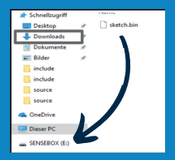
Info:
Uploading the program code

Compile

2x Reset-Button drücken



Programmcode per Drag-and-Drop übertragen

Alternative: senseBox Connect App
Anleitung zur Übertragung des Programmcodes mit dem Tablet:

senseBox

Car Traffic Counter

Blockly | Level: ★★★ | 30 min.

Cities are often full of cars that produce emissions and noise. This is why discussions about car-free city centers keep coming up. But how many cars are actually in the city? In this project, you will build a traffic counter and find out!

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, after making changes in Blockly, uploaded it again?

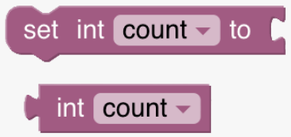
Still having trouble?
Then get in touch with a mentor!

iCODE Variablen

INFO: VARIABLEN

When programming the traffic counter, you will repeatedly need to access certain values. To make this easy, computer science offers the helpful concept of variables. They are like a box with a name on it – inside this box you can store different things, such as numbers or text, and retrieve them later.

Depending on what you want to store in a variable, you must choose the correct data type:



Character (char): For single text characters
Text (string): For whole words or sentences
Number (int): For numbers from –32,768 to +32,768
Large number (long): For numbers from –2,147,483,648 to +2,147,483,648
Decimal number (float): For decimal numbers (e.g., 25.3)
State (boolean): true or false

Variables can change their value during the course of the program. For example, the variable "count" can increase by one every time a car passes by.

MExLab ExperiMINTe | re:edu reengineering education | openSenseLab

<iCODE>MS

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

Car Traffic Counter

Connect the OLED display and the sensors to the microcontroller.

1) Create a program so that the measurement values from the distance sensor are shown on the display.

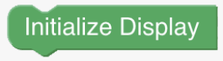
2) Extend your program code so that the number of detected cars is also shown on the display.

Step 1A

1. For programming: blockly.sensebox.de

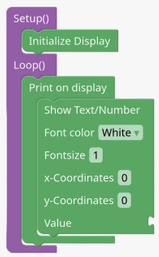
2. In Setup, some components need to be activated once at the beginning of the program.

3. The display has to be initialized in the setup.



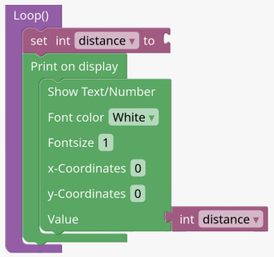
Step 1B

1. To display the measurement values on the screen, you need the "Print on display" and "Show Text/Number" blocks in the loop.

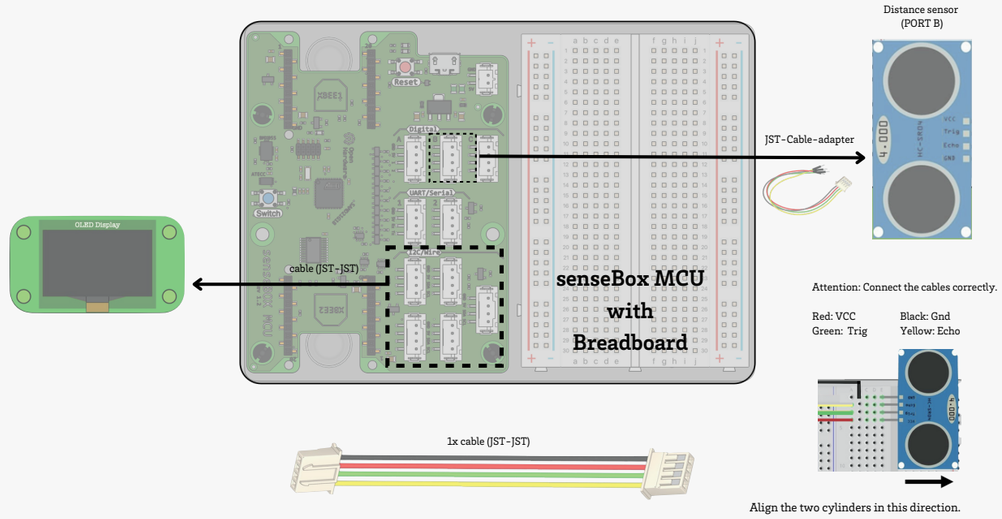


2. Instead of inserting the sensor block directly into "value", you define a variable and then insert this variable into "value".
To do this, select the "Create Typed Variable" block from the Variables category.

3. Define the variable as a number (int), name it "distance", and insert it into the loop.

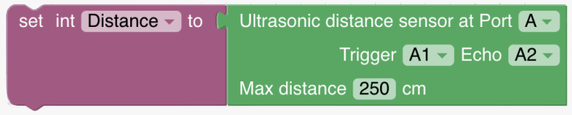


Hardware-Setup



Step 1C

The variable now has to be assigned a measurement value. Since the distance is measured with the ultrasonic distance sensor, this sensor block is connected to the variable "distance".

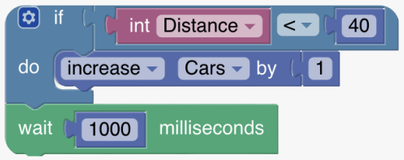


Test your Code!

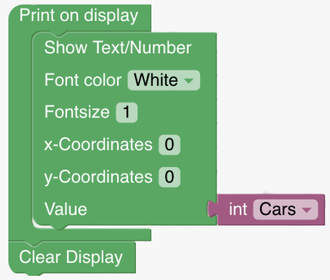
Step 2

1. The measuring device now detects distances, but not cars yet. To count them, you need to create a new variable (int) called "car". Since the car count should start at zero, you set this in Setup (category Math).

2. With an "if – do" condition, you define when a new car should be counted:
If the distance is less than 40 cm, then the car variable increases by one.
To prevent a car from being counted twice, add the "wait 1000 milliseconds" block.



3. To display the number of cars on the screen, add the "Show Text/Number" block again and insert the variable "cars" as the value. By regularly clearing the display, you ensure that the most up-to-date and correct distance to the passing car is shown. Also adjust the y-coordinate so that the distance and the count appear one below the other.



This completes the programming. Now simulate passing cars using objects and test whether everything works.

Test your Code!