

Model 11: Painting robot

Objectives and classification

Overview

The remote-controlled buggy from model 10 is converted into a painting robot.

The drawing of simple geometric figures can be automated: This requires time measurement of rotations (= angles) and straight lines (= line length).

Note: The tasks can be programmed most reliably with a USB connection to the fischertechnik BT Smart Controller, as transmission delays occur with a Bluetooth connection.



Topics

How can the buggy be turned through a specified angle? How can the buggy drive along a specified route?

Learning objectives

- Representation of angles and distances by times
- Measuring times with Scratch

Time required

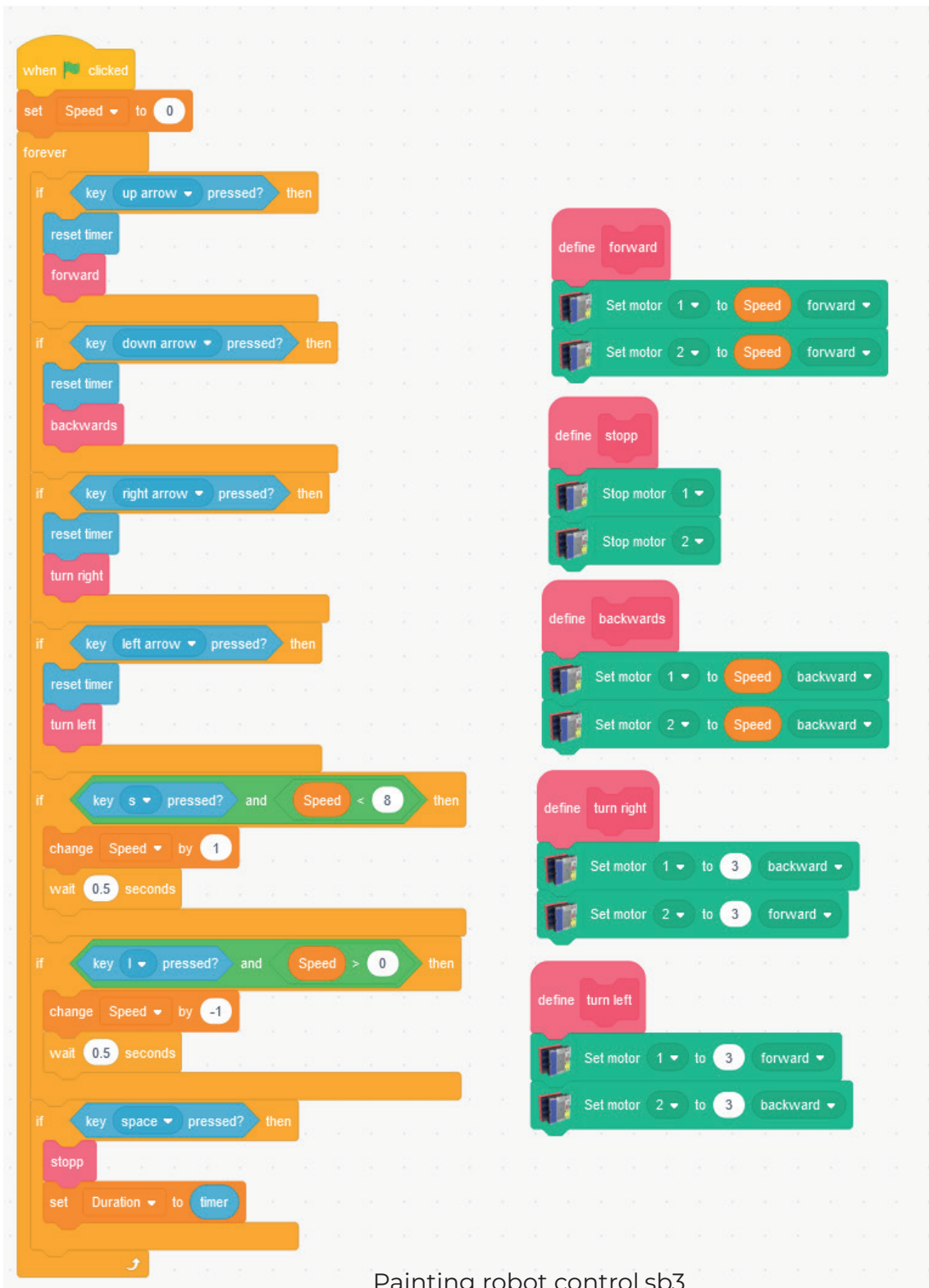
It takes about half an hour to assemble the painting robot. If the buggy from model 10 is still assembled, it can be converted and the construction time thus shortened. The first task „Control painting robot“ is a direct application of the remote control of the buggy programmed in model 10 (buggy). Task „Measuring the painting robot“ requires several measurements to determine the times required to turn through the necessary angles (120°, 90°) and to travel a certain distance. Task 3 is a direct application of the measurement results and is not difficult to program. One lesson should be sufficient to solve the tasks.

Solving the experimental task is a little trickier; it requires at least one lesson of its own. The solution can be supported by collecting interim results (such as the „painting strategy“ for the „House of Santa Claus“ or determining the required distances and angles) together.

Solutions and notes

Programming tasks Model 11: Painting robot

Solution steering of painting robot:



The image shows a Scratch script for controlling a painting robot. The main script is a 'when clicked' event that sets the speed to 0 and enters a 'forever' loop. Inside the loop, there are several 'if' statements for key presses: 'up arrow' (forward), 'down arrow' (backwards), 'right arrow' (turn right), and 'left arrow' (turn left). There are also 'if' statements for 's' (increase speed) and 'l' (decrease speed) key presses, both with a 0.5-second wait. A 'space' key press triggers a 'stopp' function and sets a 'Duration' timer. To the right, several functions are defined: 'forward' (set motors 1 and 2 to Speed forward), 'stopp' (stop motors 1 and 2), 'backwards' (set motors 1 and 2 to Speed backward), 'turn right' (set motor 1 to 3 backward, motor 2 to 3 forward), and 'turn left' (set motor 1 to 3 forward, motor 2 to 3 backward).

```

when clicked
  set Speed to 0
  forever
    if key up arrow pressed? then
      reset timer
      forward
    if key down arrow pressed? then
      reset timer
      backwards
    if key right arrow pressed? then
      reset timer
      turn right
    if key left arrow pressed? then
      reset timer
      turn left
    if key s pressed? and Speed < 8 then
      change Speed by 1
      wait 0.5 seconds
    if key l pressed? and Speed > 0 then
      change Speed by -1
      wait 0.5 seconds
    if key space pressed? then
      stopp
      set Duration to timer
  
```

define forward

```

Set motor 1 to Speed forward
Set motor 2 to Speed forward
  
```

define stopp

```

Stop motor 1
Stop motor 2
  
```

define backwards

```

Set motor 1 to Speed backward
Set motor 2 to Speed backward
  
```

define turn right

```

Set motor 1 to 3 backward
Set motor 2 to 3 forward
  
```

define turn left

```

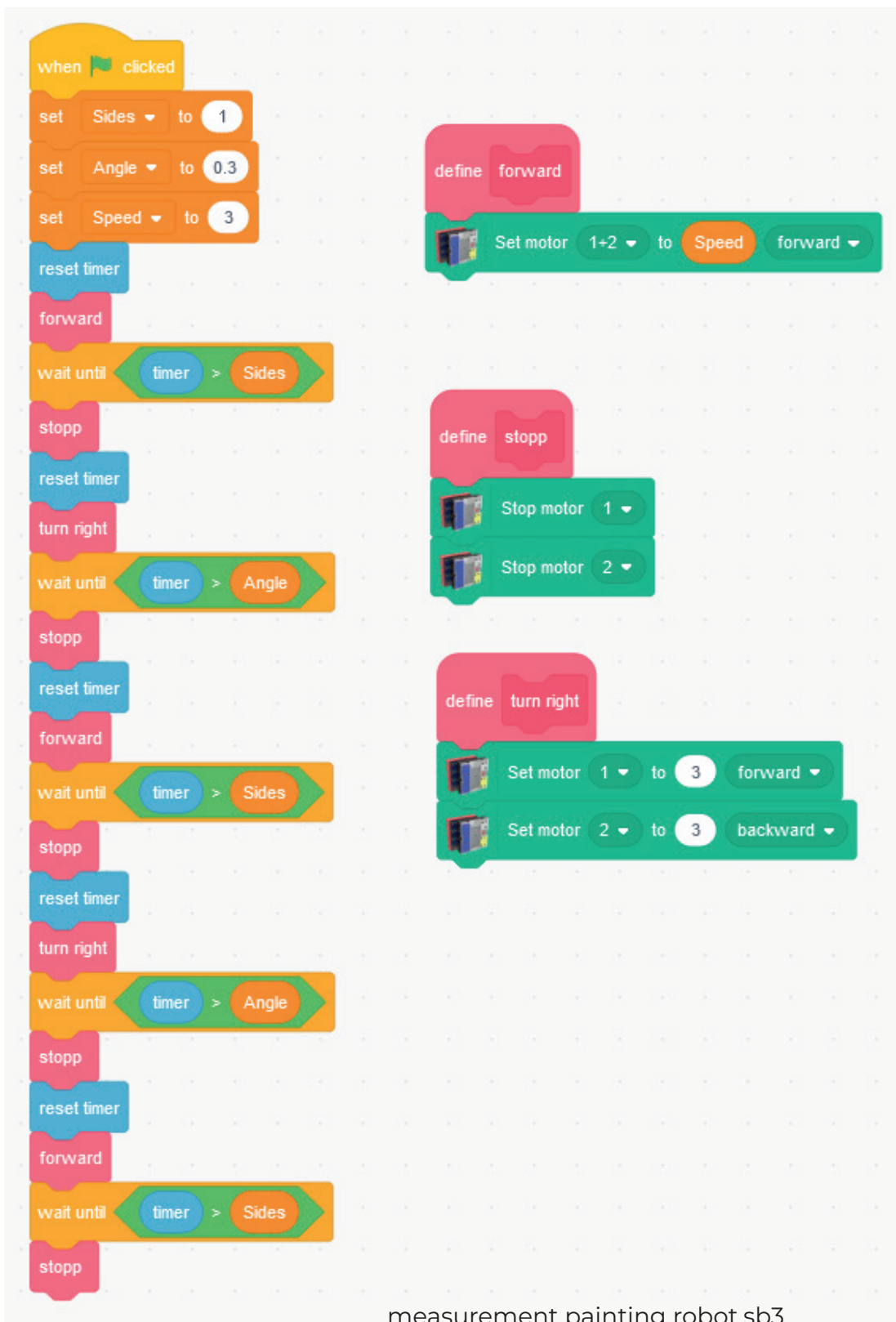
Set motor 1 to 3 forward
Set motor 2 to 3 backward
  
```

Painting robot control.sb3

Solutions and notes

Programming tasks Model 11: Painting robot

Solution Measurement painting robot:



```

when clicked
  set Sides to 1
  set Angle to 0.3
  set Speed to 3
  reset timer
  forward
  wait until timer > Sides
  stopp
  reset timer
  turn right
  wait until timer > Angle
  stopp
  reset timer
  forward
  wait until timer > Sides
  stopp
  reset timer
  turn right
  wait until timer > Angle
  stopp
  reset timer
  forward
  wait until timer > Sides
  stopp

define forward
  Set motor 1+2 to Speed forward

define stopp
  Stop motor 1
  Stop motor 2

define turn right
  Set motor 1 to 3 forward
  Set motor 2 to 3 backward
  
```

measurement painting robot.sb3