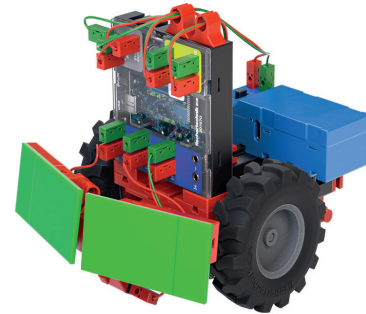


Model 12: Butler

Objectives and classification

Overview

The „Butler“ turns the buggy (model 10) into an autonomous driving robot: it learns to recognize black lines as boundaries, drive around obstacles and play a sound when it has reached its destination.



Topics

How can a vehicle recognize dark lines with a photo transistor? How can the vehicle recognize and avoid obstacles?

Learning objectives

- Programming a (partially) autonomous vehicle
- Detection of lines by means of a phototransistor
- Programming obstacle detection and avoidance

Time required

If the painting robot or buggy only needs to be rebuilt, the butler only takes a few minutes to assemble. If the model is assembled from the start, a construction time of 35 minutes can be expected.

The solution to the „Destination control“ task requires a few practical tests, but is simple. The „Reach target“ task uses a solution from model 3 (alarm system) by playing a sound. The „Line detection“ task is also easy to solve by making a small adjustment to the light barrier from model 3 (alarm system). Either the self-created program or the example program in the app can be used for this.

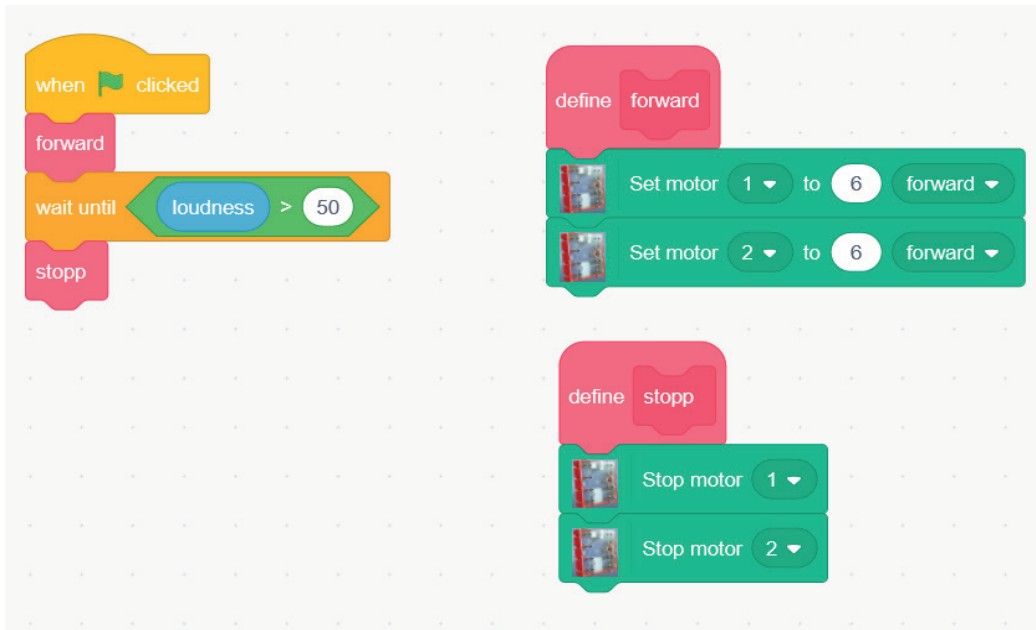
Students should be able to solve all three tasks in one lesson.

The experimental task „Avoiding obstacles“ is a little trickier and requires several tests. An additional lesson should be set aside for solving it.

Solutions and notes

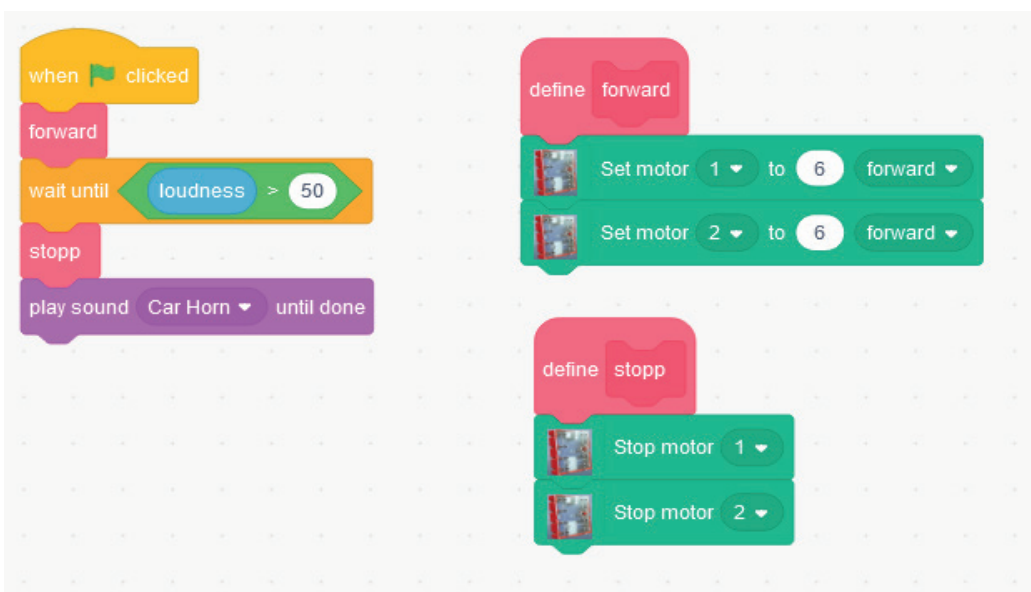
Programming tasks Model 12: Butler

Solution Go to destination:



go to destination.sb3

Solution Reach destination (horn):

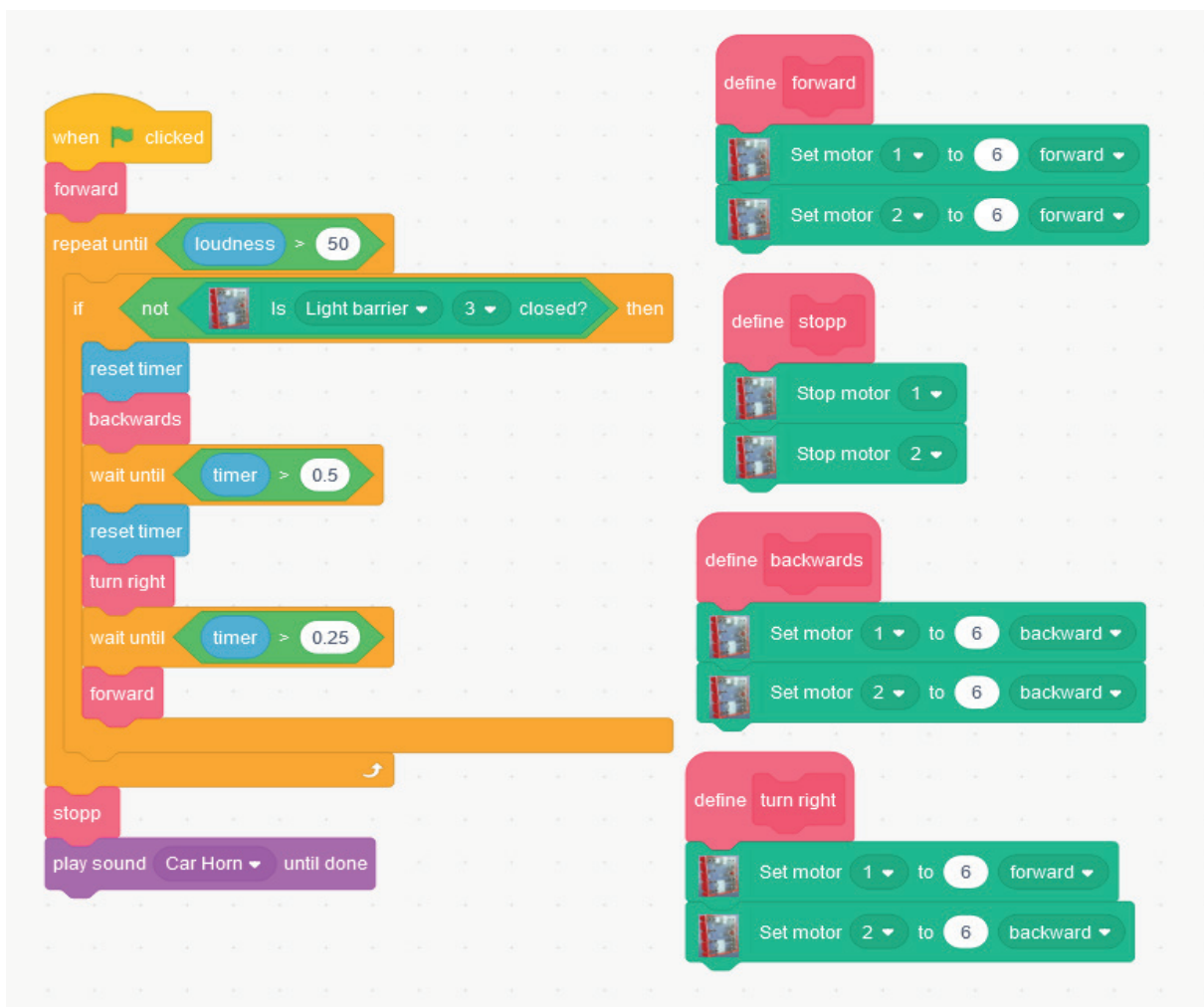


reach destination.sb3

Solutions and notes

Programming tasks Model 12: Butler

Solution Line recognition:



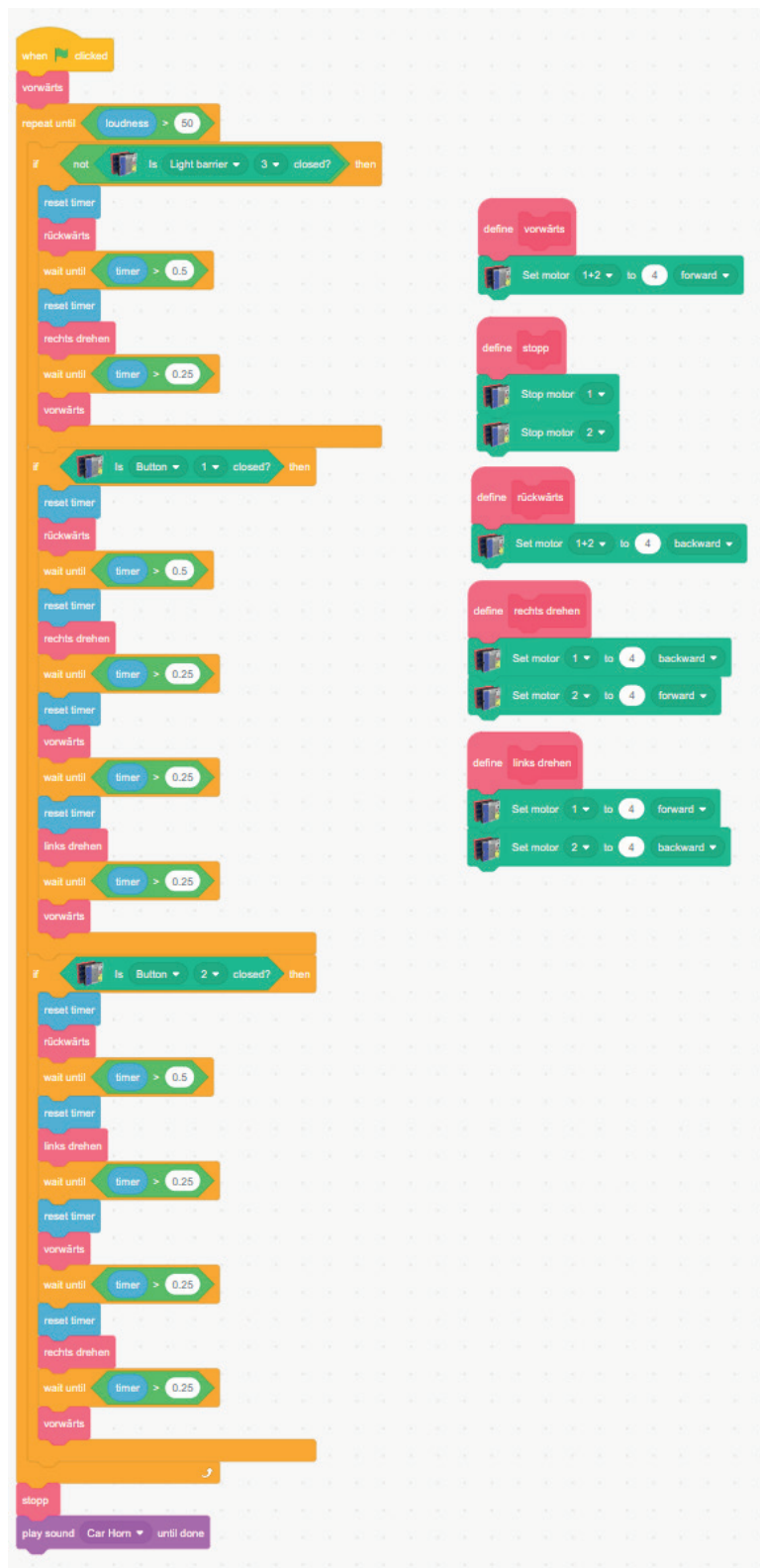
Line detection.sb3

Solutions and notes

Experimental task Model 12:

Butler

Solution Avoiding obstacles:



```

    When clicked
    vorwärts
    repeat until loudness > 50
    if not is Light barrier 3 closed? then
        reset timer
        rückwärts
        wait until timer > 0.5
        reset timer
        rechts drehen
        wait until timer > 0.25
        vorwärts
    if is Button 1 closed? then
        reset timer
        rückwärts
        wait until timer > 0.5
        reset timer
        rechts drehen
        wait until timer > 0.25
        reset timer
        vorwärts
        wait until timer > 0.25
        reset timer
        links drehen
        wait until timer > 0.25
        vorwärts
    if is Button 2 closed? then
        reset timer
        rückwärts
        wait until timer > 0.5
        reset timer
        links drehen
        wait until timer > 0.25
        reset timer
        vorwärts
        wait until timer > 0.25
        reset timer
        rechts drehen
        wait until timer > 0.25
        vorwärts
    stop
    play sound Car Horn until done
  
```

define vorwärts
Set motor 1+2 to 4 forward

define stopp
Stop motor 1
Stop motor 2

define rückwärts
Set motor 1+2 to 4 backward

define rechts drehen
Set motor 1 to 4 backward
Set motor 2 to 4 forward

define links drehen
Set motor 1 to 4 forward
Set motor 2 to 4 backward

avoiding obstacles.sb3