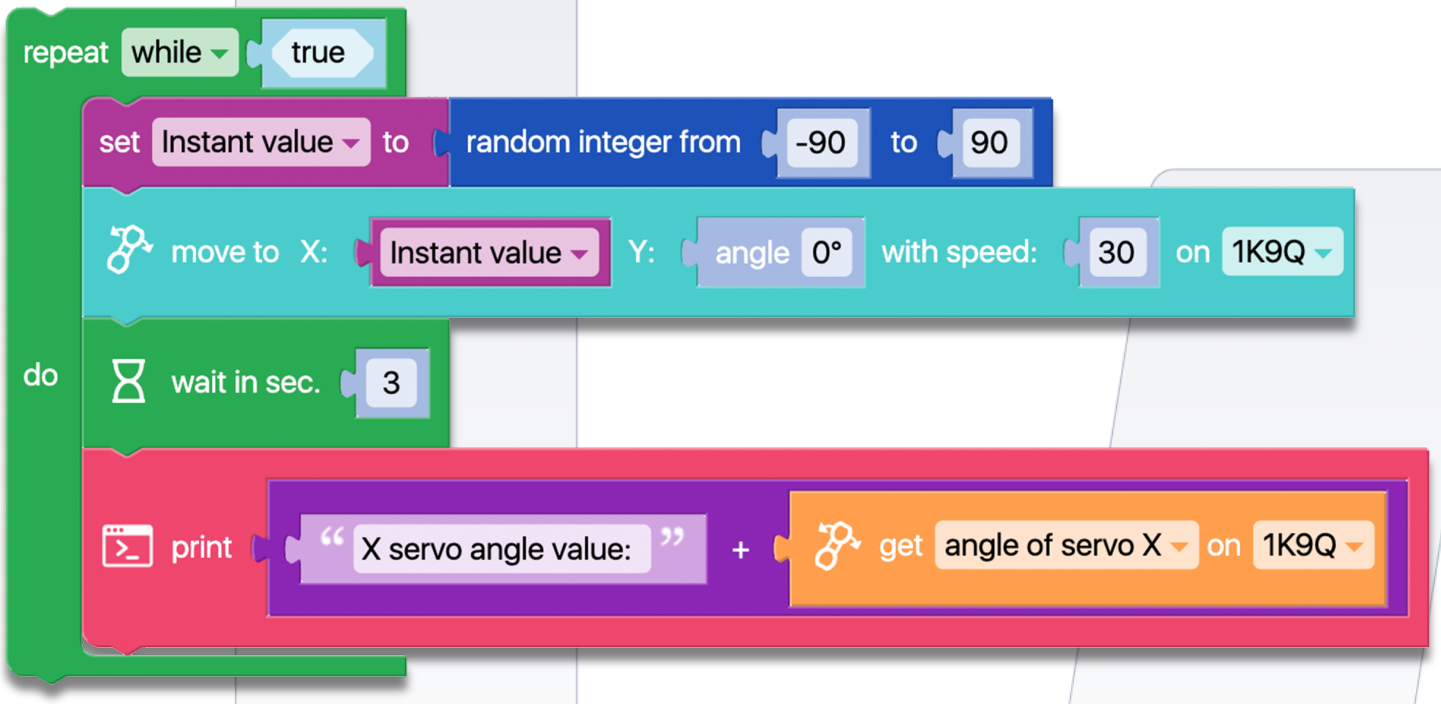

set Variable 1 to 0
repeat 3 times
change Variable 1 by 1
repeat Variable 1 times
light Hub
wait in sec. 0.1
do
do
light Hub
wait in sec. 0.1
wait in sec. 1

Code 64:
Generating random values for Joint's servomotor

- The program generates a random number between -90 and 90, representing the range of movement of the X servo motor on the Joint module. This random number is stored in the variable "Instant value" and displayed in the Output console. A new movement occurs every 3 seconds.

WATCH VIDEO

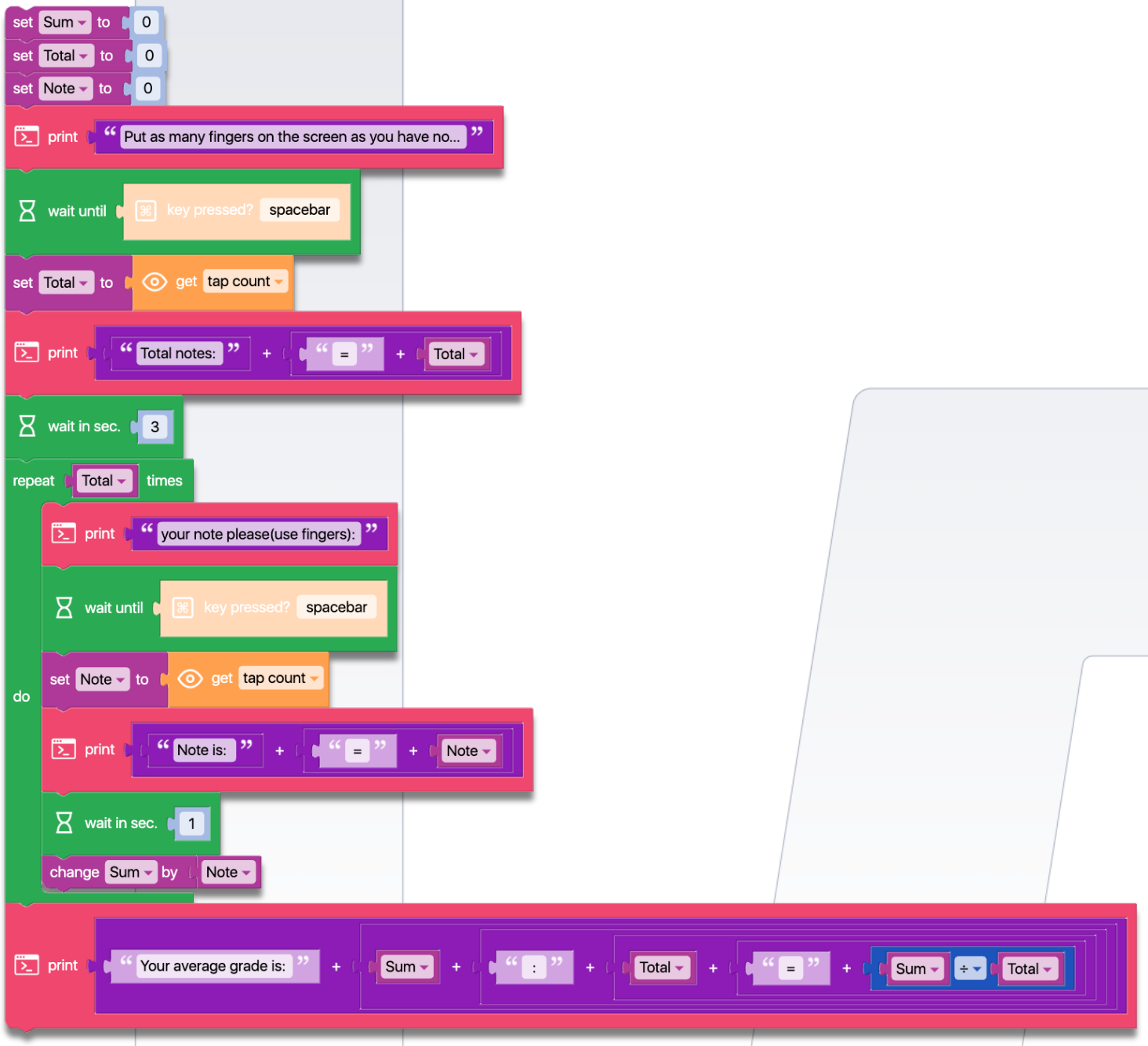
shape robotics
SMART LEARNING



Code 65:
Calculating the Arithmetic Average Grade

- The program is designed to calculate the arithmetic average of grades. Grades, ranging from 1 to 5, are inputted into the program by tapping on the Fable Face application screen on your phone. For instance, if three fingers are placed on the screen, the program registers the value 3. This method is utilized for inputting both the total number of grades and each individual grade.

WATCH VIDEO


set Sum to 0
set Total to 0
set Note to 0
print Put as many fingers on the screen as you have no...
wait until key pressed? spacebar
set Total to get tap count
print Total notes: + = + Total
wait in sec. 3
repeat Total times
do
print your note please(use fingers):
wait until key pressed? spacebar
set Note to get tap count
print Note is: + = + Note
wait in sec. 1
change Sum by Note
print Your average grade is: + Sum + : + Total + = + Sum ÷ Total

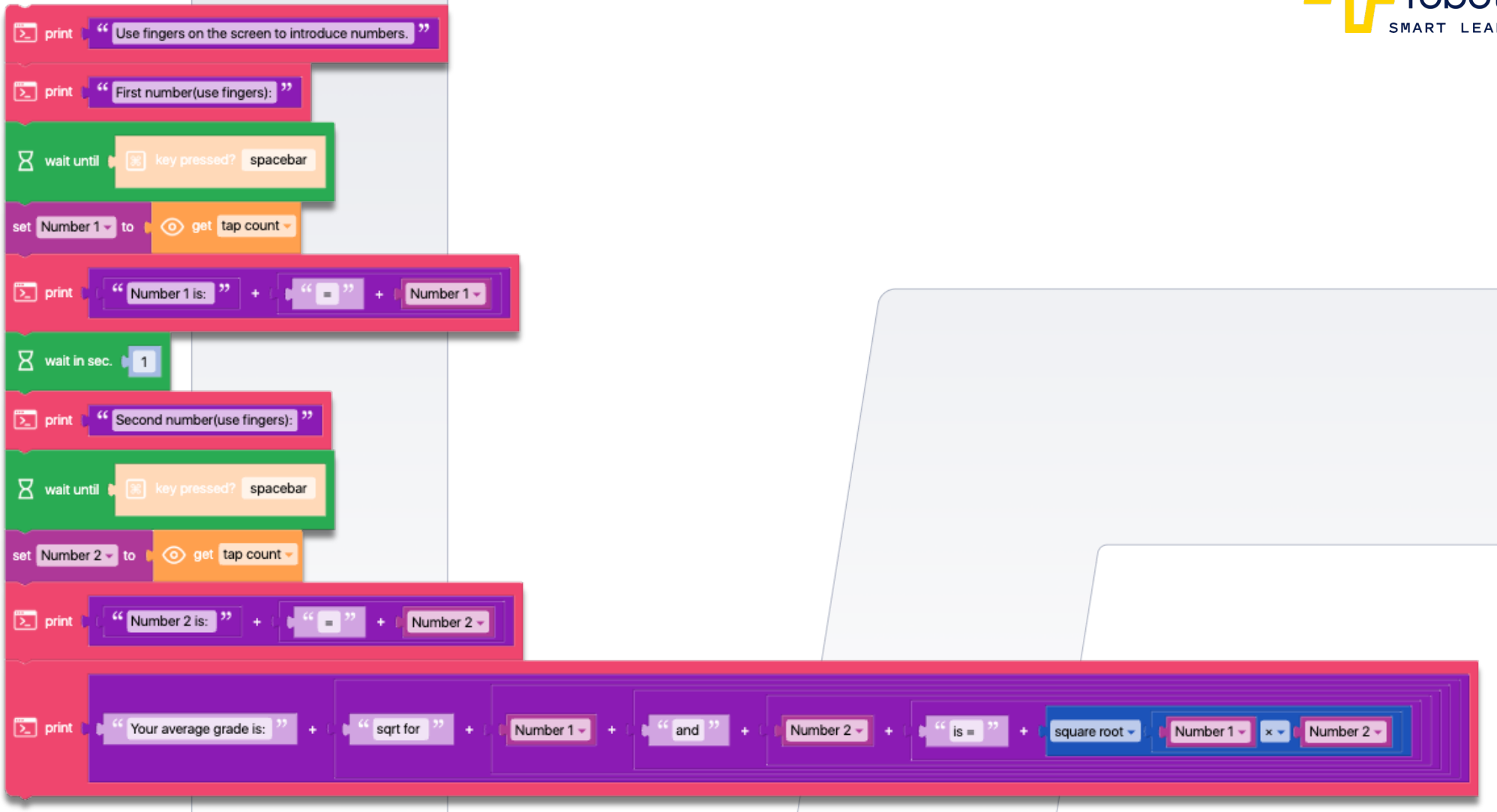
shape robotics
SMART LEARNING

Code 66:
Calculating the Geometric Mean for Two Numbers

- The program reads two numbers from the phone screen, calculates and displays the geometric mean for them.

WATCH VIDEO

shape robotics
SMART LEARNING

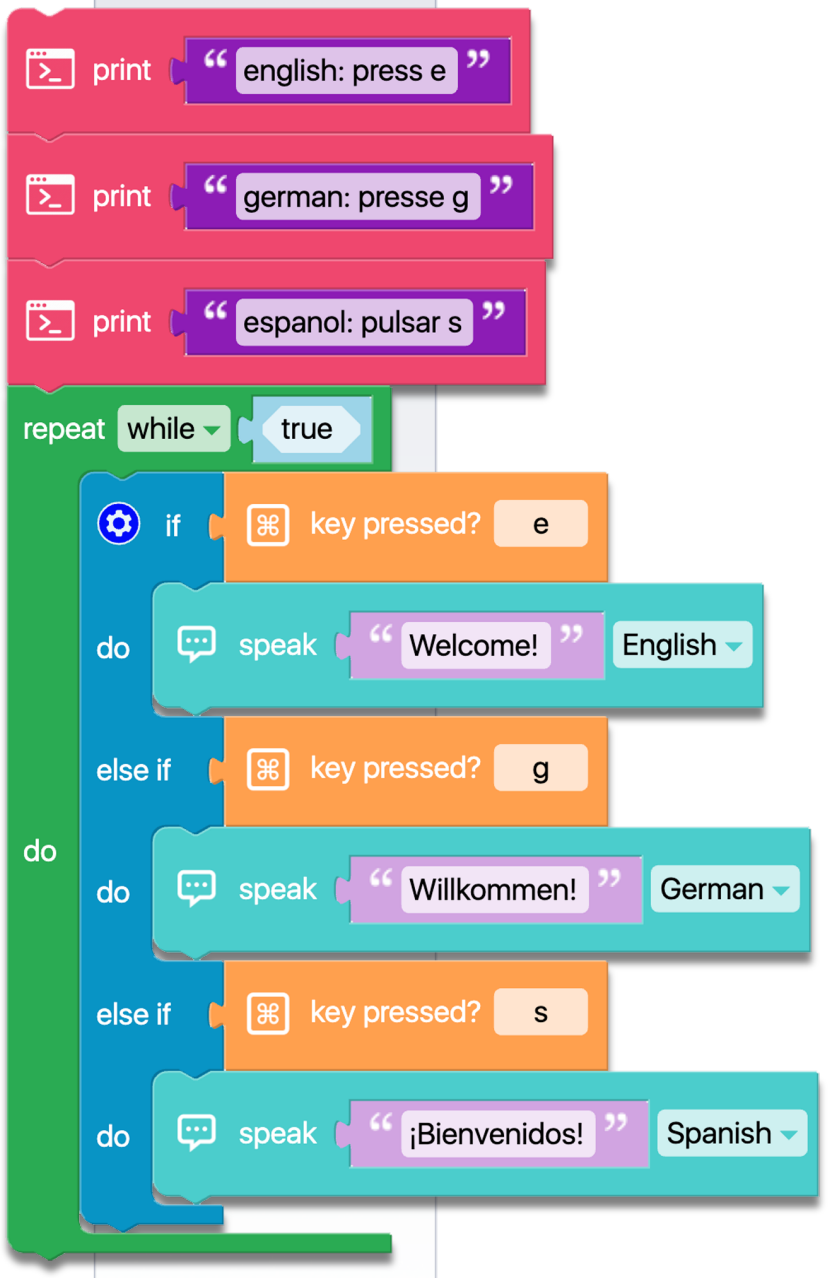


shape robotics
SMART LEARNING

Code 67:
Converting Text to Speech

- The code sequence processes a text and converts it into speech. Users have the option to select the language, and the resulting speech will reflect the accent corresponding to that language. This code can be useful for sending audio messages or for various other applications requiring text-to-speech functionality.

WATCH VIDEO

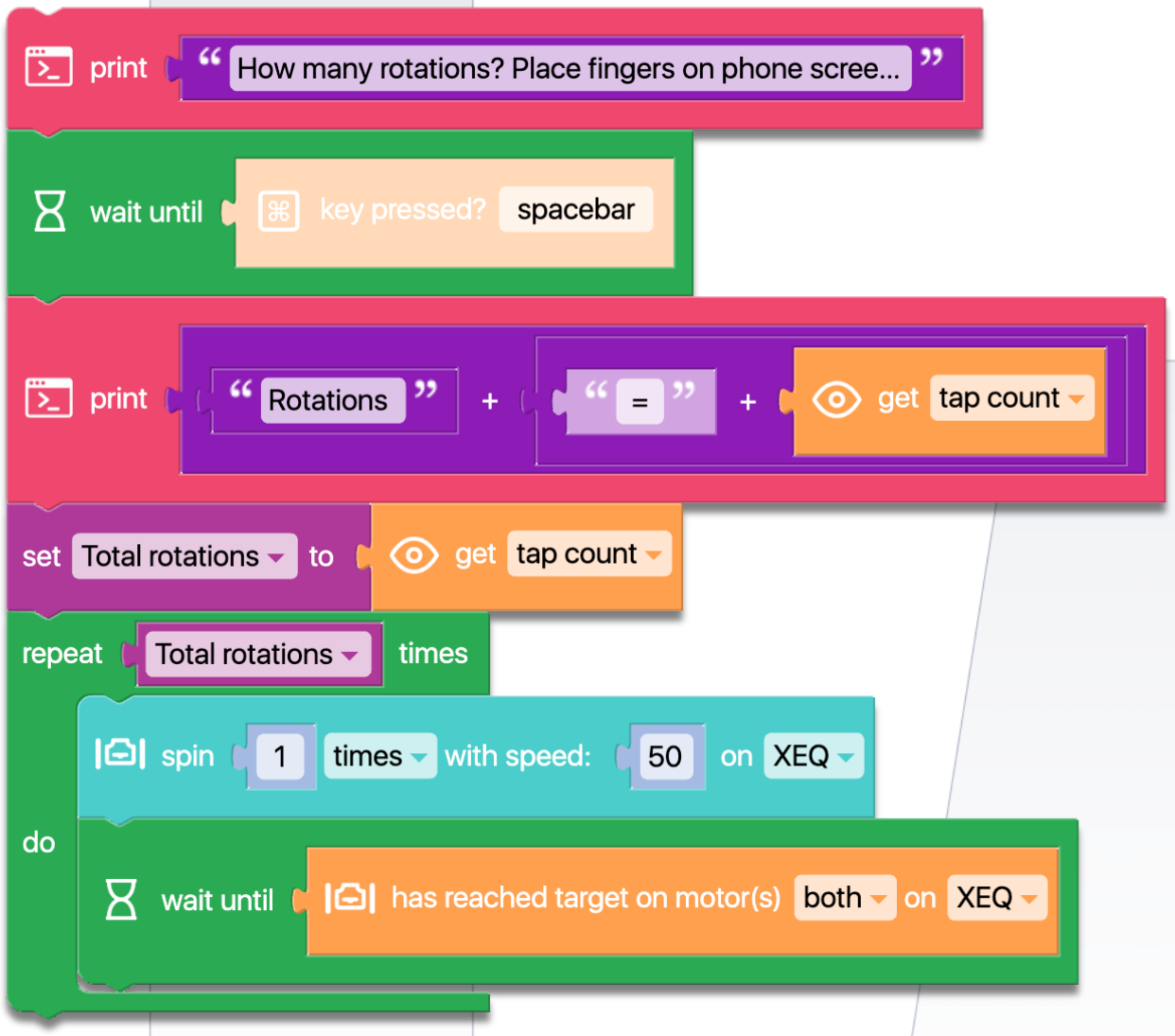

print
english: press e
print
german: presse g
print
espanol: pulsar s
repeat
while
true
do
if
key pressed?
e
do
speak
Welcome!
English
else if
key pressed?
g
do
speak
Willkommen!
German
else if
key pressed?
s
do
speak
¡Bienvenidos!
Spanish

shape robotics
SMART LEARNING

Code 68:
Read Numbers from the Phone Screen

- The program is designed to read numbers from the phone screen (with the Fable Face app open). When two fingers touch the screen and the SPACE key is pressed, the program utilizes this number to rotate a Spin module as many times as the number entered, in this instance, two times. This method of entering numbers via the phone can serve as an input keypad. Feel free to try with other numbers.

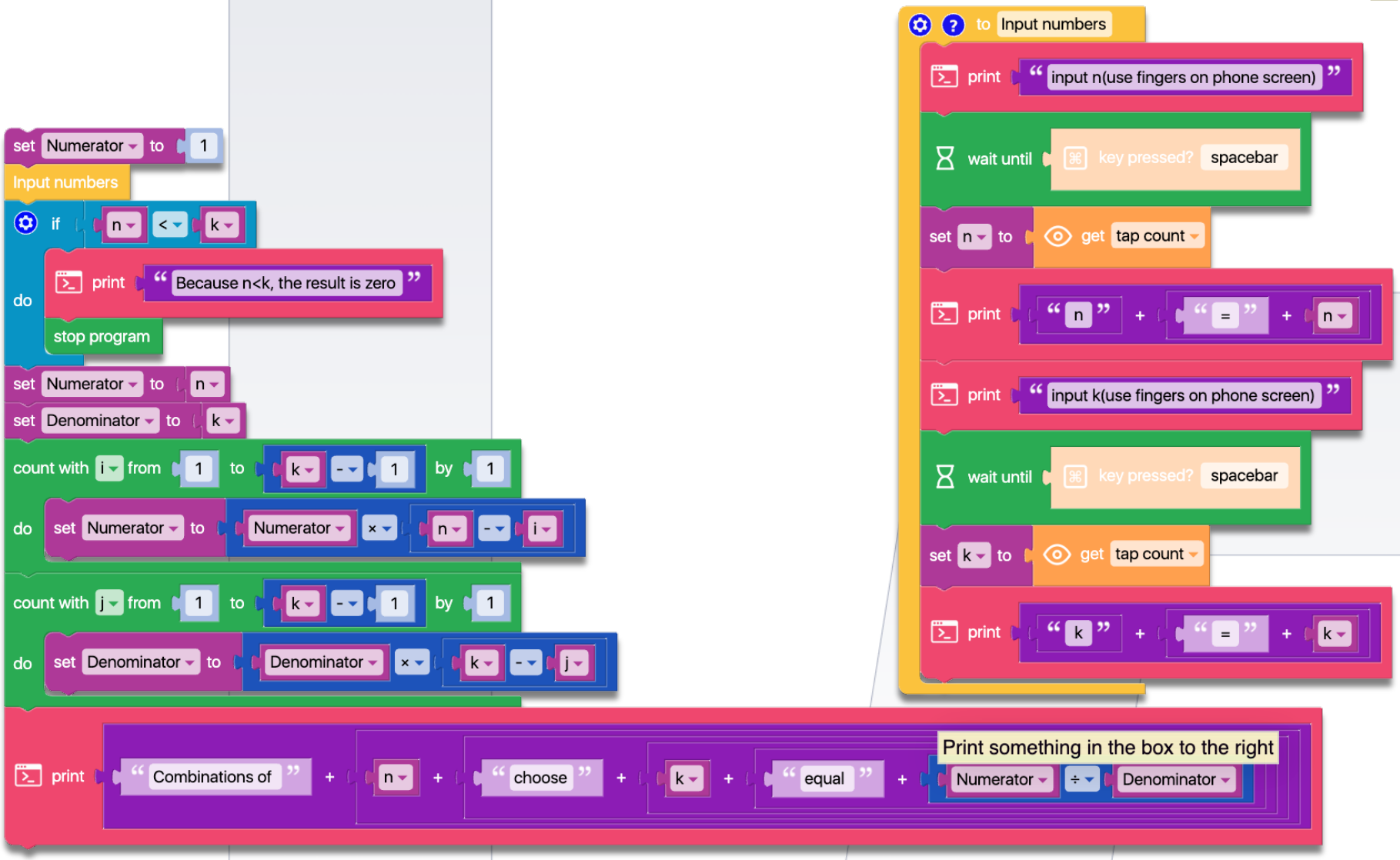
WATCH VIDEO


print
How many rotations? Place fingers on phone scree...
wait until
key pressed?
spacebar
print
Rotations
+
=
+
get
tap count
set
Total rotations
to
get
tap count
repeat
Total rotations
times
do
spin
1
times
with speed:
50
on
XEQ
wait until
has reached target on motor(s)
both
on
XEQ

Code 69:
Read Numbers from the Phone Screen n choose k formula

- The program uses Fable Face to read two values that are entered by detecting the number of fingers touching the screen. These values are then used to calculate combinations of n taken as k, where n and k are the numbers entered via the phone. The result is displayed in the Output Console.

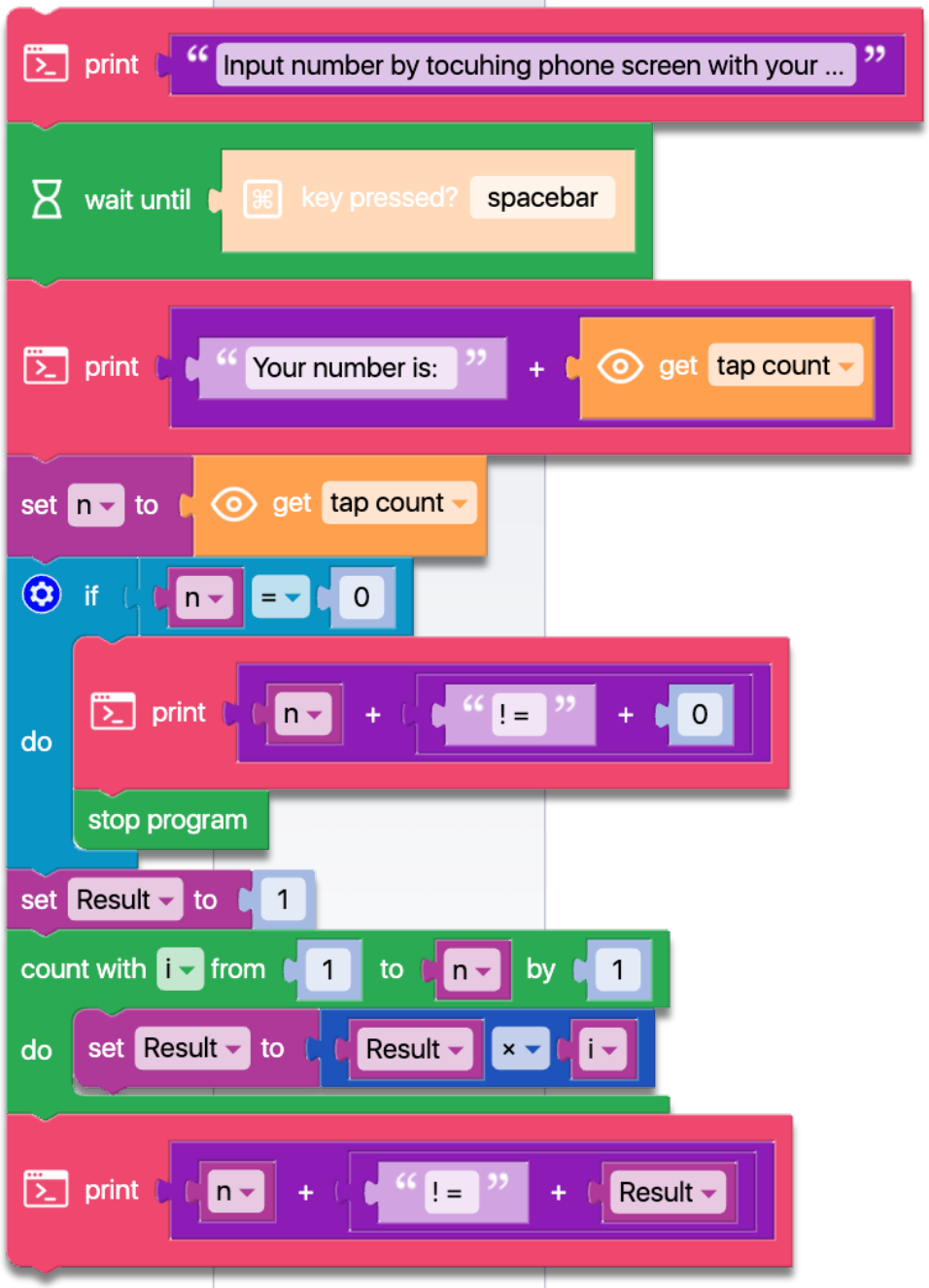
WATCH VIDEO


set Numerator to 1
Input numbers
if n < k
do print "Because n<k, the result is zero"
stop program
set Numerator to n
set Denominator to k
count with i from 1 to k - 1 by 1
do set Numerator to Numerator × n - i
count with j from 1 to k - 1 by 1
do set Denominator to Denominator × k - j
print "Combinations of" + n + "choose" + k + "equal" + Numerator ÷ Denominator
Print something in the box to the right
to Input numbers
print "input n(use fingers on phone screen)"
wait until key pressed? spacebar
set n to get tap count
print "n" + "=" + n
print "input k(use fingers on phone screen)"
wait until key pressed? spacebar
set k to get tap count
print "k" + "=" + k

Code 70:
Read Numbers from the Phone Screen n! formula

- The program calculates n! by detecting the finger count on the Fable Face screen. Both the input data and the final result are displayed on the output console.

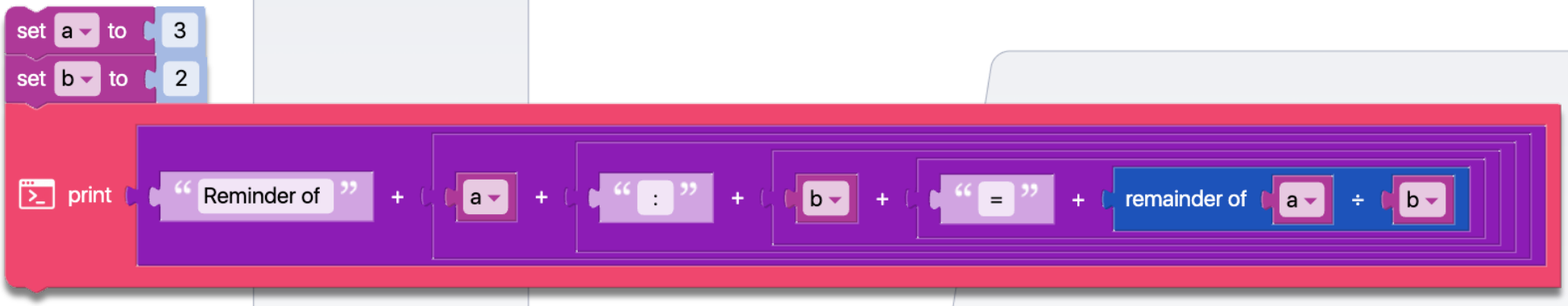
WATCH VIDEO


print
Input number by tocuhing phone screen with your ...
wait until
key pressed?
spacebar
print
Your number is:
+
get
tap count
set
n
to
get
tap count
if
n
=
0
do
print
n
+
! =
+
0
stop program
set
Result
to
1
count with
i
from
1
to
n
by
1
do
set
Result
to
Result
×
i
print
n
+
! =
+
Result

Code 71:
Calculate the Reminder of the Division Method 1

- The program sequence requires two numerical data inputs and calculates the remainder of the division of the first number by the second number. The initial numeric data must be entered manually at the beginning of the sequence. However, in a larger program, these inputs can be sourced from sensors or other calculations performed in preceding lines of code. The result is displayed in the Output console.

WATCH VIDEO

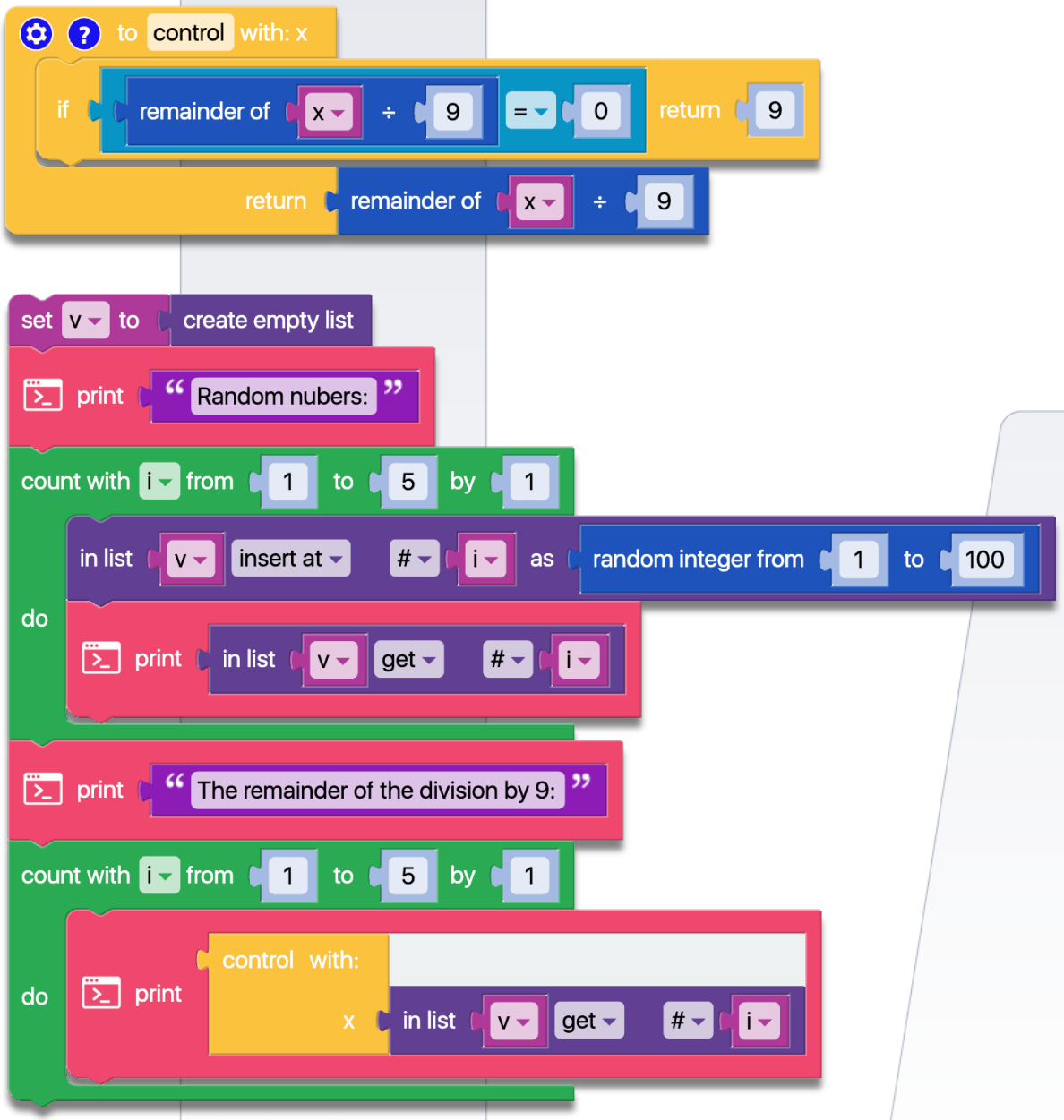

set a to 3
set b to 2
print
Reminder of
+
a
+
:
+
b
+
=
+
remainder of
a
÷
b

shape robotics
SMART LEARNING

Code 72:
Calculate the Reminder of the Division Method 2

- The program utilizes a return function. Five random numbers within the range [1,100] are generated. Each number is then checked to determine if it is divisible by 9, and the remainder of the division is stored. The remainder values are saved in a list and then displayed in the Output Console.

WATCH VIDEO

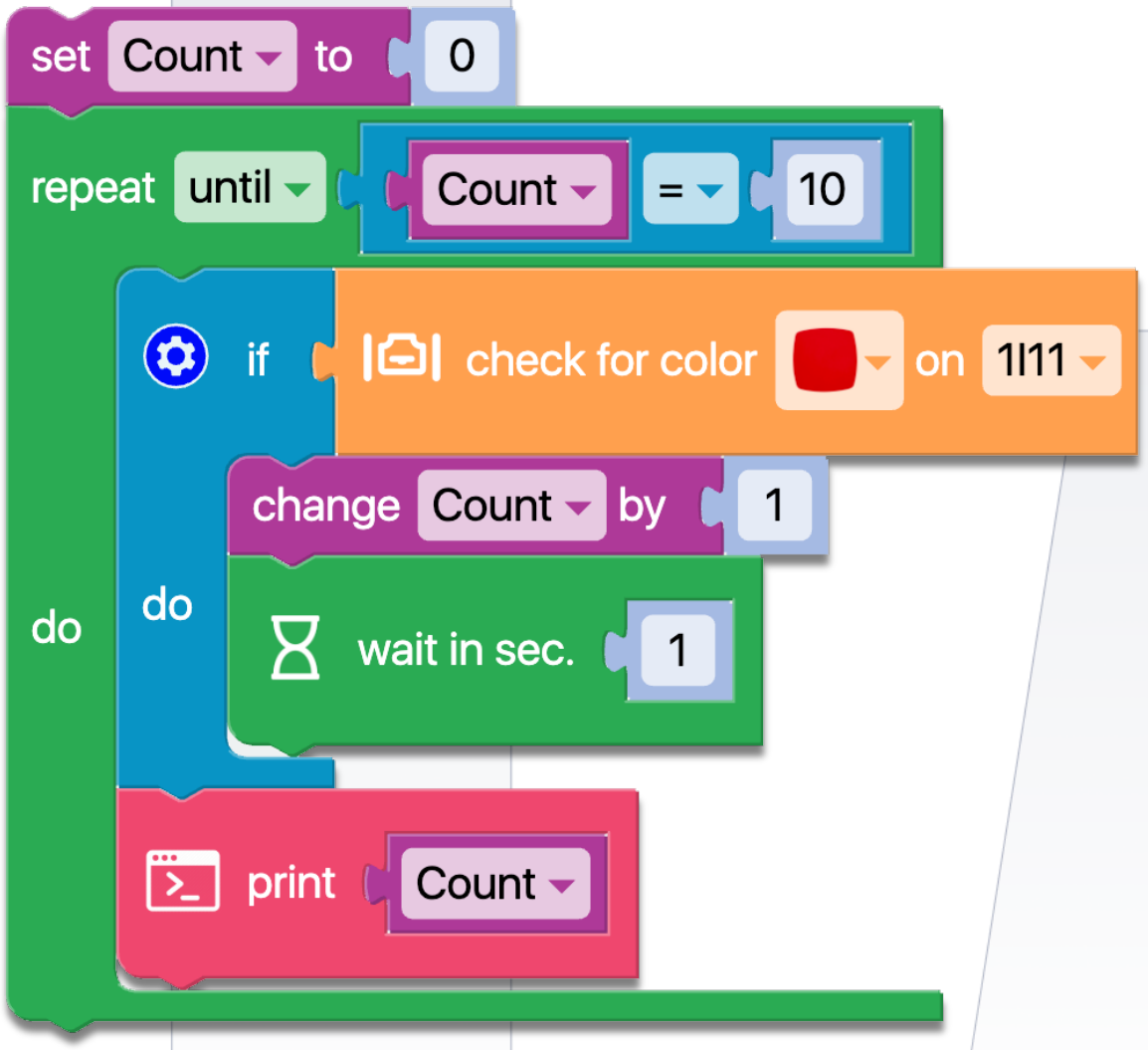

to control with: x
if remainder of x ÷ 9 = 0 return 9
return remainder of x ÷ 9
set v to create empty list
print "Random nubers:"
count with i from 1 to 5 by 1
do
in list v insert at # i as random integer from 1 to 100
print in list v get # i
print "The remainder of the division by 9:"
count with i from 1 to 5 by 1
do
print control with: x in list v get # i

shape robotics
SMART LEARNING

Code 73:
Calculating Occurrences

- This code sequence is designed to count the occurrences of the color red in front of the Spin module. It will display a counter and stop once the number of occurrences reaches ten. This sequence is useful for store management programs and can be employed to monitor stock levels. It is capable of both adding to and subtracting from inventory, as well as setting alerts for when stock levels fall below a certain threshold.

WATCH VIDEO


set Count to 0
repeat until Count = 10
if check for color on 1l11
change Count by 1
do
wait in sec. 1
do
print Count

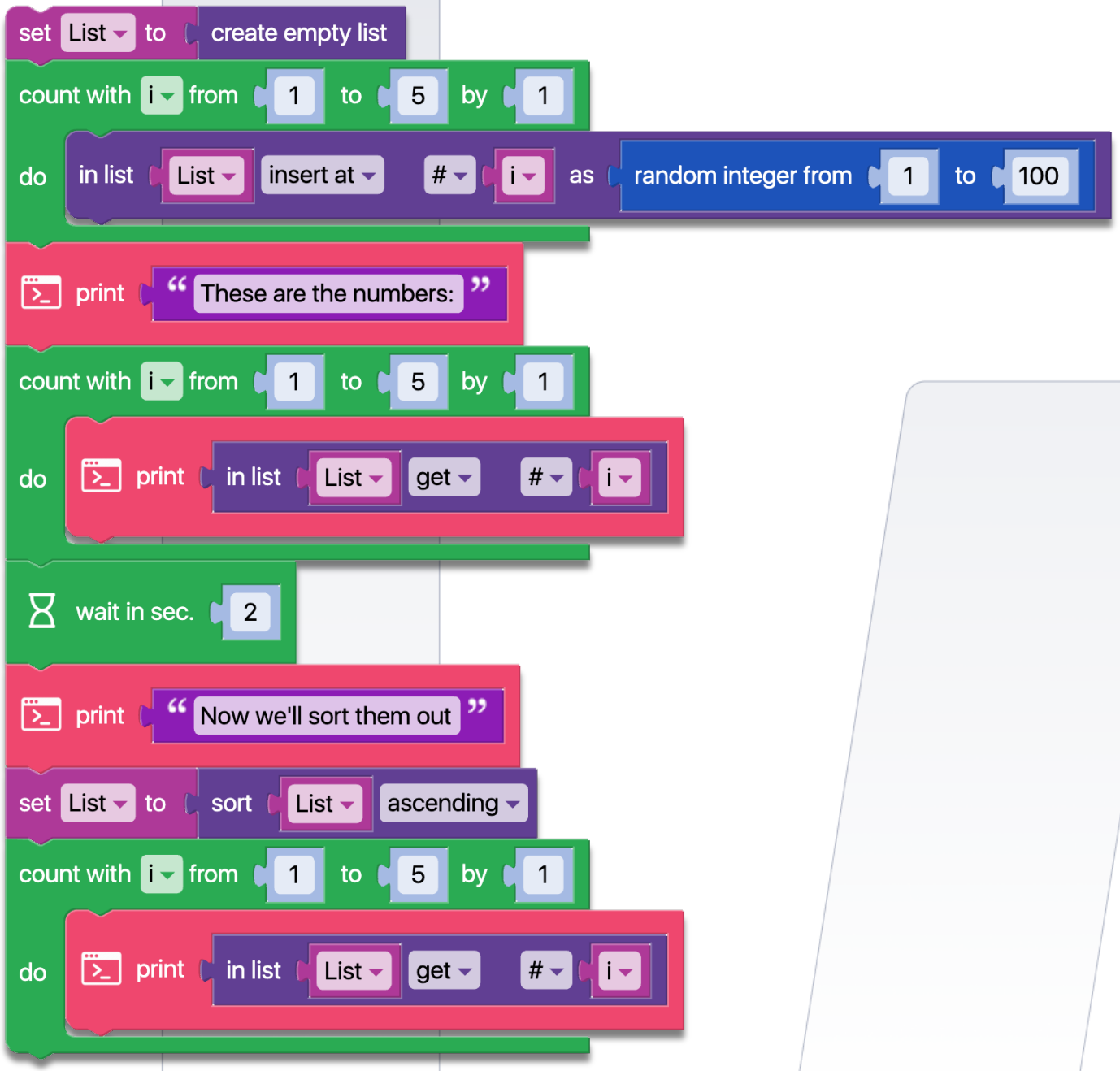
shape robotics
SMART LEARNING

Code 74:
Sorting numbers in Ascending Order

- The program generates 5 random numbers, adds them to a list and sorts them in ascending order.

WATCH VIDEO

Hub Module



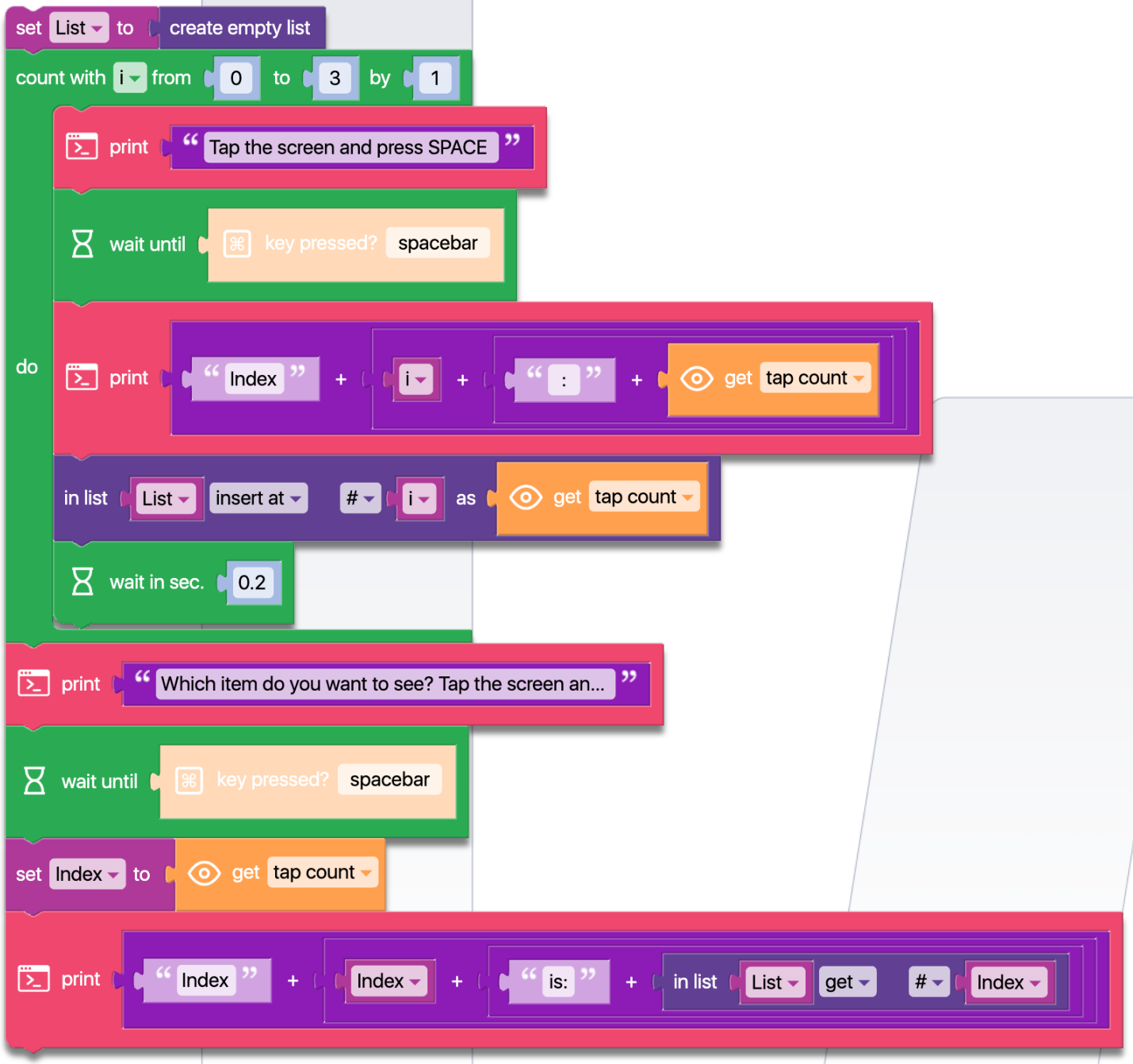
shape robotics
SMART LEARNING

shape robotics
SMART LEARNING

Code 75:
Introducing data in a specific order (a List)

- The code sequence creates a list of four items: index 0, index 1, index 2, index 3. The values for each element (index) are entered using Fable Face by counting how many fingers are on the screen. When asked "Which item do you want to see?", we also use Fable Face to choose the index we want to display, which is printed in the Output Console.

WATCH VIDEO



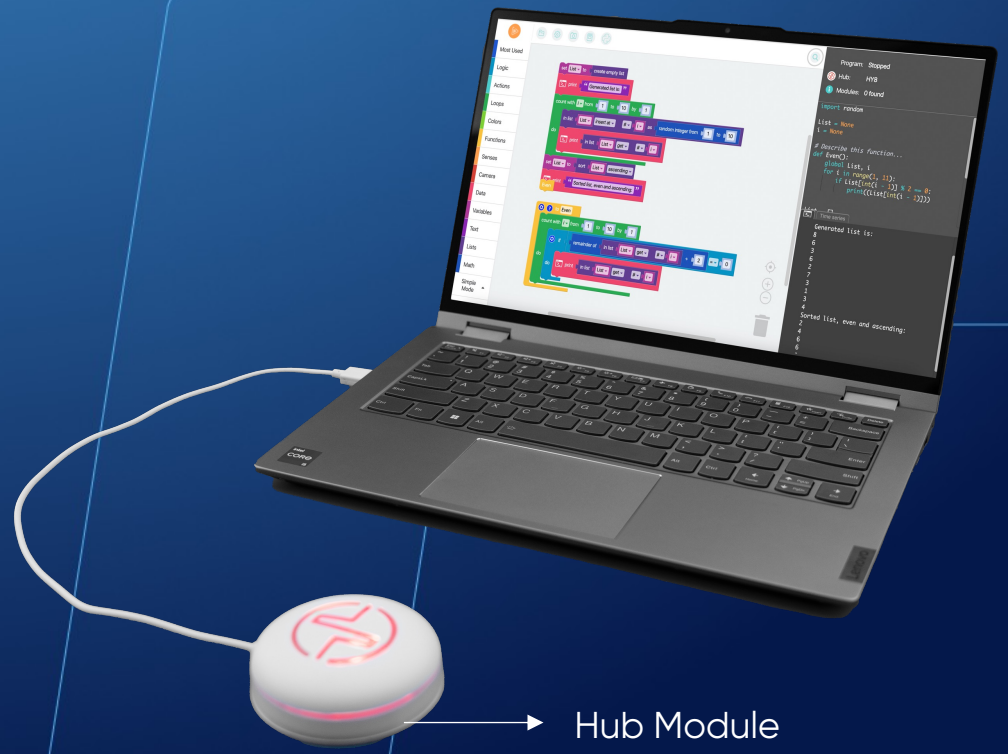
shape robotics
SMART LEARNING

shape robotics
SMART LEARNING

Code 76:
Sorting Even numbers in Ascending order

- The program generates a list of 10 elements, recording randomly generated numbers between 1 and 10. The list items are sorted in ascending order, and then a function is called to extract only even numbers by checking the remainder of division by 2. Finally, the numbers are displayed in the Output Console.

WATCH VIDEO



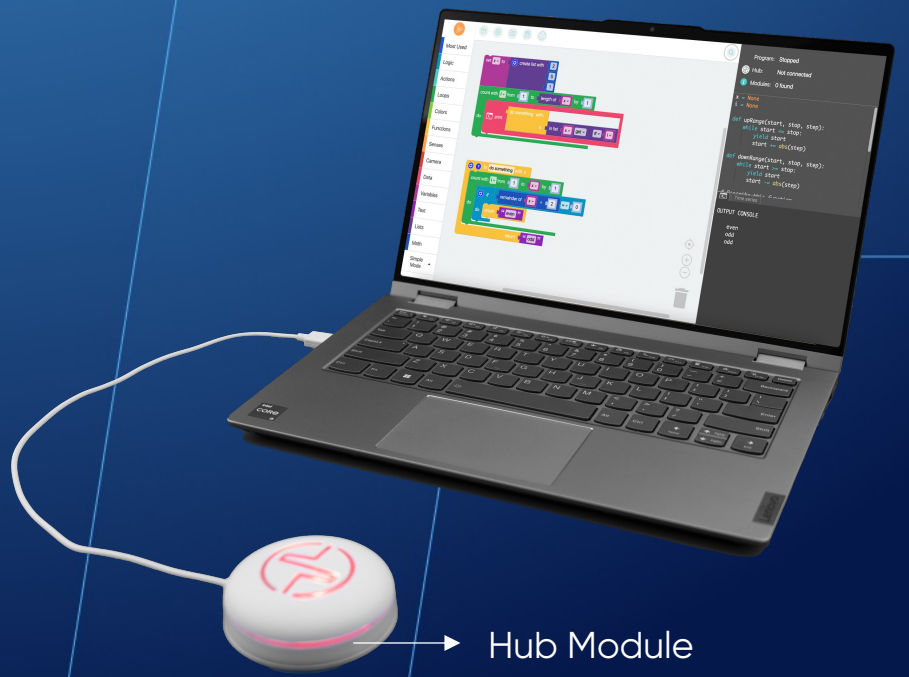
shape robotics
SMART LEARNING

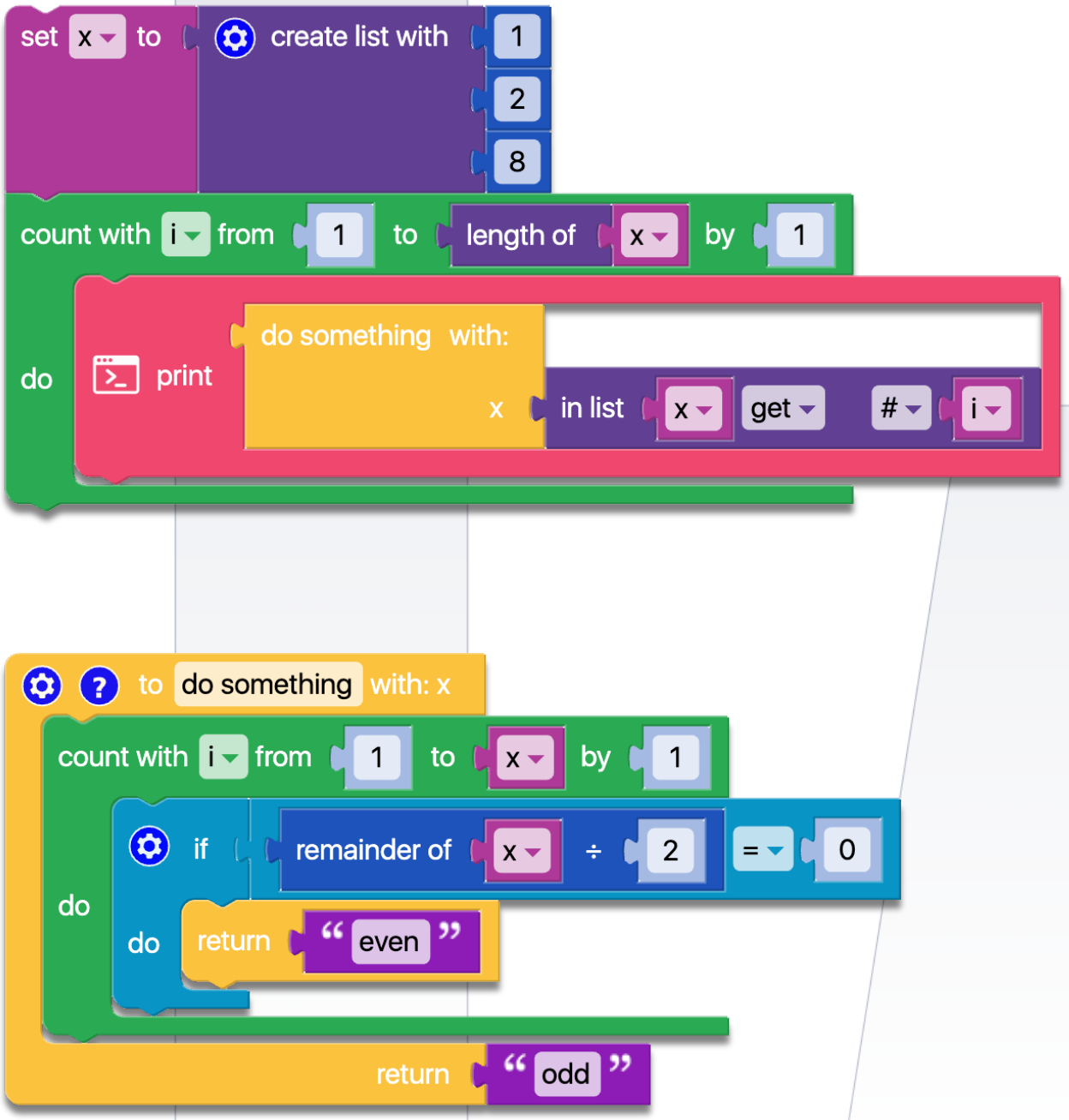
shape robotics
SMART LEARNING

Code 77:
Extract Even/Odd numbers with List&Function

- Before running the program, the user is prompted to enter three numbers via the keyboard. These numbers can be obtained from sensors or intermediate calculations, and they are stored in a list. A function is then employed to determine whether each number in the list is even or odd. The resulting output is displayed in the console.

WATCH VIDEO



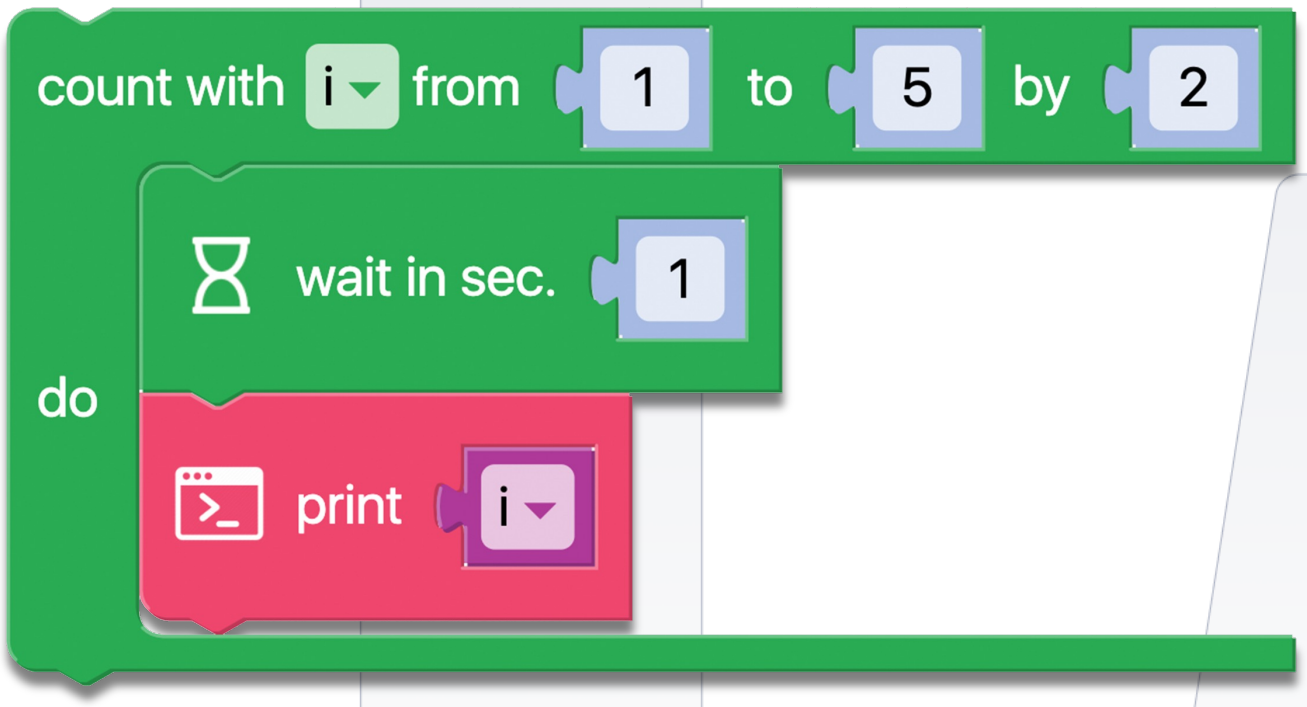

set x to create list with 1 2 8
count with i from 1 to length of x by 1
do print do something with: x in list x get # i
to do something with: x
count with i from 1 to x by 1
do if remainder of x ÷ 2 = 0
do return "even"
return "odd"

shape robotics
SMART LEARNING

Code 78:
Using “count with” command

- The program sequence is a repetitive structure useful when we want to execute instructions a fixed number of times, as opposed to executing code based on a condition (while). In this sequence, odd numbers in the closed interval [1, 5] are displayed.

WATCH VIDEO

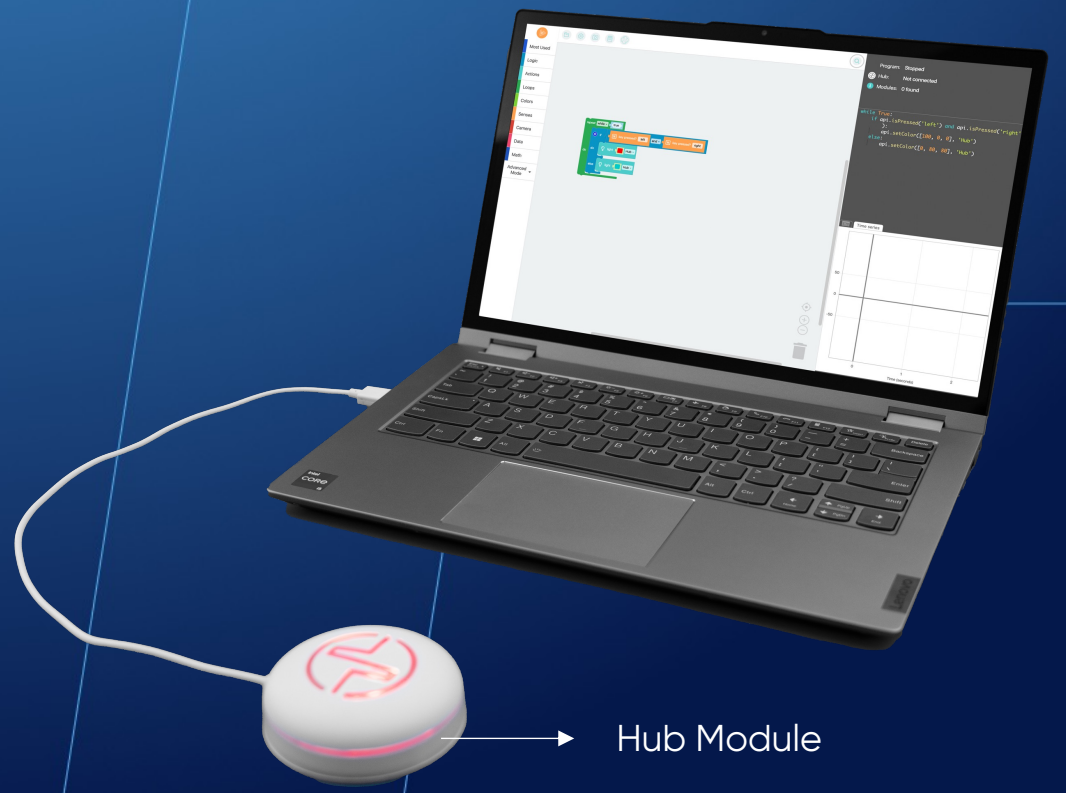

count with i from 1 to 5 by 2
wait in sec. 1
do
print i

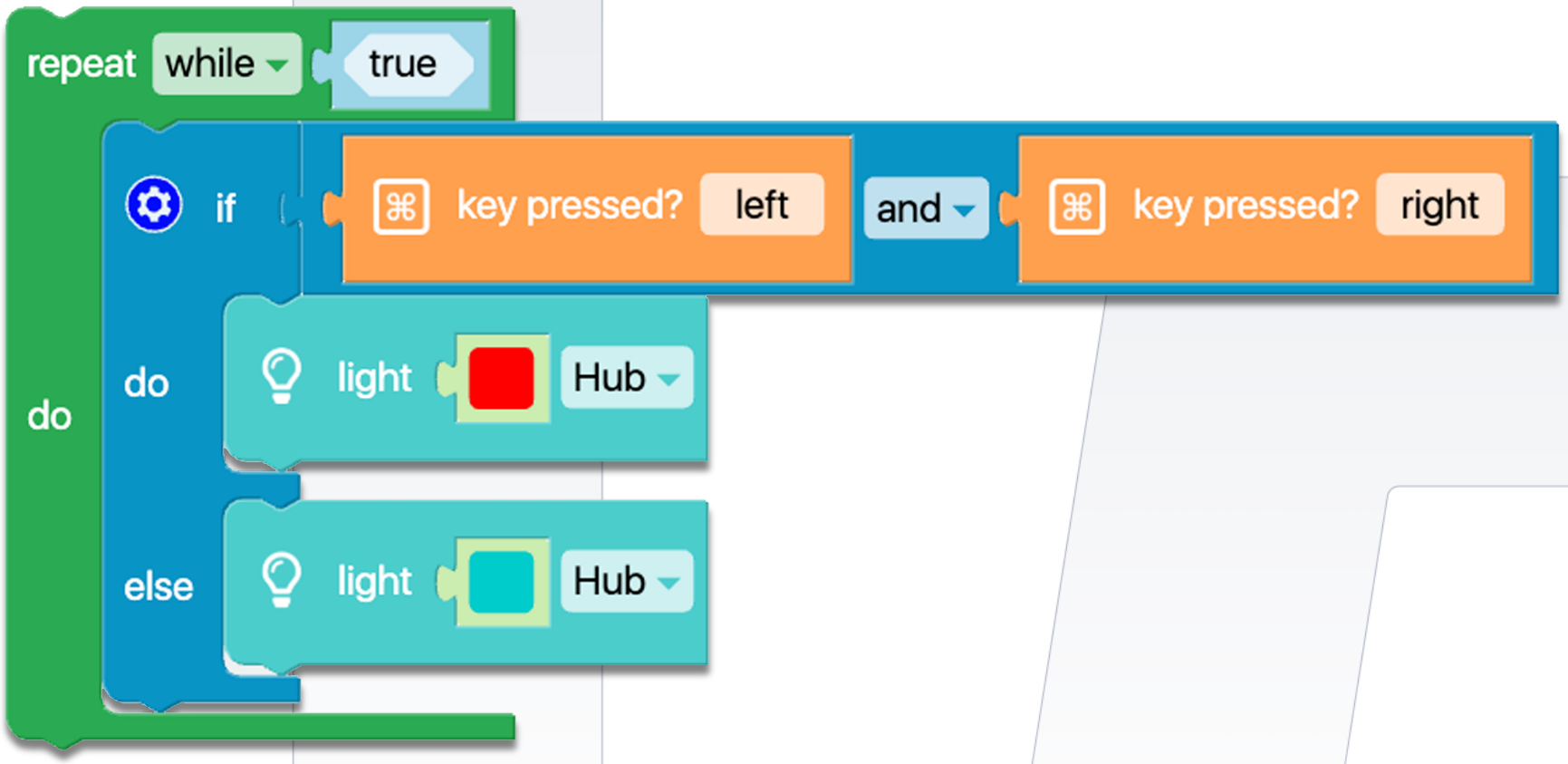
shape robotics
SMART LEARNING

Code 79:
Boolean Logical Operator - AND

- After the condition, the code sequence is executed. The Hub light will remain green as long as neither the left arrow nor the right arrow keys are pressed. However, if both keys are pressed simultaneously, the light will change to red. This operator is particularly useful when reading data from multiple sensors and making a decision based on a combination of their readings.

WATCH VIDEO



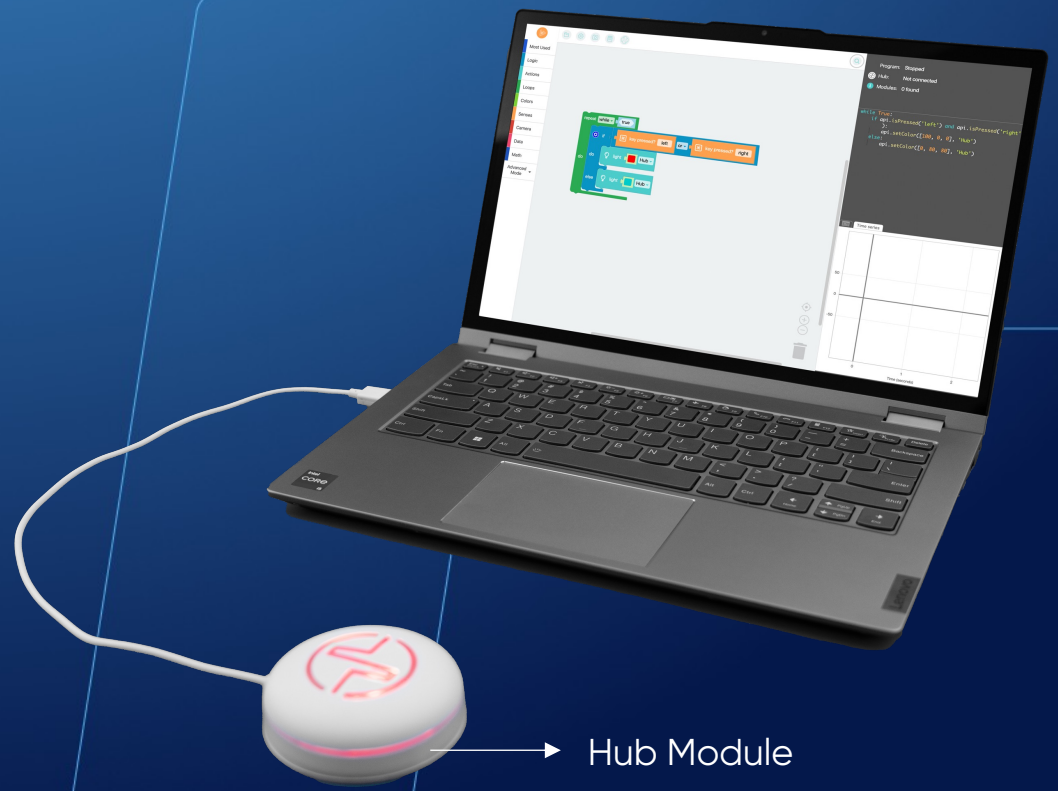

repeat
while
true
do
if
key pressed?
left
and
key pressed?
right
do
light
Hub
else
light
Hub

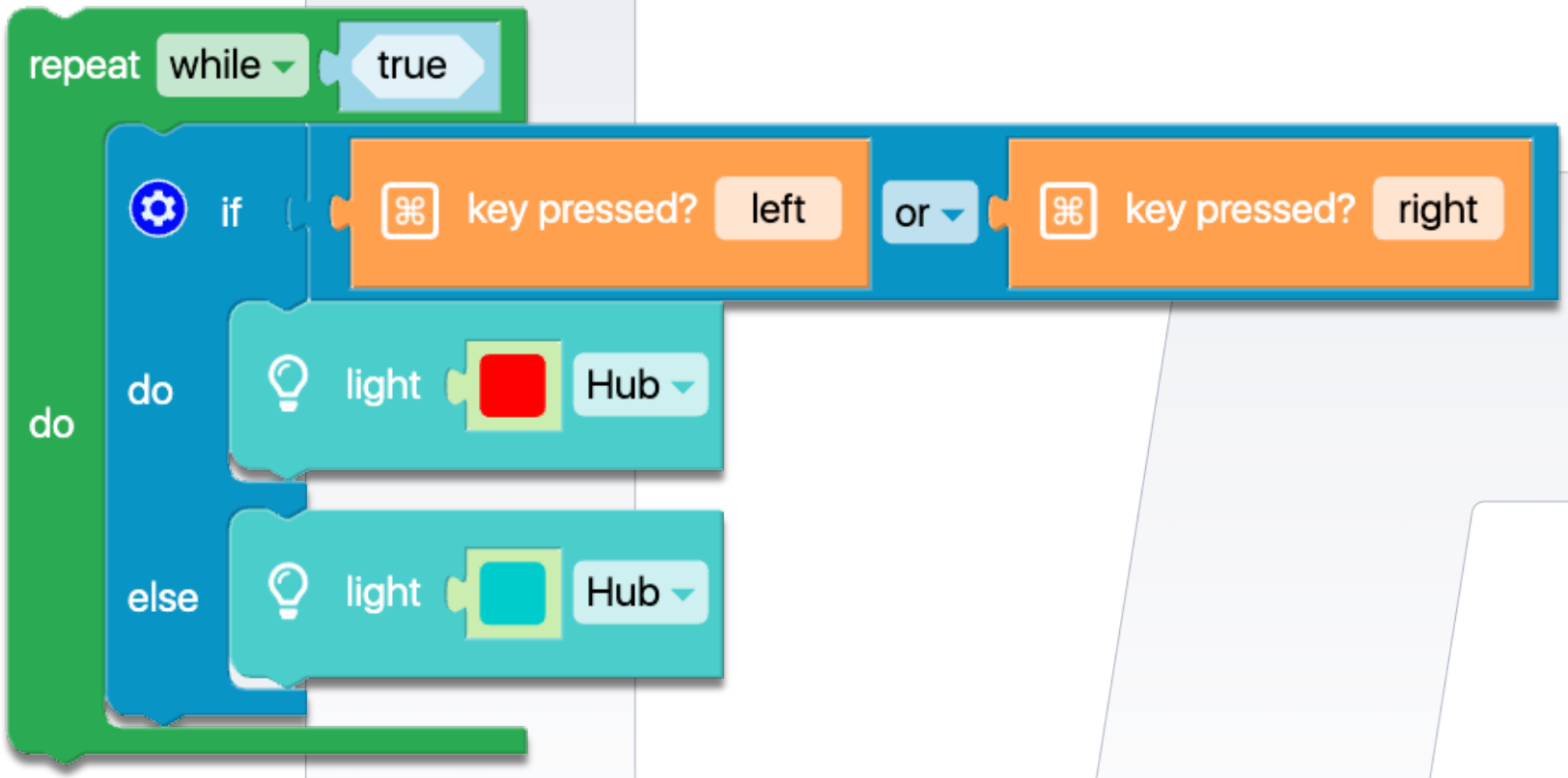
shape robotics
SMART LEARNING

Code 80:
Boolean Logical Operator - OR

- The code sequence ensures that the Hub remains blue as long as the left arrow key OR the right arrow key is pressed on the keyboard. However, if one of these keys is pressed, the Hub will turn red. This sequence is useful in programs where a condition can be met in two different ways, such as by pressing a key or by activating a sensor.

WATCH VIDEO



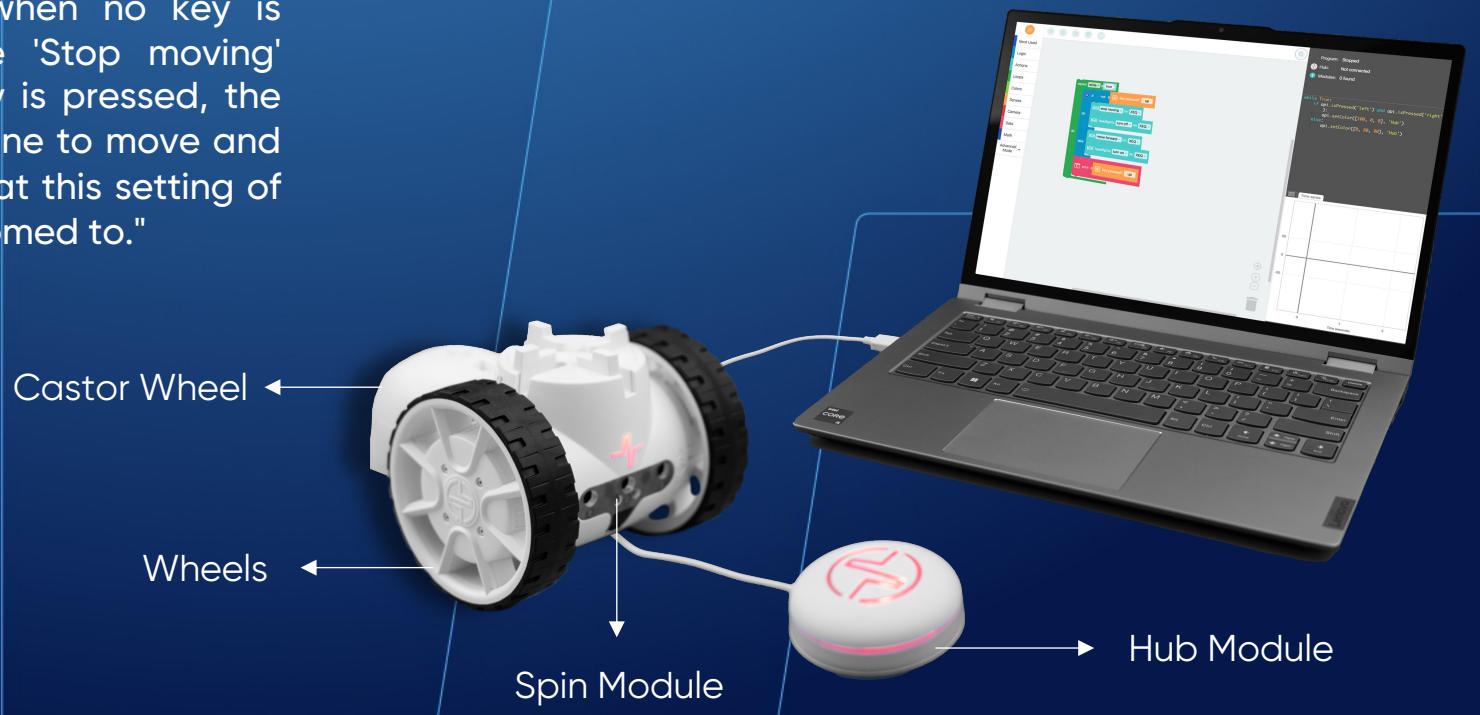

repeat
while
true
do
if
key pressed?
left
or
key pressed?
right
do
light
Hub
else
light
Hub

shape robotics
SMART LEARNING

Code 81:
Boolean Logical Operator - NOT

- This code sequence is designed to make the Spin module move forward exclusively when the up-arrow key is pressed. A NON-operator is employed, triggered when no key is pressed, activating the IF body with the 'Stop moving' command if the lights are off. When no key is pressed, the ELSE body becomes active, causing the engine to move and the lights to be turned on. It's noteworthy that this setting of conditions is contrary to what we are accustomed to."

WATCH VIDEO



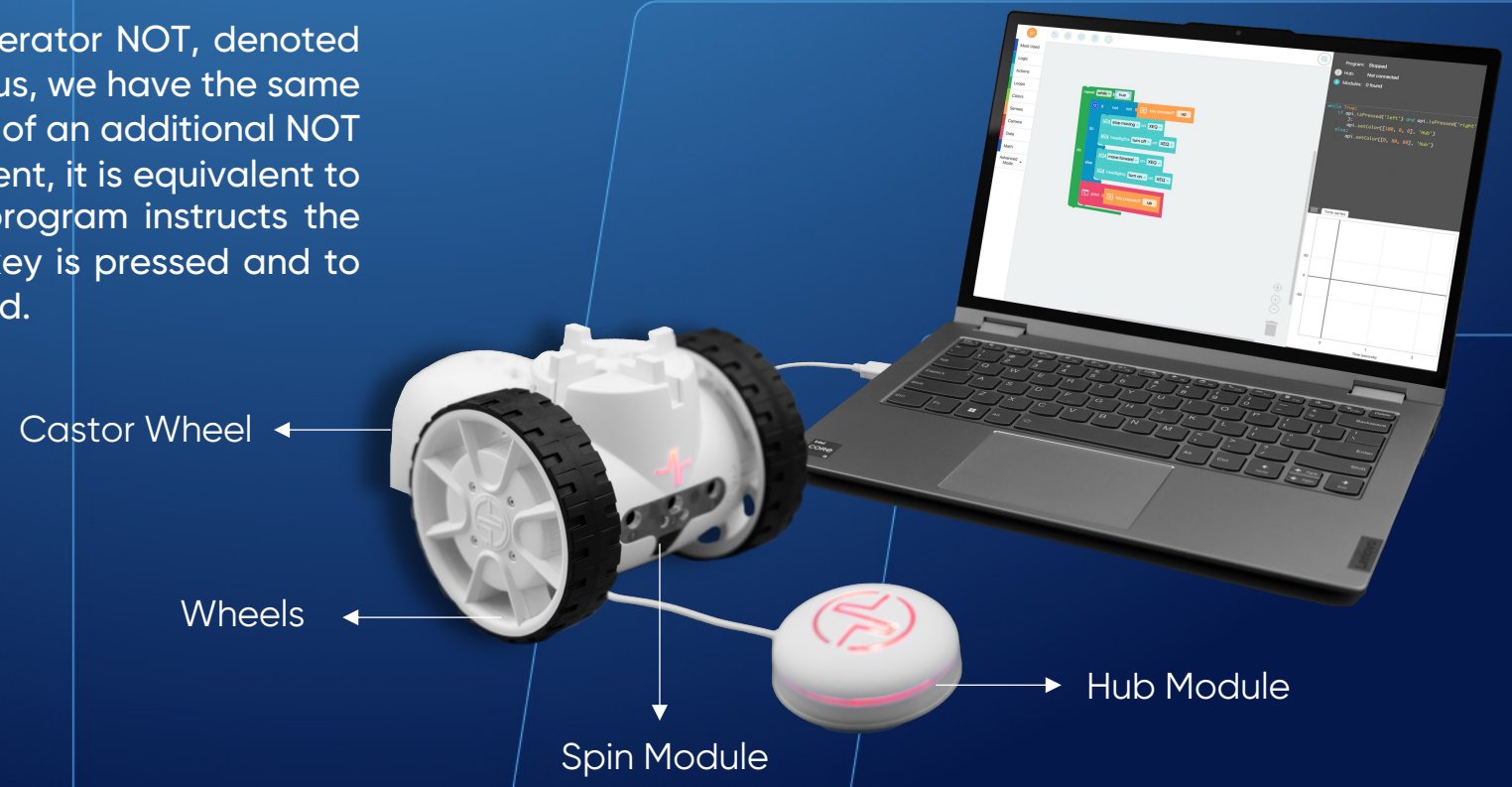

repeat while true
if not key pressed? up
do
stop moving on XEQ
headlights turn off on XEQ
do
move forward on XEQ
else
headlights turn on on XEQ
print key pressed? up

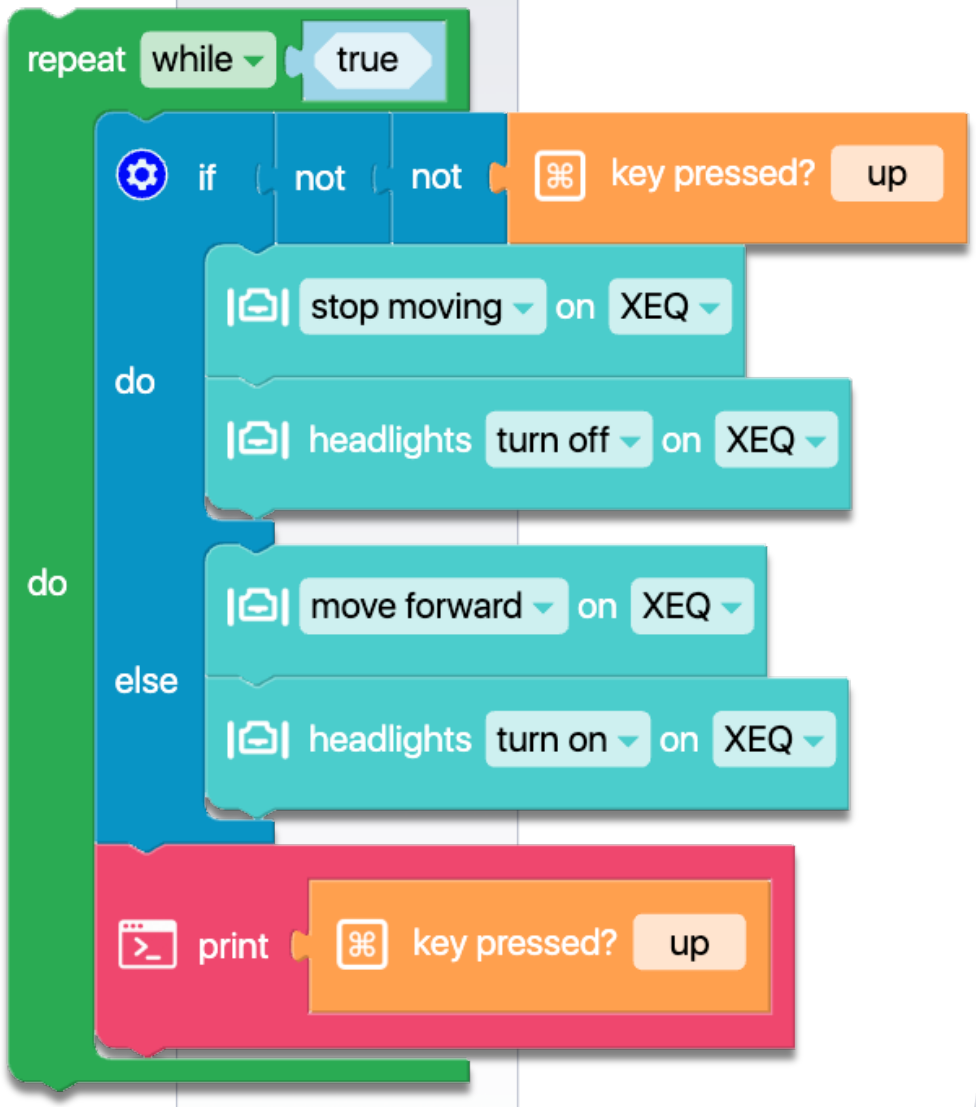
shape robotics
SMART LEARNING

Code 82:
Double Negation

- This code utilizes the Boolean logical operator NOT, denoted as “Boolean Logical Operator – NOT”. Thus, we have the same program as before, but with the inclusion of an additional NOT command. When two negations are present, it is equivalent to having none at all. Consequently, this program instructs the Spin module to move forward when no key is pressed and to halt when the forward arrow key is pressed.

WATCH VIDEO



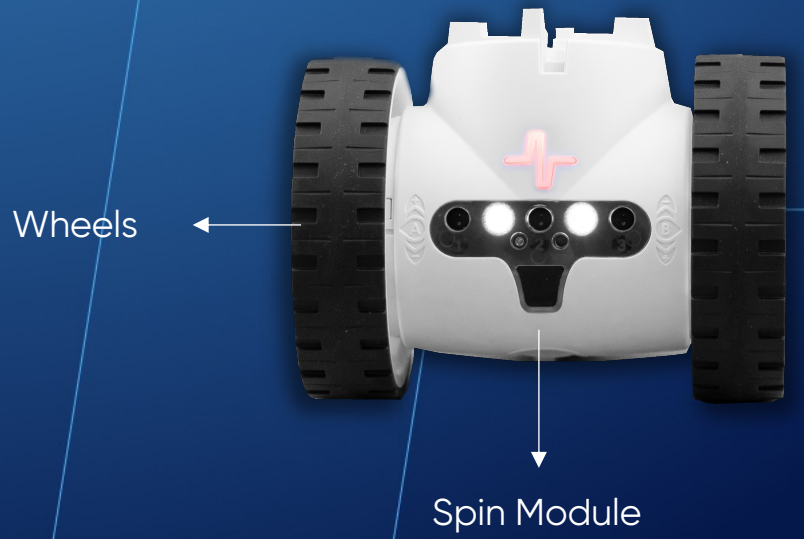

repeat
while
true
if
not
not
key pressed?
up
do
stop moving
on
XEQ
headlights
turn off
on
XEQ
do
move forward
on
XEQ
else
headlights
turn on
on
XEQ
print
key pressed?
up

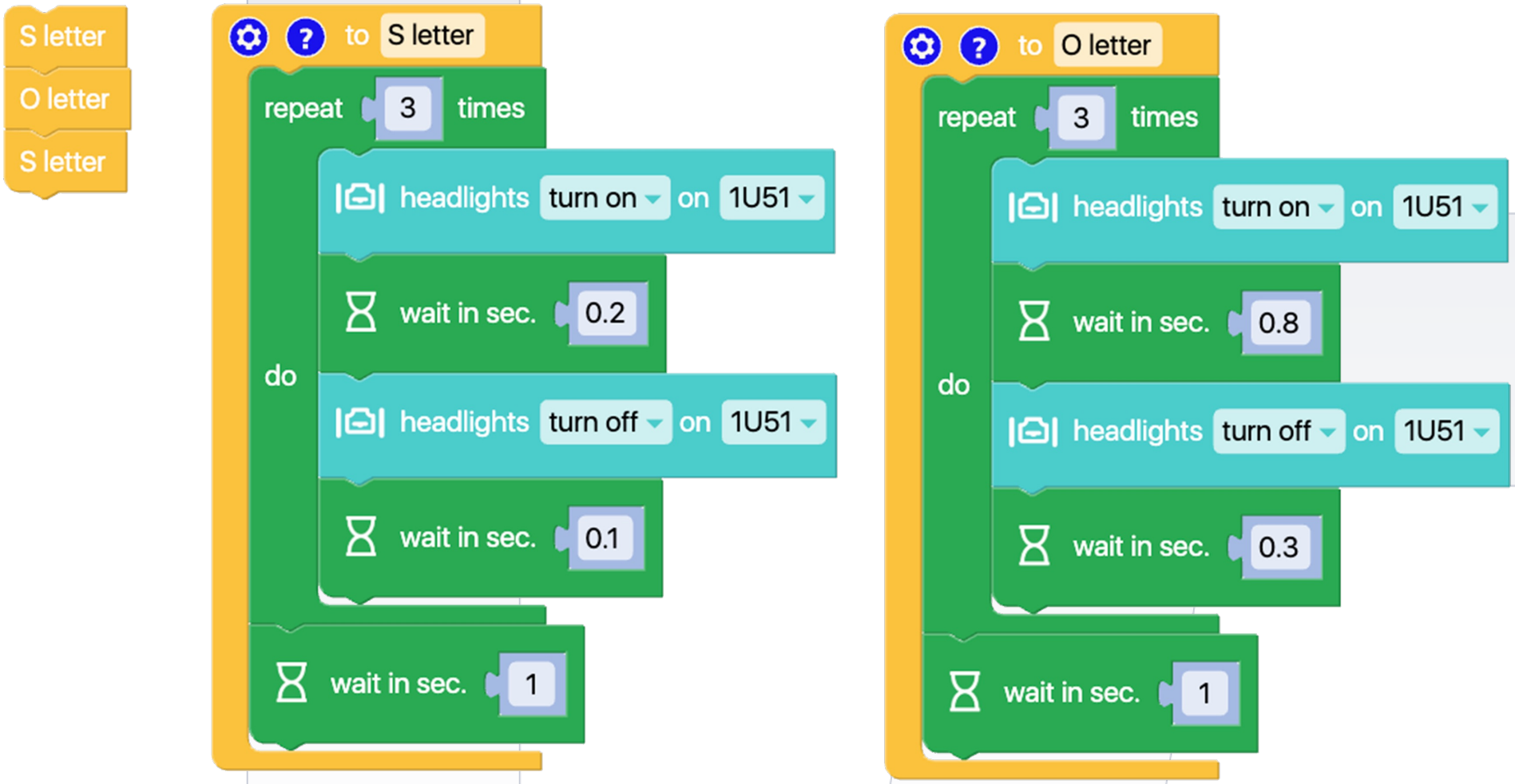
shape robotics
SMART LEARNING

Code 83:
Intermittent Lights with Spin

- By utilizing a Spin module and the in-built LEDs, the robot will send out the SOS code through the LEDs. You can modify the on and off timings to determine if the message is still understandable. The pause between letters is represented by the one-second command. Two separate functions have been utilized - one for the letter S and another for the letter O.

WATCH VIDEO



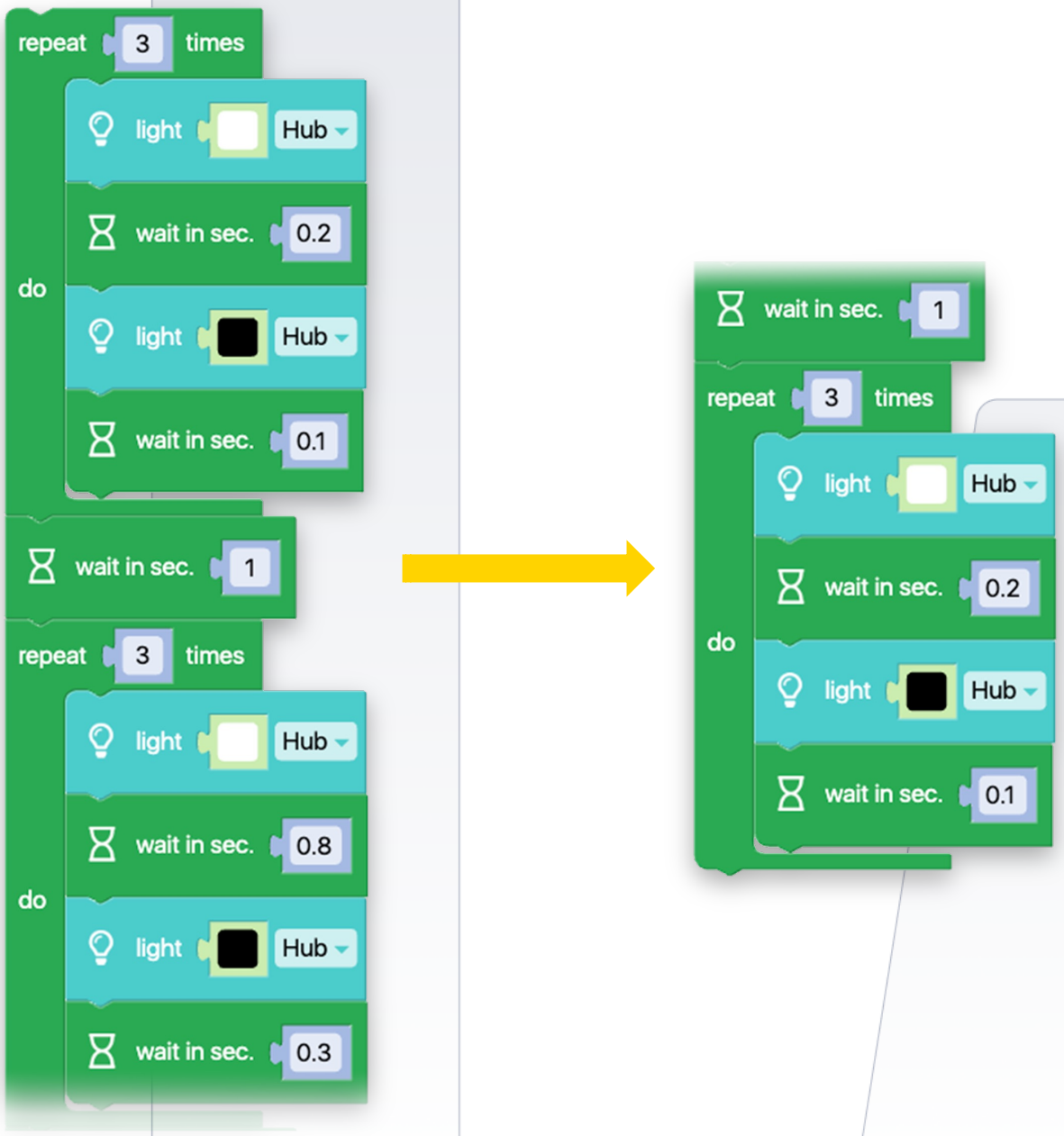

S letter
O letter
S letter
to S letter
repeat 3 times
do
headlights turn on on 1U51
wait in sec. 0.2
headlights turn off on 1U51
wait in sec. 0.1
wait in sec. 1
to O letter
repeat 3 times
do
headlights turn on on 1U51
wait in sec. 0.8
headlights turn off on 1U51
wait in sec. 0.3
wait in sec. 1

shape robotics
SMART LEARNING

Code 84:
Intermittent Lights with Hub

- The program uses the Hub to transmit a bright SOS signal, similar to Morse code. Altering the timings will affect how long the Hub light is on or off.

WATCH VIDEO


repeat 3 times
light Hub
wait in sec. 0.2
do
light Hub
wait in sec. 0.1
wait in sec. 1
repeat 3 times
light Hub
wait in sec. 0.8
do
light Hub
wait in sec. 0.3
wait in sec. 1
repeat 3 times
light Hub
wait in sec. 0.2
do
light Hub
wait in sec. 0.1

shape robotics
SMART LEARNING

Code 85:
Print in Console

- The output console can be accessed by clicking the >_ button. When the run button is pressed, the console displays the text "Hello! I am Fable!", which can be modified by the programmer.

WATCH VIDEO

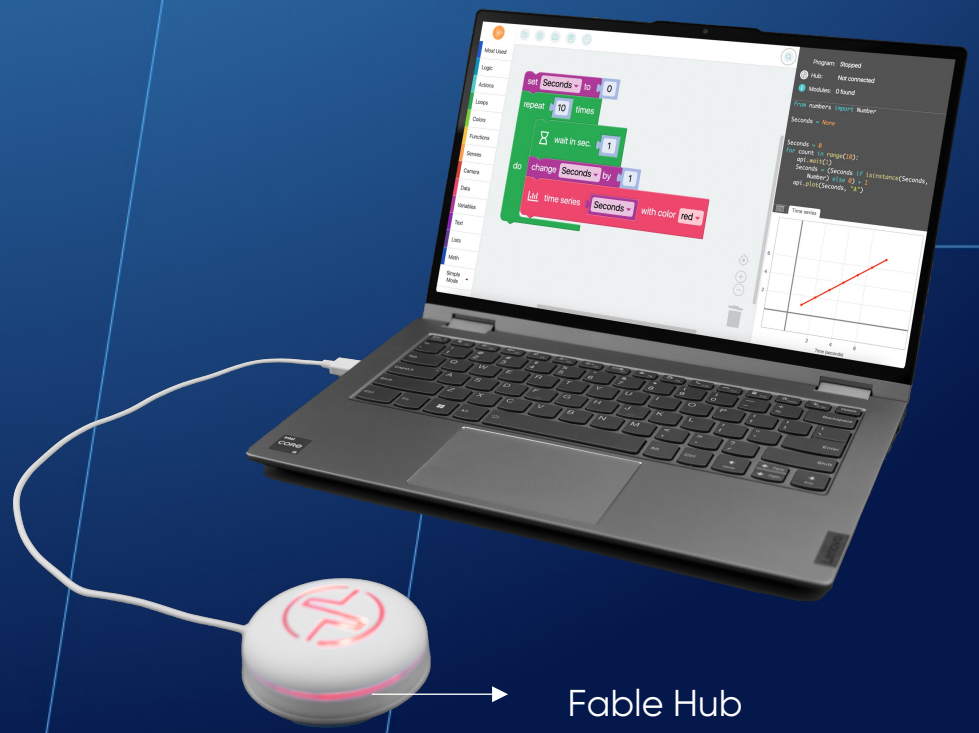
shape robotics
SMART LEARNING

shape robotics
SMART LEARNING

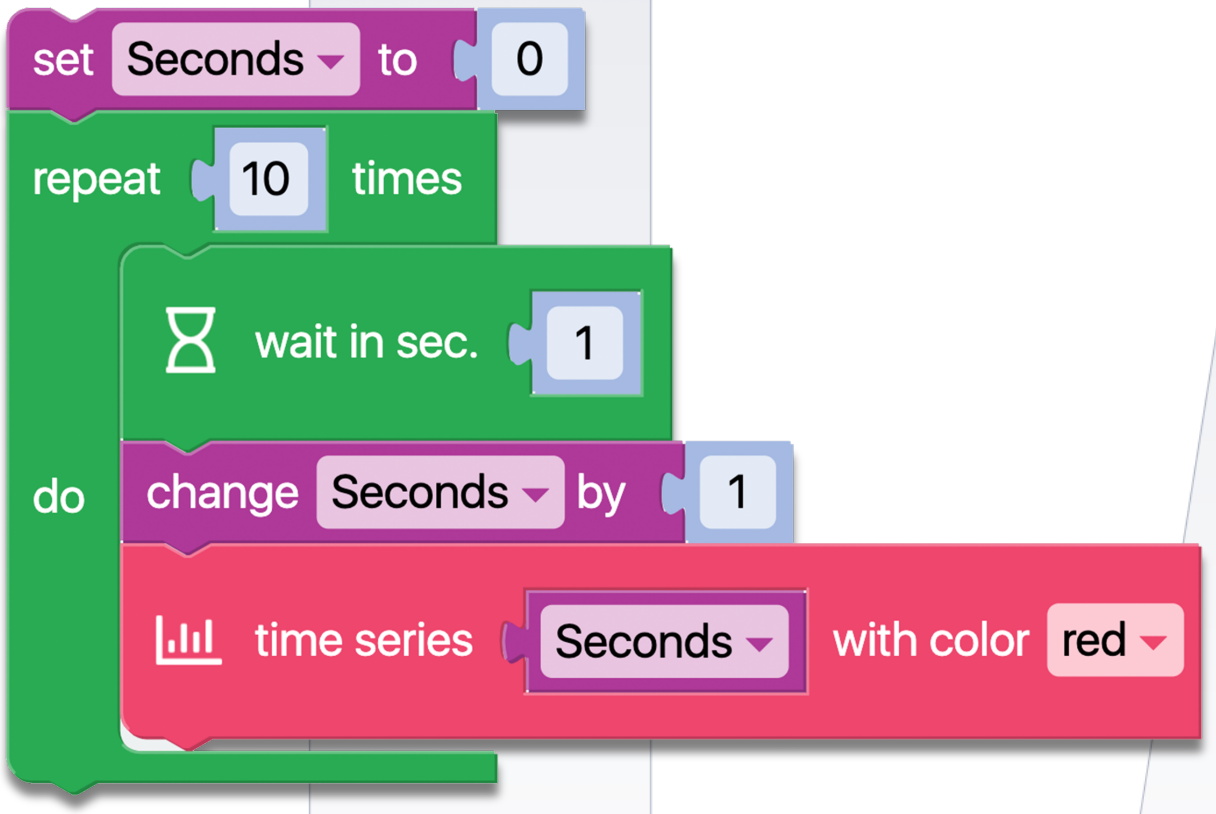
Code 86:
Display a Time Series Graph

- The program initializes a variable to 0 and increments it by 1 every second. It displays a red graphic with the variable value and allows the user to change the color.

WATCH VIDEO



shape robotics
SMART LEARNING

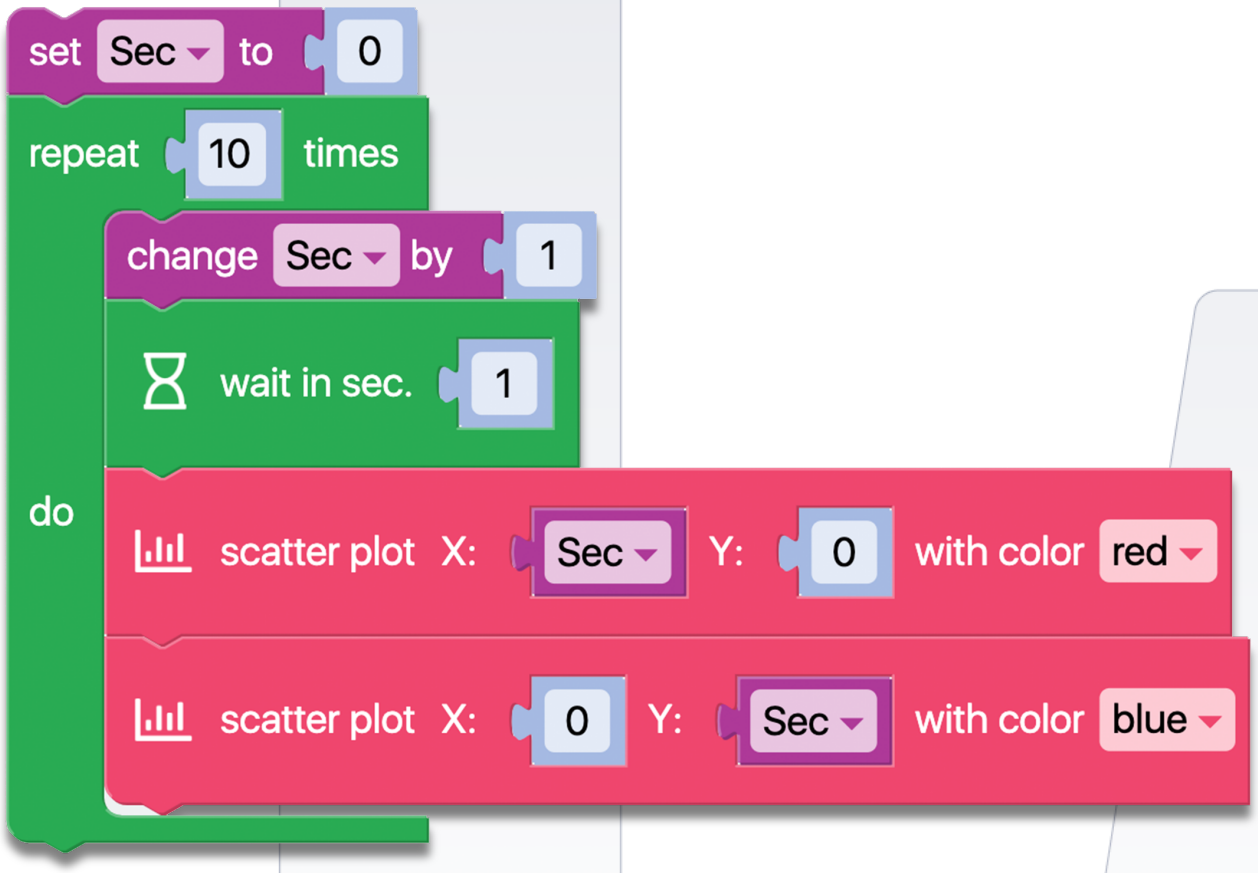


shape robotics
SMART LEARNING

Code 87:
Display a Scatter Plot

- The program sets up a variable called "sec" (seconds) and updates it every second. It then uses different colors to display the value of "sec" on the X and Y axes. Blue is used to show seconds on the Y-axis, and red displays seconds on the X-axis. This kind of scatter plot can be utilized to retrieve data from a Joint, Spin, or phone module (with the Fable Face application).

WATCH VIDEO

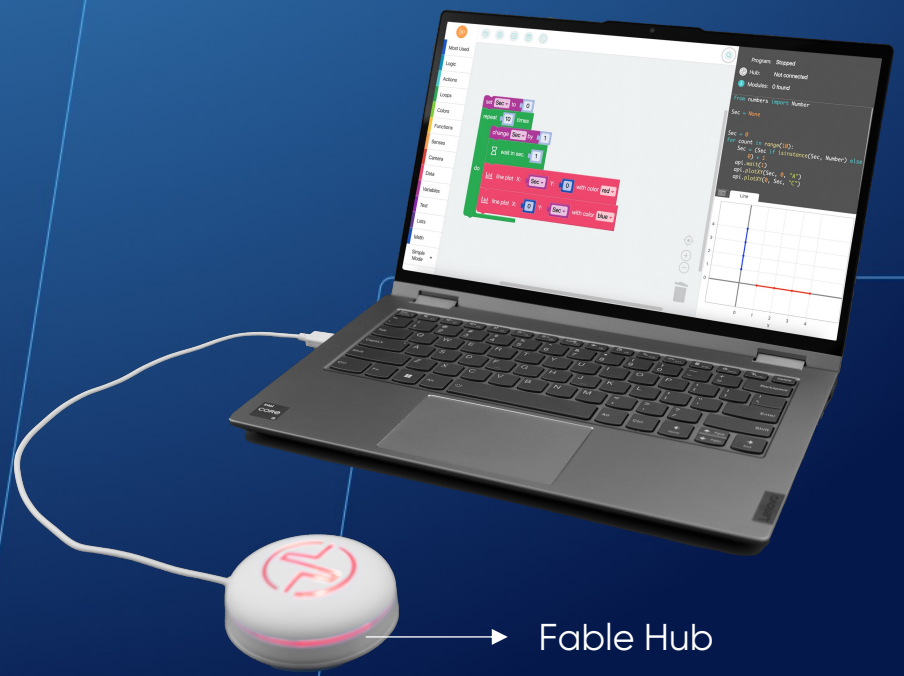

set Sec to 0
repeat 10 times
change Sec by 1
wait in sec. 1
do
scatter plot X: Sec Y: 0 with color red
scatter plot X: 0 Y: Sec with color blue

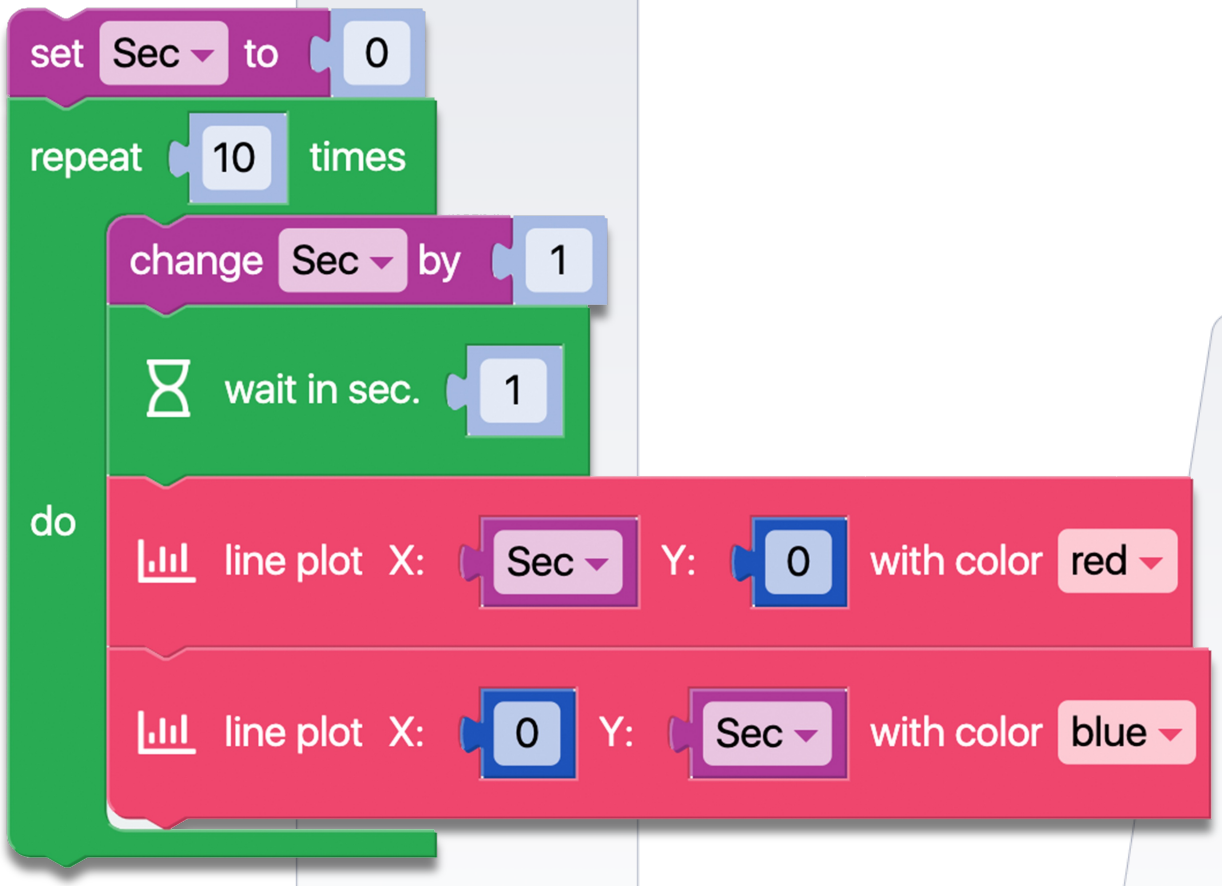
shape robotics
SMART LEARNING

Code 88:
Display a Line Plot

- The program sets up a variable called "sec" (seconds) and updates it every second. It then uses different colors to display the value of "sec" on the X and Y axes. Blue is used to show seconds on the Y-axis, and red displays seconds on the X-axis. This kind of line plot can be utilized to retrieve data from a Joint, Spin, or phone (with the Fable Face application).

WATCH VIDEO



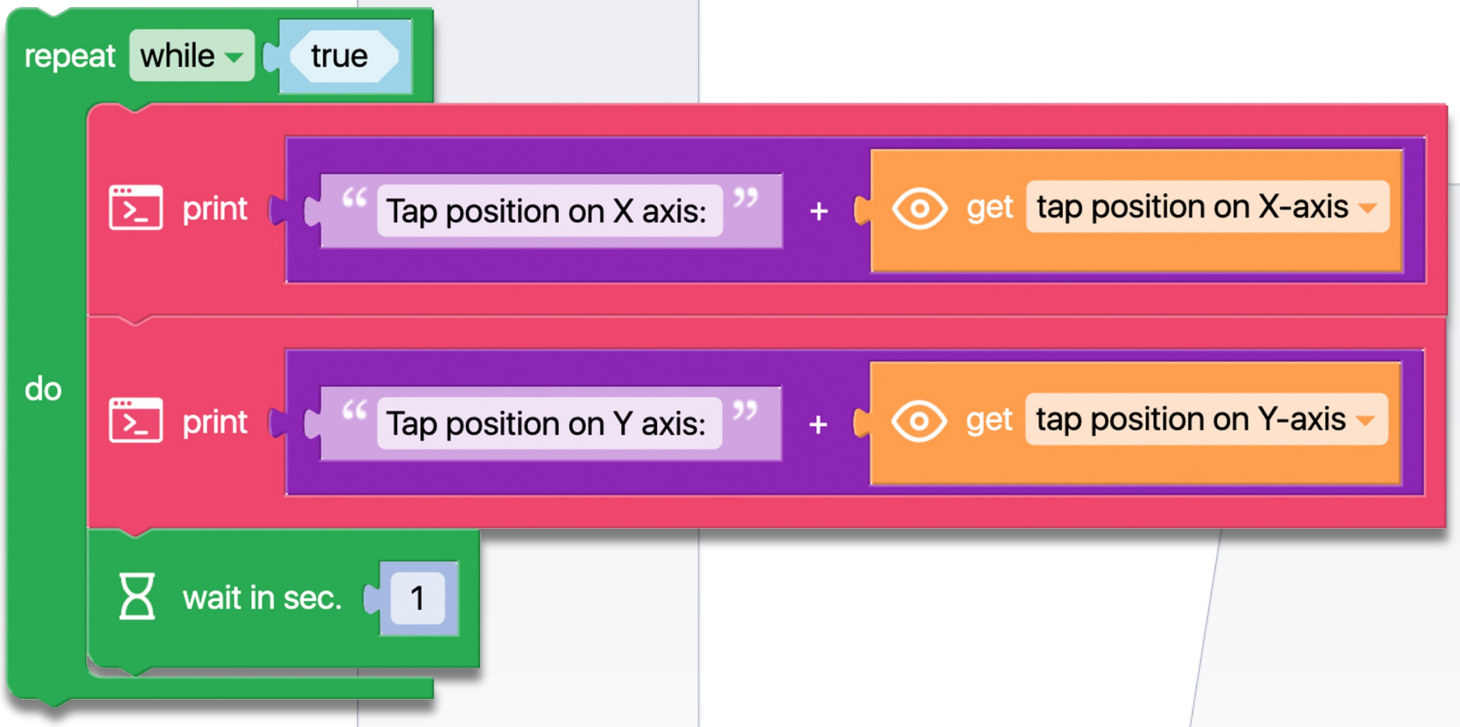

set Sec to 0
repeat 10 times
change Sec by 1
wait in sec. 1
do
line plot X: Sec Y: 0 with color red
line plot X: 0 Y: Sec with color blue

shape robotics
SMART LEARNING

Code 89:
Display Screen Tap Position

- The program displays the coordinates of a finger's position on the phone screen connected to the Hub every second in the console, represented numerically concerning the orthogonal axes.

WATCH VIDEO

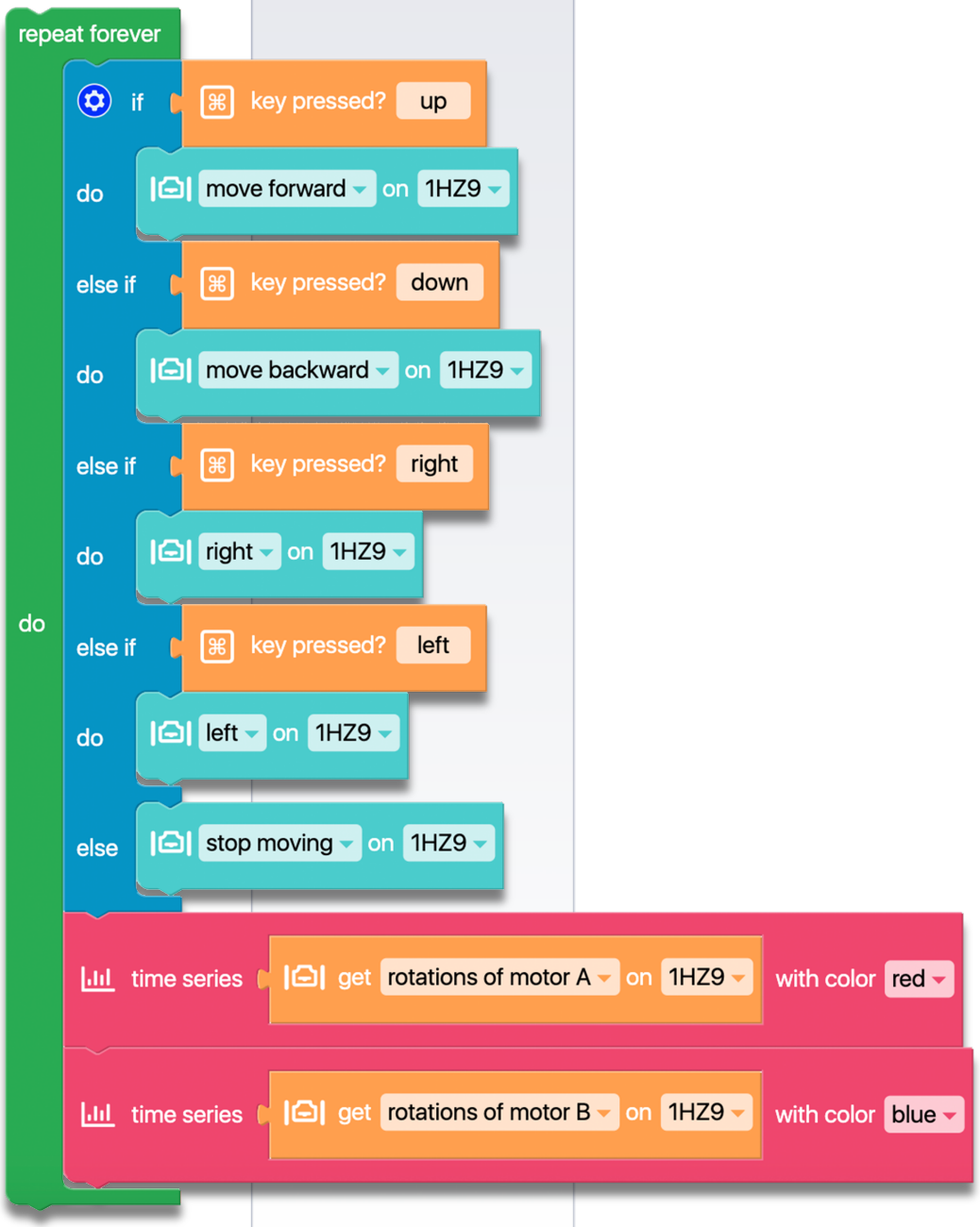

repeat
while
true
do
print
Tap position on X axis:
+
get
tap position on X-axis
print
Tap position on Y axis:
+
get
tap position on Y-axis
wait in sec.
1

shape robotics
SMART LEARNING

Code 90:
Graphic Spin Module Rotations

- The “Spin Key Control” program controls the Spin module using keys. Two new commands, "Time series graph", have been added to display a graphic in the Output Console. This graph shows the rotations of the two motors, allowing us to analyze with students the direction of the Spin module's movement after the data has been recorded.

WATCH VIDEO



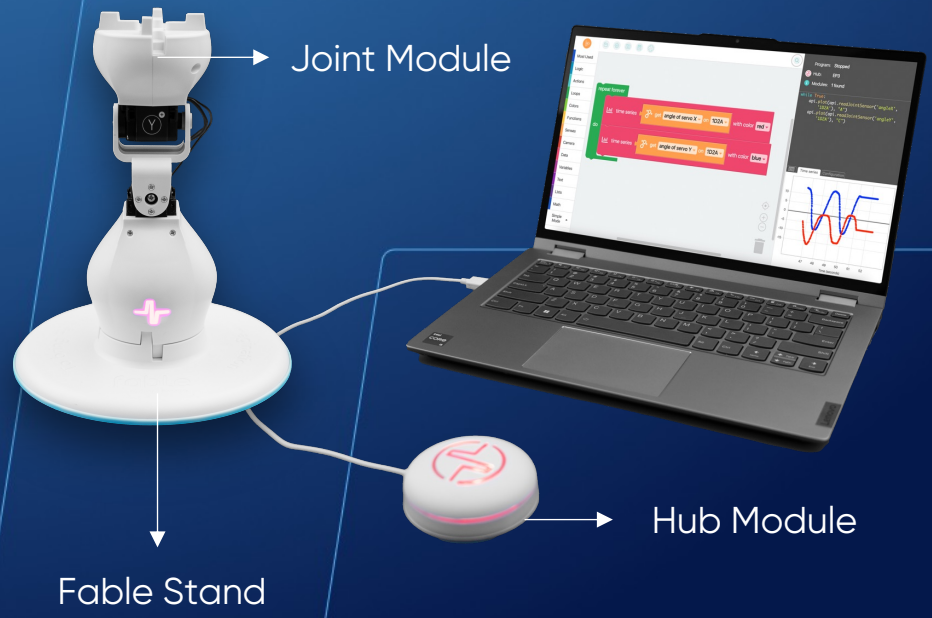
shape robotics
SMART LEARNING

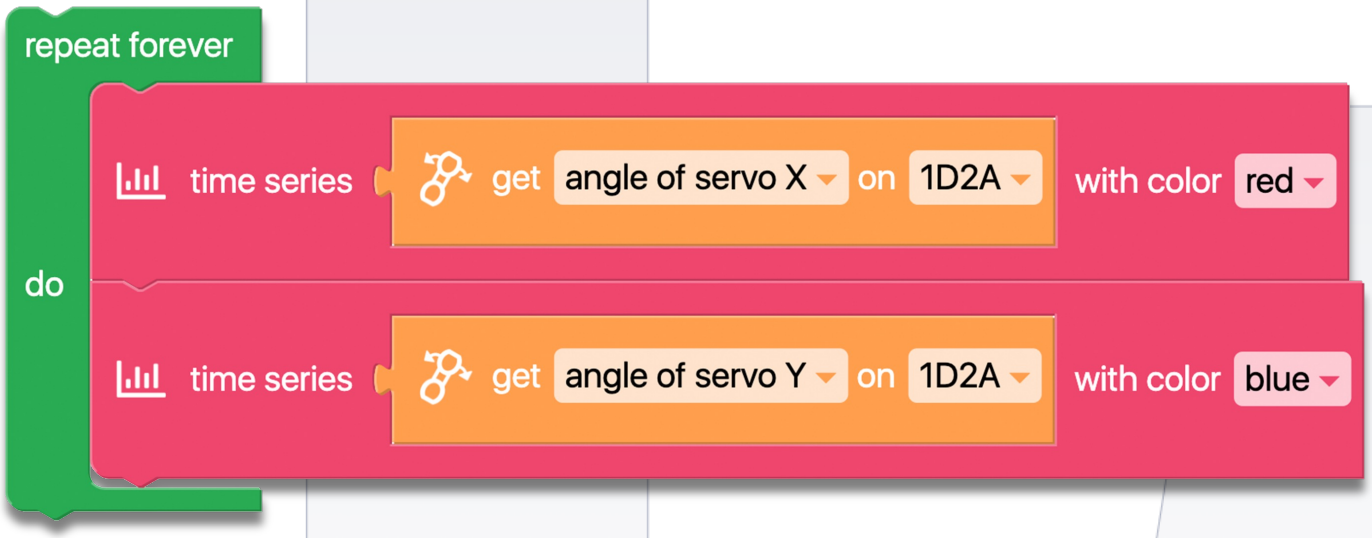
shape robotics
SMART LEARNING

Code 91:
Display Joint angle Graph

- The program generates a graphic that displays the angles of the Joint module. The X motor is represented by the color red and the Y motor is represented by the color blue. Once you have run the program, move the Joint robot slightly to observe the changes reflected in the graphic.

WATCH VIDEO



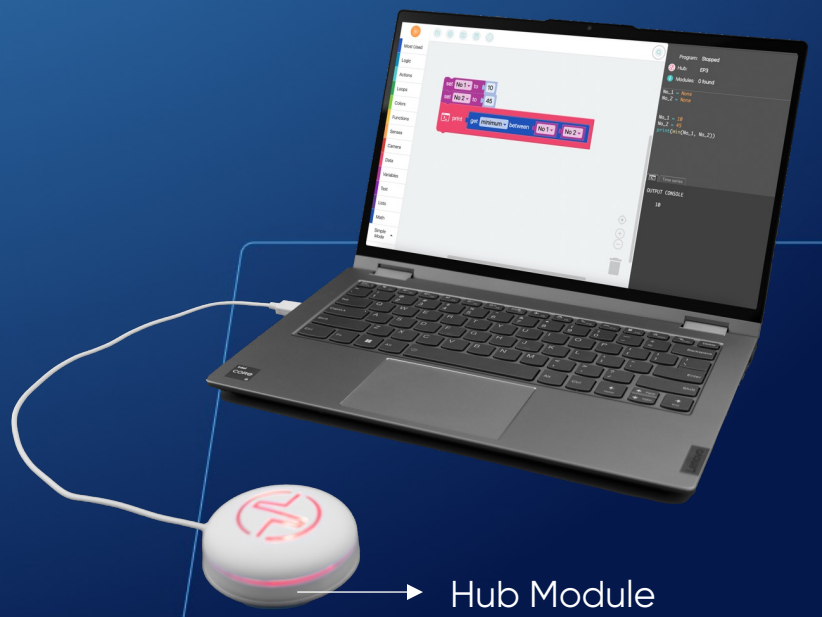

repeat forever
do
time series
get
angle of servo X
on
1D2A
with color
red
time series
get
angle of servo Y
on
1D2A
with color
blue

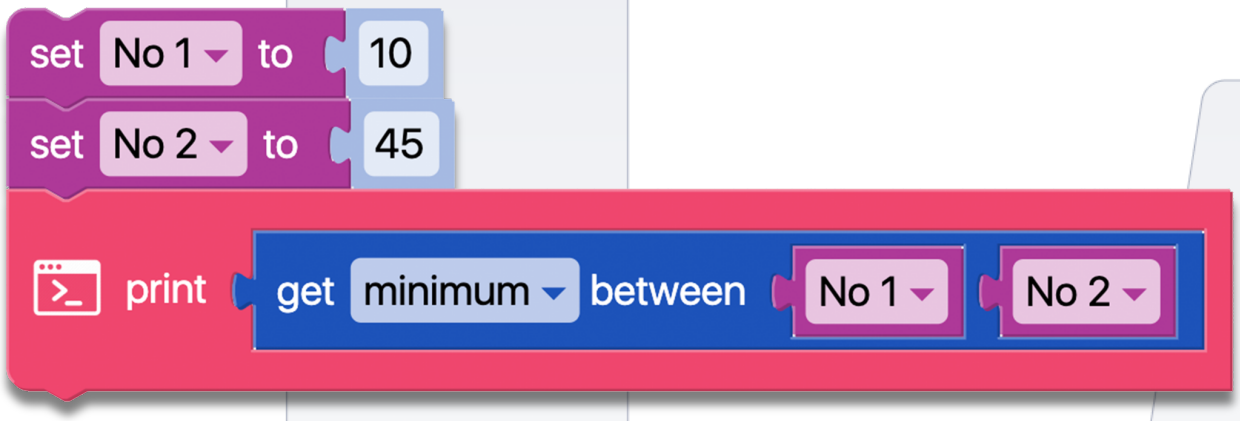
shape robotics
SMART LEARNING

Code 92:
Comparing two presset values

- The program assigns values to two variables and then compares them. These values can be entered from the keyboard at the start of the program or obtained from data collected within the program through sensors or calculations. By using the "get minimum between" command along with the "print" command, the program can output the value of the lower number in the console. In case both variables store the same value, the program will display that value. To display the maximum value, the same program can be used, with the only change being from "minimum" to "maximum".

WATCH VIDEO



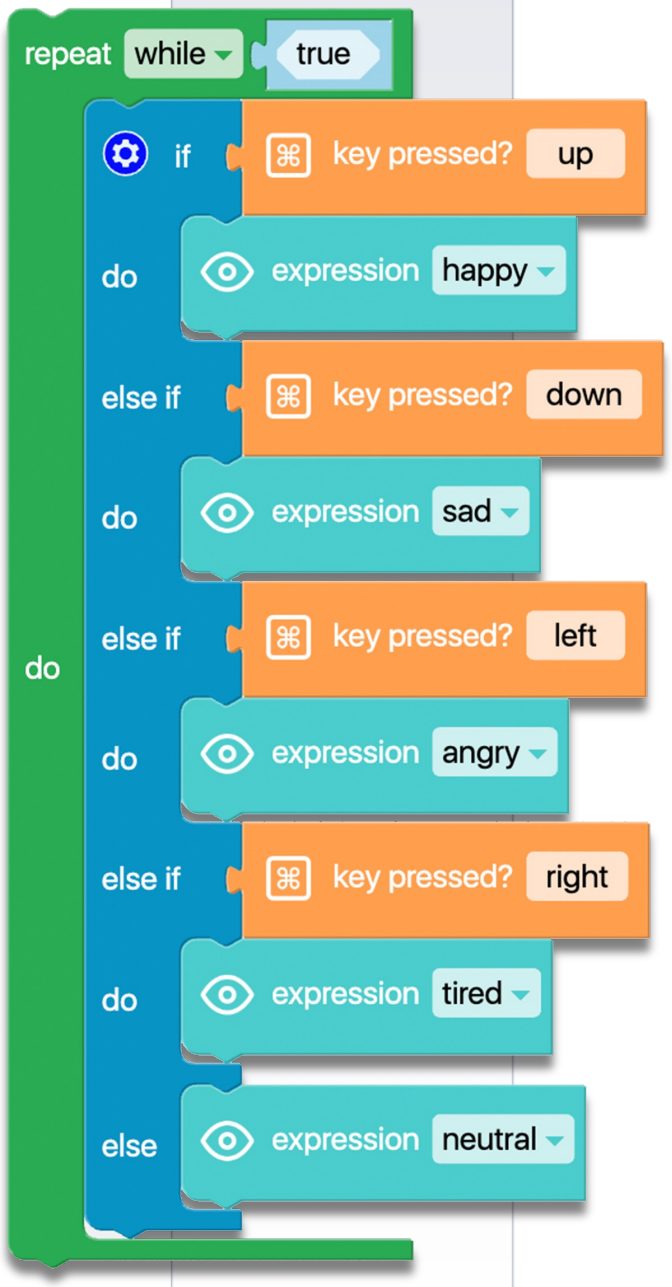

set No 1 to 10
set No 2 to 45
print get minimum between No 1 No 2

shape robotics
SMART LEARNING

Code 93:
Set Fable Face expressions

- The user can control four different face expressions - happy, sad, angry, and neutral - by pressing the directional keys. These expressions will be displayed on the phone screen as long as the corresponding key is pressed.

WATCH VIDEO



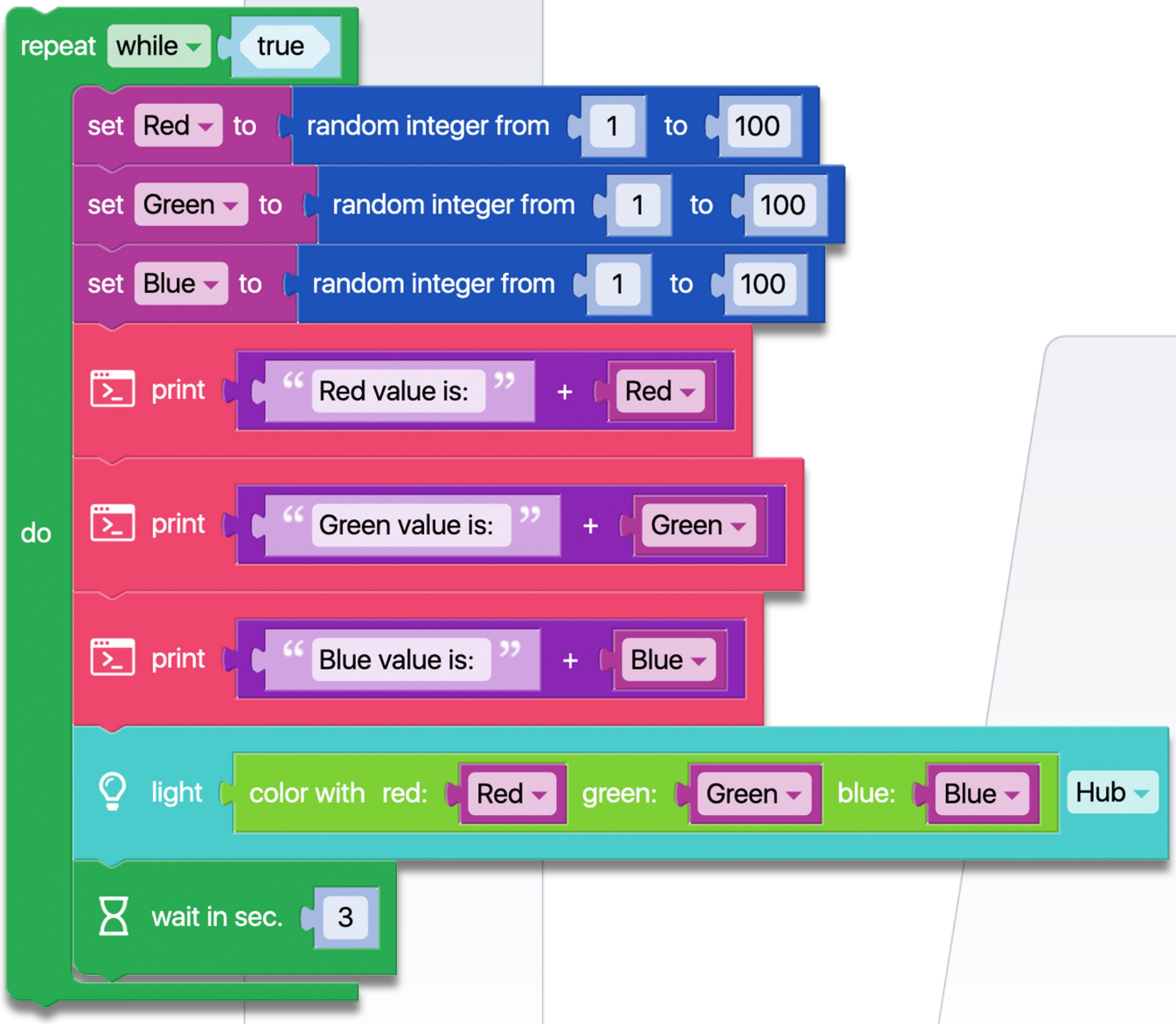
shape robotics
SMART LEARNING

shape robotics
SMART LEARNING

Code 94:
Mixing R,G,B, lights

- Every three seconds, random values between 1 and 100 are assigned to the variables Red, Green, and Blue. These values are used to create a new color, which is displayed by the Hub.

WATCH VIDEO

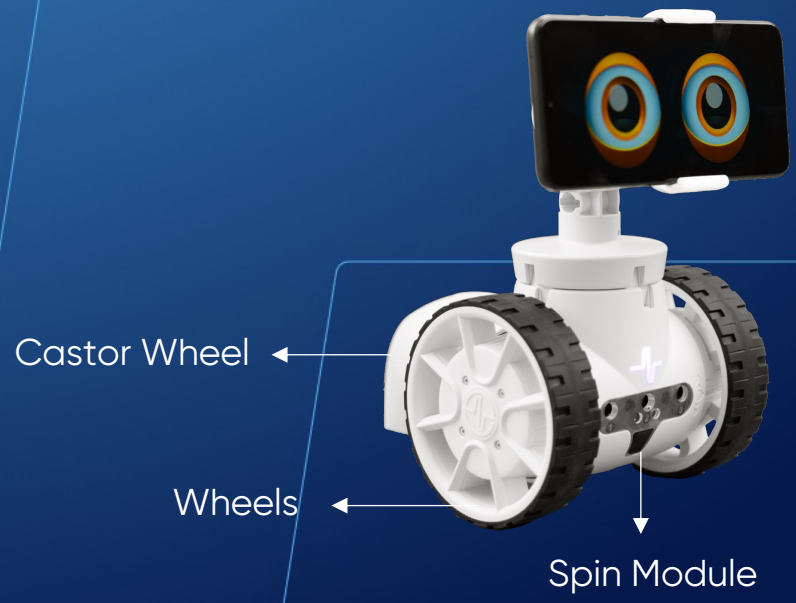

repeat while true
do
set Red to random integer from 1 to 100
set Green to random integer from 1 to 100
set Blue to random integer from 1 to 100
print "Red value is:" + Red
print "Green value is:" + Green
print "Blue value is:" + Blue
light color with red: Red green: Green blue: Blue Hub
wait in sec. 3

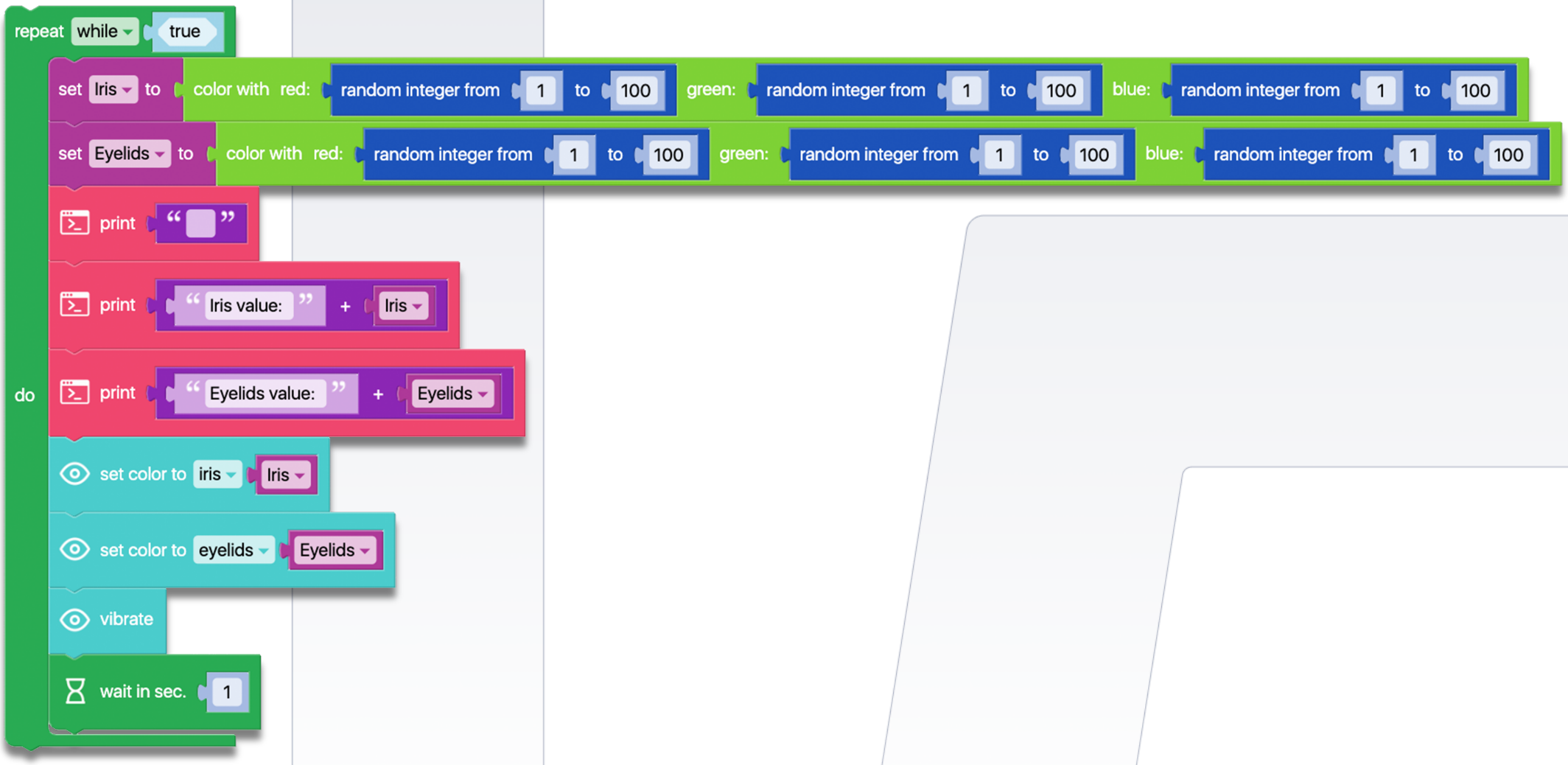
shape robotics
SMART LEARNING

Code 95:
Set iris/eyelids colors

- The program generates different colors for the iris and eyelids of the eyes in Fable Face by using random numbers between 1 and 100 to create combinations of Red, Green, and Blue. The color changes occur every second, and the RGB values obtained are displayed in the Output console. The data in the console is presented in an easy-to-read format using the Print with no text command, by inserting a space between lines.

WATCH VIDEO



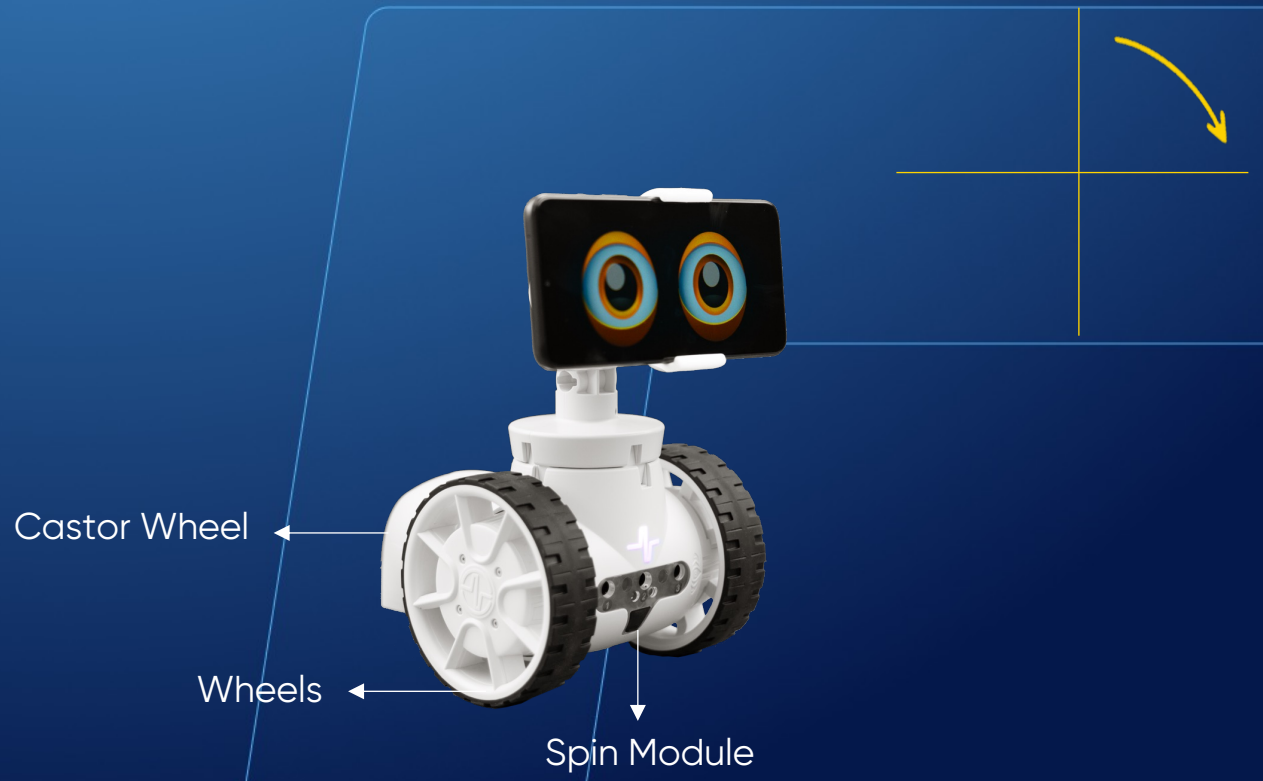

repeat while true
do
set Iris to color with red: random integer from 1 to 100 green: random integer from 1 to 100 blue: random integer from 1 to 100
set Eyelids to color with red: random integer from 1 to 100 green: random integer from 1 to 100 blue: random integer from 1 to 100
print
print Iris value: + Iris
print Eyelids value: + Eyelids
set color to iris Iris
set color to eyelids Eyelids
vibrate
wait in sec. 1

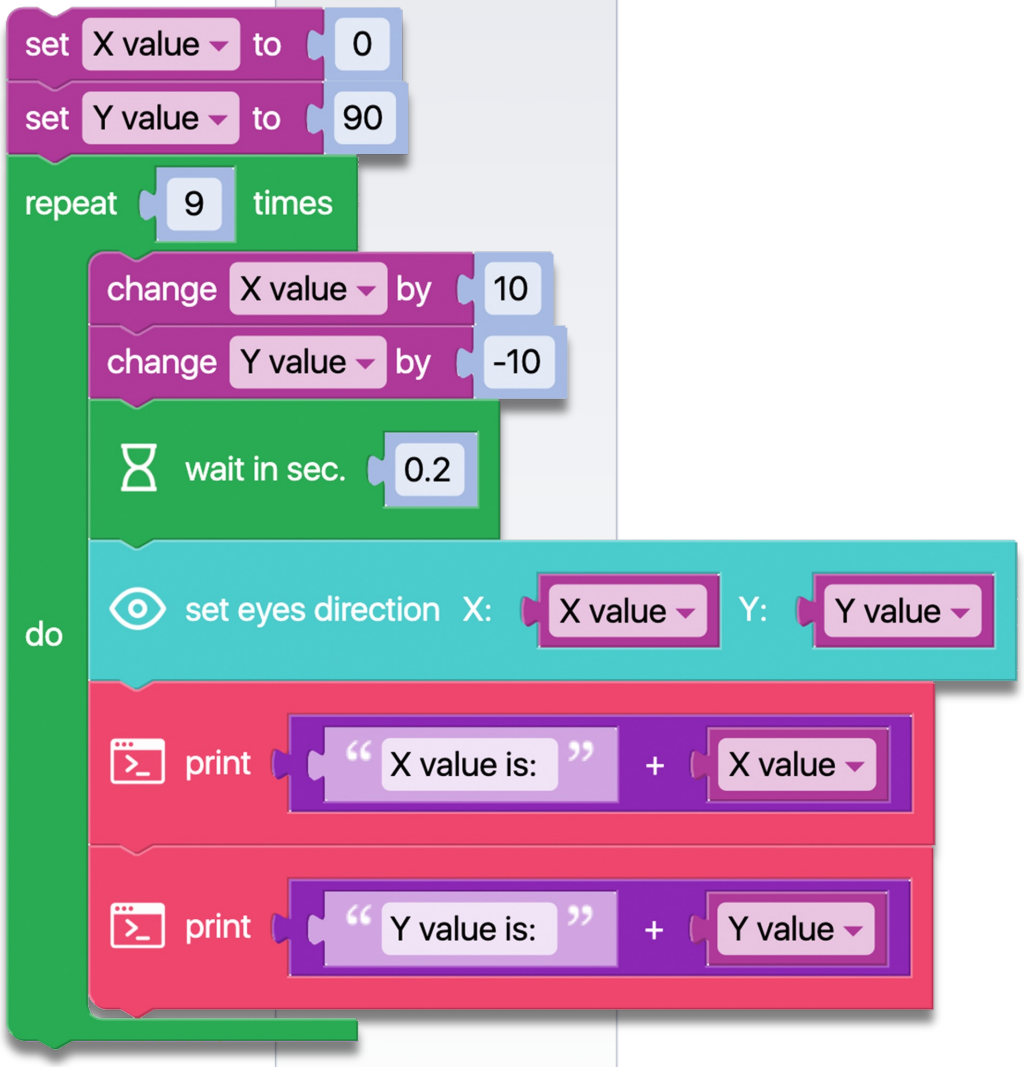
shape robotics
SMART LEARNING

Code 96:
Set eyes Direction

- The program sequence commands Fable's eyes in the Fable Face app to move to quadrant 1 determined by the orthogonal axes on the screen. The ordinate value decreases every 0.2 seconds, while the abscissa value increases every 0.2 seconds.

WATCH VIDEO



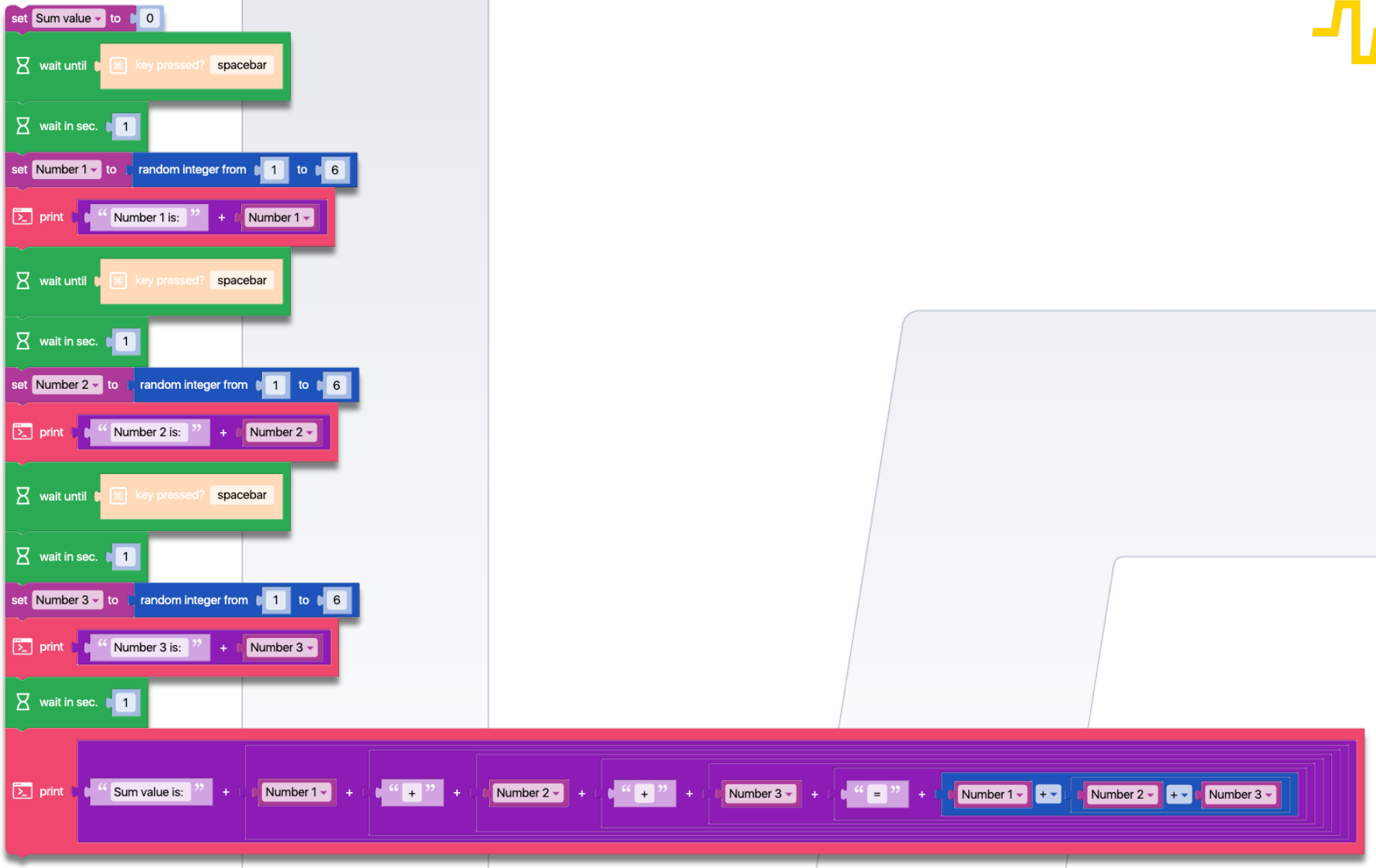

set X value to 0
set Y value to 90
repeat 9 times
change X value by 10
change Y value by -10
wait in sec. 0.2
do
set eyes direction X: X value Y: Y value
print "X value is:" + X value
print "Y value is:" + Y value

shape robotics
SMART LEARNING

Code 97:
Merged Data displayed into a Single Line

- The program generates three random numbers each time the space key is pressed and displays them in a single line in the Output Console. Subsequently, the program presents the numbers individually, separated by plus signs, and then displays their sum in a single line. This program serves as an example of a situation where multiple data need to be displayed within a single line of text.

WATCH VIDEO

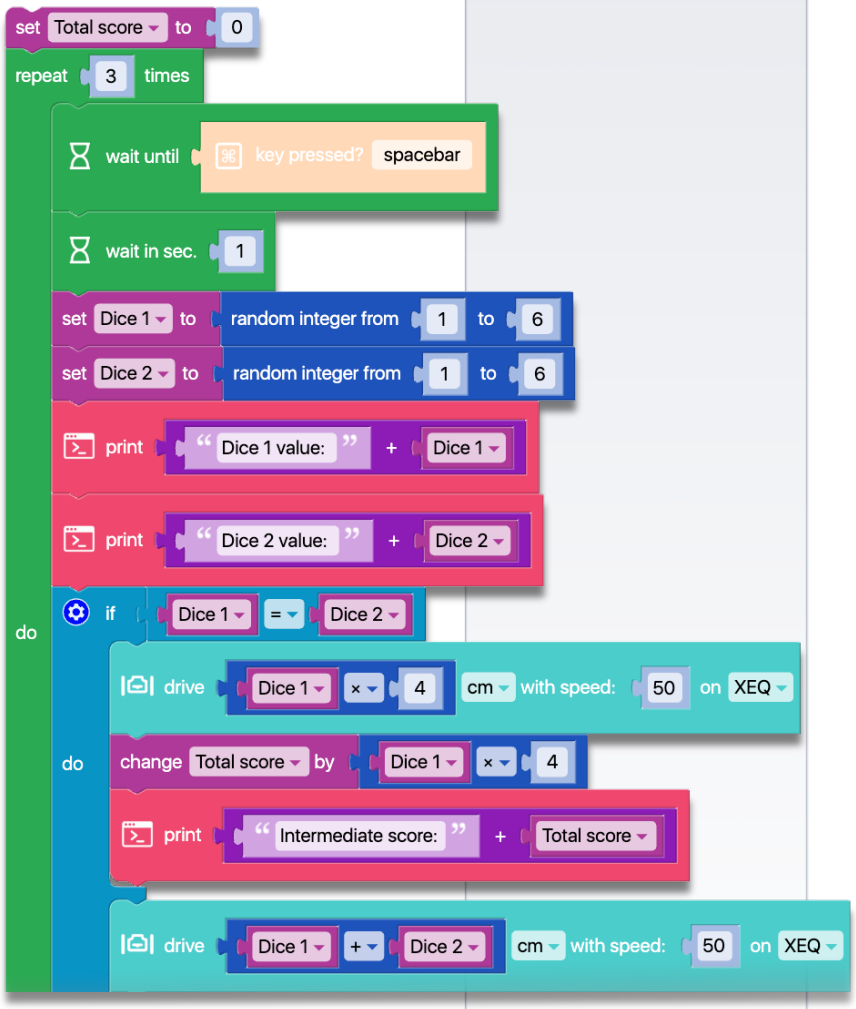

set Sum value to 0
wait until key pressed? spacebar
wait in sec. 1
set Number 1 to random integer from 1 to 6
print "Number 1 is:" + Number 1
wait until key pressed? spacebar
wait in sec. 1
set Number 2 to random integer from 1 to 6
print "Number 2 is:" + Number 2
wait until key pressed? spacebar
wait in sec. 1
set Number 3 to random integer from 1 to 6
print "Number 3 is:" + Number 3
wait in sec. 1
print "Sum value is:" + Number 1 + "+" + Number 2 + "+" + Number 3 + "=" + Number 1 + + Number 2 + + Number 3

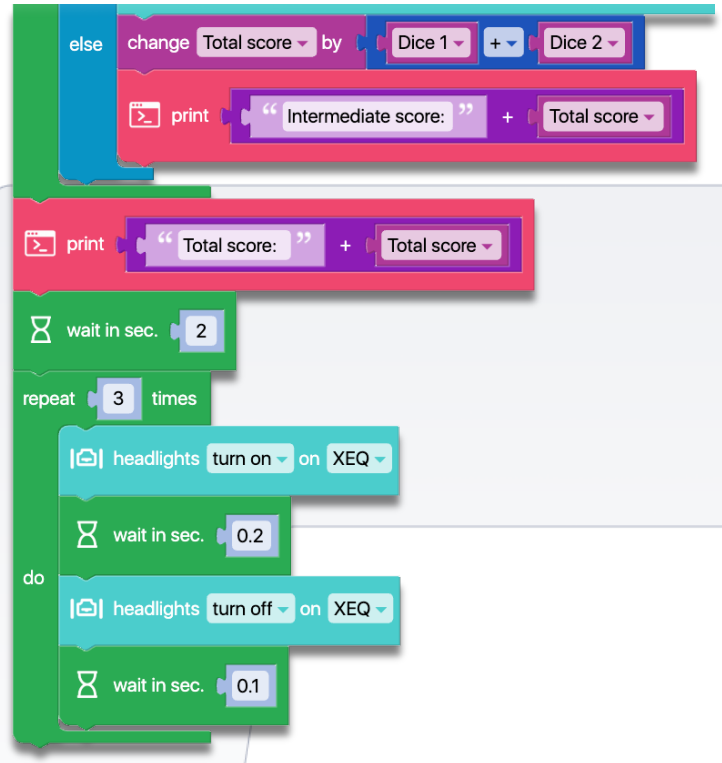
shape robotics
SMART LEARNING

Code 98:
Dice Game

- The program can function as a dice game where the player receives three rolls of two dice. If the numbers on the dice are the same, the Spin module moves a distance four times the value shown on the dice. Alternatively, if the numbers on the dice are not equal, the Spin module moves a distance equal to the sum of the two numbers. After three rolls, the program announces the total score, determining the player who has achieved the furthest distance with the Spin module as the winner. The values of the dice, the intermediate displacement, and the totals are displayed in the console output.

WATCH VIDEO


set Total score to 0
repeat 3 times
wait until key pressed? spacebar
wait in sec. 1
set Dice 1 to random integer from 1 to 6
set Dice 2 to random integer from 1 to 6
print Dice 1 value: + Dice 1
print Dice 2 value: + Dice 2
do
if Dice 1 = Dice 2
drive Dice 1 × 4 cm with speed: 50 on XEQ
do
change Total score by Dice 1 × 4
print Intermediate score: + Total score
drive Dice 1 + Dice 2 cm with speed: 50 on XEQ

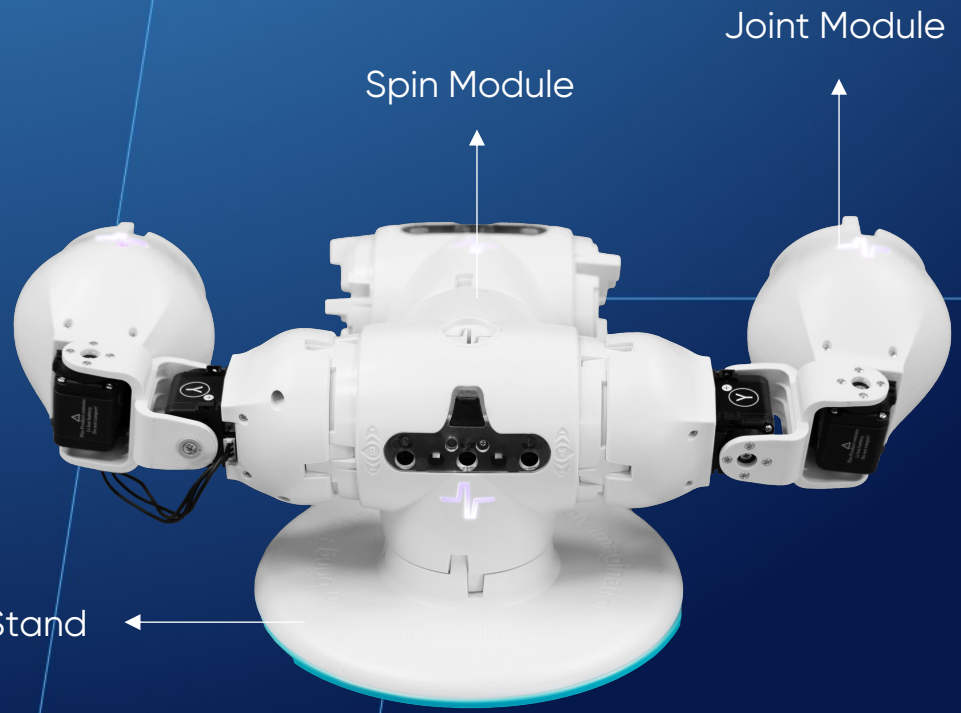

else
change Total score by Dice 1 + Dice 2
print Intermediate score: + Total score
print Total score: + Total score
wait in sec. 2
repeat 3 times
do
headlights turn on on XEQ
wait in sec. 0.2
headlights turn off on XEQ
wait in sec. 0.1

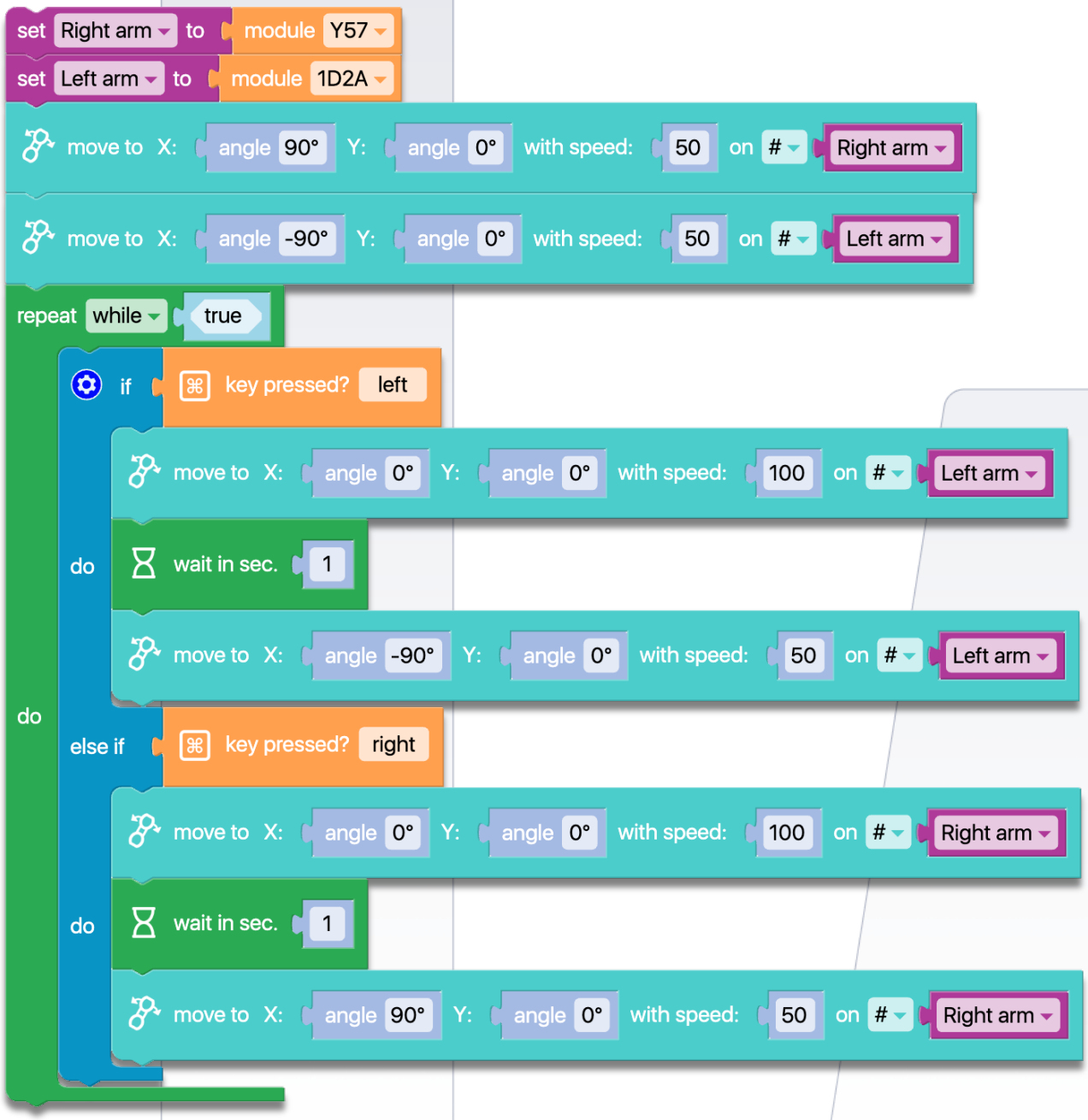
shape robotics
SMART LEARNING

Code 99:
Goalkeeper Game

- The program operates two Joint modules acting as a goalkeeper's hands. Pressing the left and right keys allows you to control these arms. However, it's important to note that due to the time required for them to reach the defensive position, precise timing is essential to initiate the movement command accurately. Shots will be aimed towards the left and right, corresponding to the positions of the goalkeeper's hands.

WATCH VIDEO



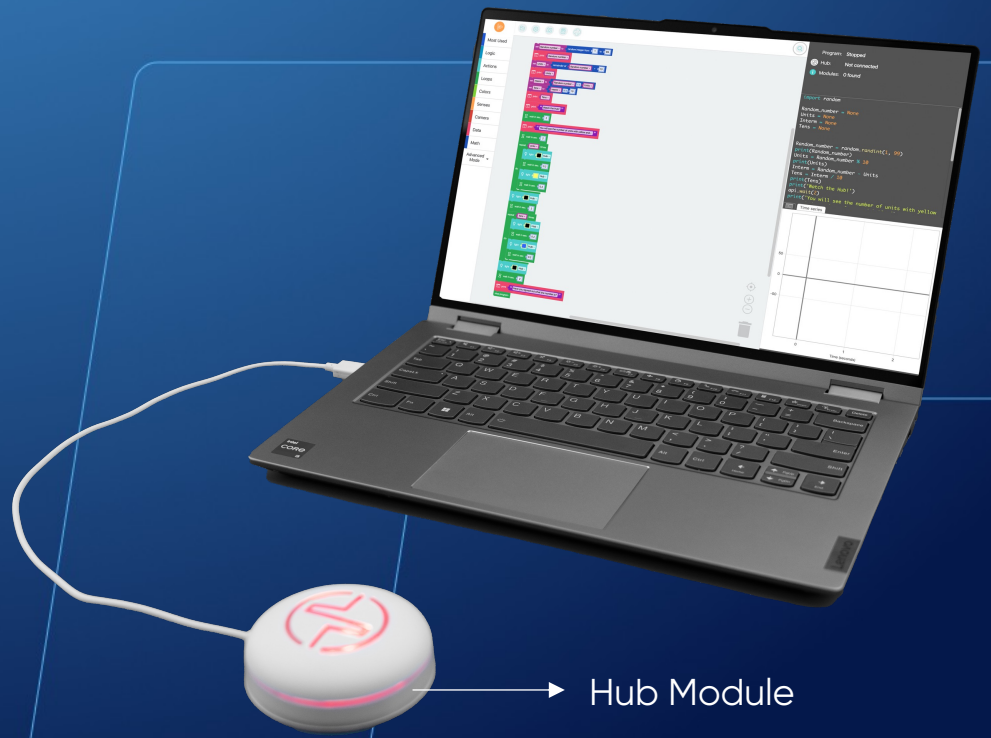

set Right arm to module Y57
set Left arm to module 1D2A
move to X: angle 90° Y: angle 0° with speed: 50 on # Right arm
move to X: angle -90° Y: angle 0° with speed: 50 on # Left arm
repeat while true
if key pressed? left
move to X: angle 0° Y: angle 0° with speed: 100 on # Left arm
do
wait in sec. 1
move to X: angle -90° Y: angle 0° with speed: 50 on # Left arm
do
else if key pressed? right
move to X: angle 0° Y: angle 0° with speed: 100 on # Right arm
do
wait in sec. 1
move to X: angle 90° Y: angle 0° with speed: 50 on # Right arm

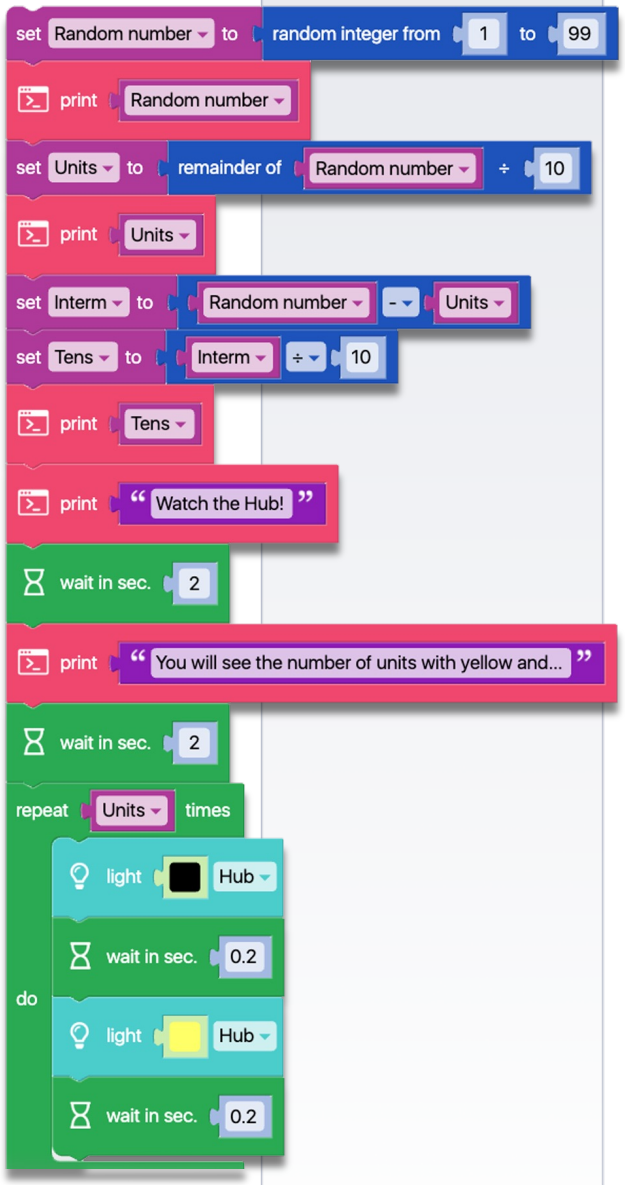
shape robotics
SMART LEARNING

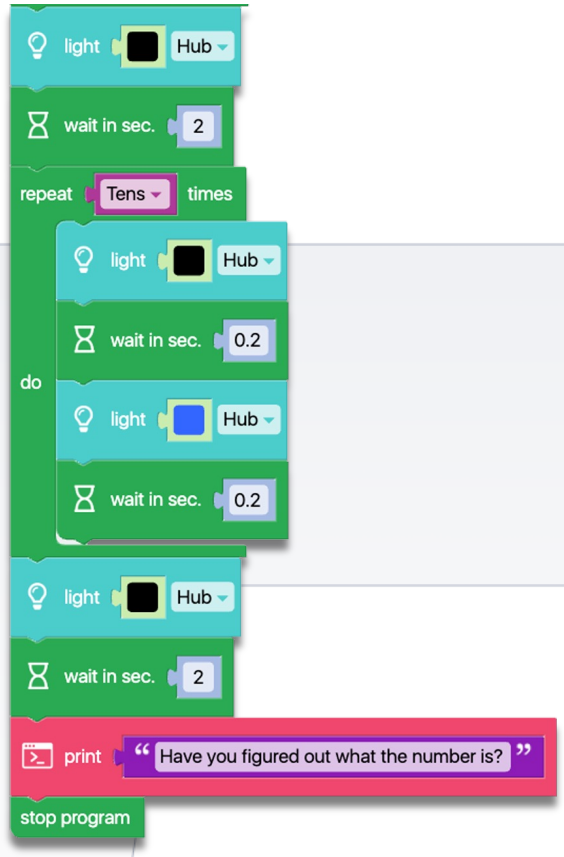
Code 100:
Guess the Number

- The program can also operate as a game. It randomly selects a number between 1 and 99 and displays it on the Hub by illuminating the corresponding units and tens. For example, if the number is 14, it will illuminate yellow four times and blue once. The goal of the game is to guess the number solely by observing the Hub. To increase complexity, the on and off intervals for the Hub light can be shortened.

WATCH VIDEO




set Random number to random integer from 1 to 99
print Random number
set Units to remainder of Random number ÷ 10
print Units
set Interm to Random number - Units
set Tens to Interm ÷ 10
print Tens
print "Watch the Hub!"
wait in sec. 2
print "You will see the number of units with yellow and..."
wait in sec. 2
repeat Units times
do
light Hub
wait in sec. 0.2
light Hub
wait in sec. 0.2


light Hub
wait in sec. 2
repeat Tens times
do
light Hub
wait in sec. 0.2
light Hub
wait in sec. 0.2
light Hub
wait in sec. 2
print "Have you figured out what the number is?"
stop program

shape robotics
SMART LEARNING

Sharing is Caring

Fuel our community's passion,
send your robot codes to hello@shaperobotics.com and let the joy spread!