

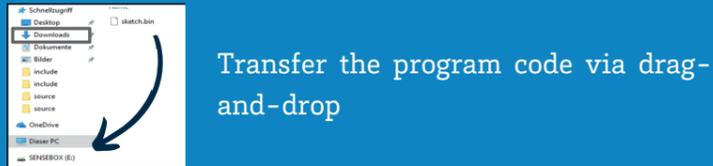
## Info: Uploading the program code



Compile



Turn on the senseBox



Transfer the program code via drag-and-drop

Alternative: senseBox Connect App  
Instructions for transferring the program code using a tablet:



## Smart Street Lighting



Streetlights bring light into the darkness—but they also consume a lot of energy! In Germany alone, up to four billion kWh of electricity are consumed each year to light streets, paths, and public spaces. To change this, in this project you will learn how to build and control smart street lighting.

## 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!

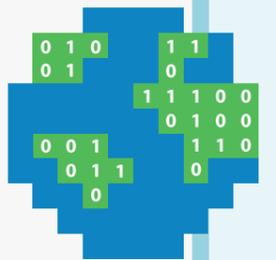
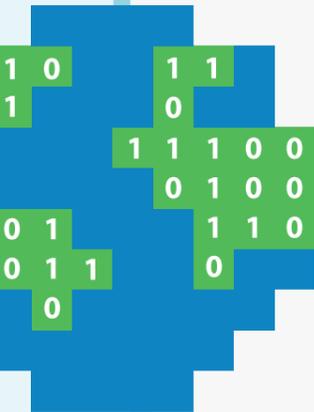
## iCODE Propositions

### INFO: PROPOSITIONAL LOGIC

"Turn on the lantern as soon as it gets dark **AND/OR** there are people nearby." The choice between **AND & OR** can make a big difference in whether a command is executed or not: In computer science, individual statements/propositions are evaluated as **true** (correct) and **false** (incorrect) and linked using the logical operators **AND & OR**. Here is an example of the conditions under which the command (turn on the lantern) is executed:

It is dark	...	People are nearby	Result
true	AND	true	true (lantern on)
false	AND	false	false (lantern off)
true	AND	false	false (lantern off)
true	OR	true	true (lantern on)
false	OR	false	false (lantern off)
true	OR	false	true (lantern on)

With the **AND** operator, both propositions must be true in order for the command to be executed.  
With the **OR** operator, it is sufficient if one of the two statements is true.





### Step 1A

1. For programming: [blockly.sensebox.de](http://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:

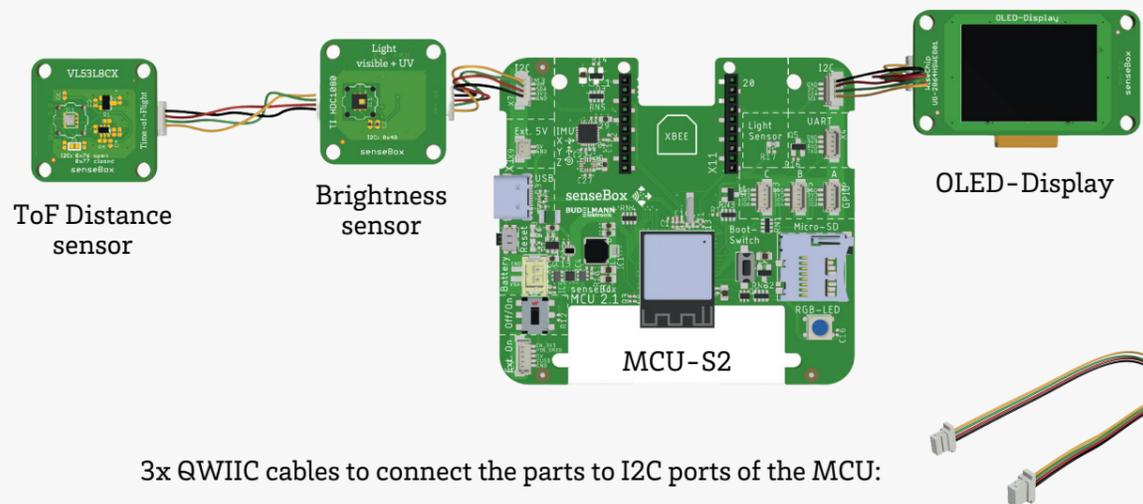


## Smart Street Lighting

Connect the OLED display and the sensors to the microcontroller.

- 1) Create a program so that the measured values of the brightness and distance sensors are shown on the display.
- 2) Expand your program code so that the RGB LED lights up as long as it is dark and people are nearby.

### Hardware-Setup



### Step 2B

1. Now, the **condition** must be expanded to include the operator **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 already used statement: the **illuminance (brightness) is < 10 lux**.



3. On the right side is the new condition: The measured **distance** of the sensor must be **less than 200 cm** to execute the "do" action:



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.

### 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 ToF distance sensors** to read the measurements, and label each value with a title and a unit:

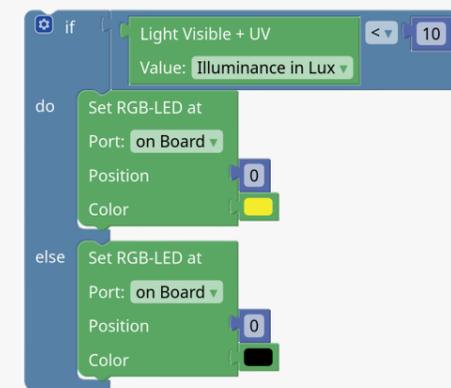


Test your code!

### Step 2A

To make the LED light up when it gets dark, you need an **if, do... else condition** from the category "Logic":

**If:** The **illuminance** is less than (< - category Logic) **10** (category Math) lux...  
**Then:** The **RGB-LED** lights up **yellow**...  
**Else:** The **RGB-LED** doesn't light up (color: black).



Test your code!