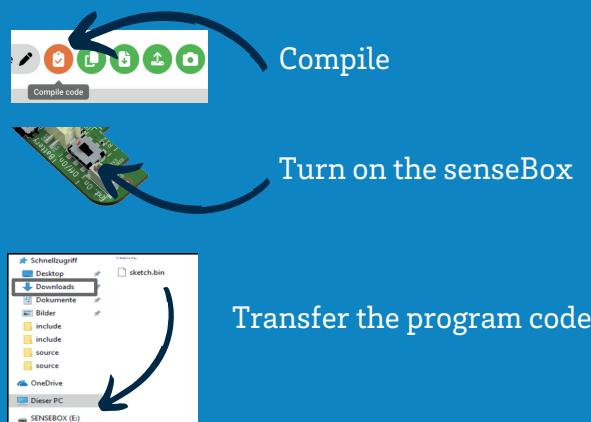


## Info: Uploading the program code



### Alternative: senseBox Connect App

Instructions for transferring the program code using a tablet:



## iCODE Loops

# Motion Detector for Nature Reserves



Level: ★ ★ ★



Certain places are closed to the public for nature conservation reasons, but people still visit them. A motion detector is often enough to prevent these unwanted visitors. Of course, the light should only stay on until the person has left. Find out how this works in this project!

## The code isn't working? Troubleshooting tips

- Are your cables plugged in exactly as shown in the illustration?
- Are your command blocks really connected like small "puzzle pieces"?
- Have you deleted all blocks that are not connected to your main block?
- Have you compiled the latest version of your program code and uploaded it again after making changes in Blockly?

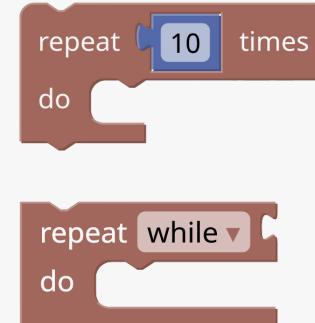
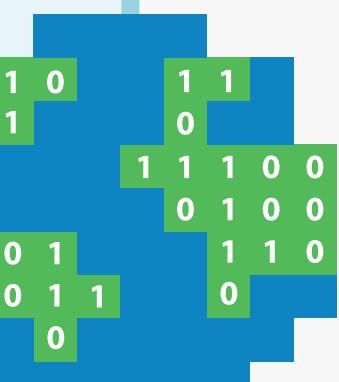
Still having trouble?

Then get in touch with a mentor!

INFO: FOR-LOOPS OR REPETITIONS

Loops are useful control structures in computer science for specifying how long the motion detector should remain illuminated. They are helpful whenever you want to execute one or more instructions up to a certain point in time or when you want to execute a certain number of repetitions.

There are different types of loops:

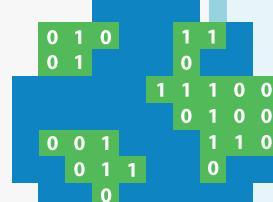


Everything within the loop is repeated **10 times**. Afterwards, everything before/after the block is executed again.



Everything **within the loop** is executed until the inserted condition is **true**. As soon as the condition is false, the loop is exited.

A variable is defined with a **start value** and a **limit value**. It is then **incremented** by a specified amount each time. The variable can also be used within the statement.



< iCODE > MS



Kofinanziert von der  
Europäischen Union  
Ministerium für Wirtschaft,  
Industrie, Klimaschutz und Energie  
des Landes Nordrhein-Westfalen



## Step 1A

1. For programming: [blockly.sensebox.de](https://blockly.sensebox.de)
2. In **Setup**, some components need to be activated once at the beginning of the program.



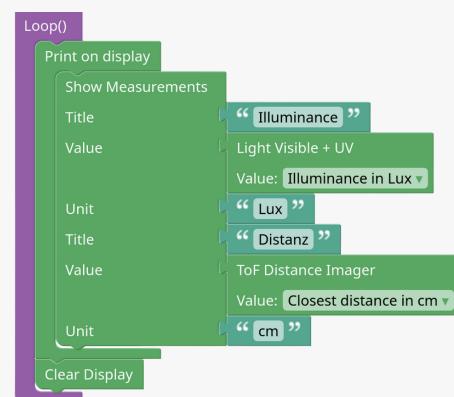
3. Both the **display** and the **RGB LED** (on board) have to be **initialized** in the setup:



## Step 1B

1. To display the measurements on the screen, you need the blocks "**Print on display**" and "**Show Measurements**" in the infinite loop.

2. Now use the **blocks for the brightness and distance sensors** to read the measurements, and label each value with a title and a unit:

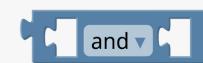


Test your code!



## Step 2B

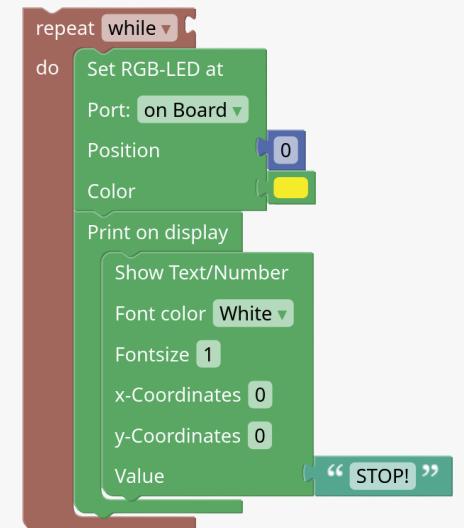
1. Now, the **condition** must be extended with an **AND** (category logic) so that the **RGB LED** only lights up when, in addition to darkness, there are people **near the distance sensor**:



2. On the left side of the operator is the statement that the **brightness is < 10 lux**. On the right side is the statement about distance that has already been used in step 2A.



3. As long as this condition is met, a warning should be shown on the **display** in addition to the illuminated **LED**. For this, you will also need the blocks "**Print on display**" and "**Show Text/Number**". You can find the text field for "**STOP!**" in the "**Text**" category.

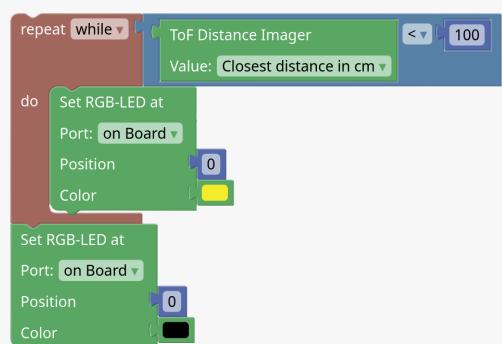


## Step 2A

To keep the LED lit as long as someone is near the nature reserve, you need a loop:

**Repeat while:** the distance of the ToF sensor is **less than (<) 100 cm**  
**Do:** The RGB LED lights up yellow.

Outside of this loop, the LED should not light up.



4. Now combine all the programming steps and test your code!  
You can influence the brightness and distance by covering the brightness sensor with your fist or holding a sheet of paper at different distances in front of the distance sensor.

Test your code!