Lesson 2:

Sensing = data

Subject: STEAM

Grade(s): 5th and up

Duration: 45 Minutes

Difficulty: Beginner

Lesson Objectives

At the end of this lesson, students will be able to:

- Recognize and use the code blocks for operating the sensors
- Show the data from the sensors on the display of the mBot2
- Build your own computer program in mBlock for the mBot2 to control the sensors and show the data on the display

Overview

A robot can be a fun toy, but did you know that you can also use a robot in research? A robot has several sensors (=senses) with which it can perceive its surroundings and also collect data. For example, you can have the mBot2 measure sound and temperature. You can read the data on the screen of the mBot2 and use it for research into sound density or the temperature of the classroom, for example.

Focus

At the end of this lesson, students will know:

- What sensors the mBot2 has
- How the sensors of the mBot2 work
- What the difference is between live mode and upload mode
- How to show the data from the sensors on the display of the CyberPi

Pre-lesson Checklist

What do you need?

- PC or laptop (with USB output) with the mBlock software installed, the web version (also for ChromeBook), or a tablet with the mBlock app installed
- The mBot2 with a CyberPi
- A USB-C cable or Makeblock Bluetooth dongle

Lesson plan

This lesson consists of four steps and takes a total of 45 minutes.

Duration	Contents
5 minutes	1. Warming up • Sensors and data in everyday life. • What is the CyberPi?
10 minutes	2. Hands-on • Getting acquainted with the sensors of the mBot2. • Extend and test some programming examples of the sensors. • Showing the data from the sensors on the display. • Difference between live mode and upload mode.
25 minutes	3. Trying out • Writing your own program for the robot.
5 minutes	4. Wrap-up • Showtime: show what you did with your robot in a fun, short movie for later discussion. • If your teacher allows, share the end result on social media with the hashtag #mBot2data • Reflection: What are you most proud of? What would you like to improve about your robot?

makeblock education

Activities

1. Warming up (5 min)

Step 1: Warming up

This step consists of two parts:

1. Sensors and data in everyday life
2. What is the CyberPi?

1. Sensors and data in everyday life

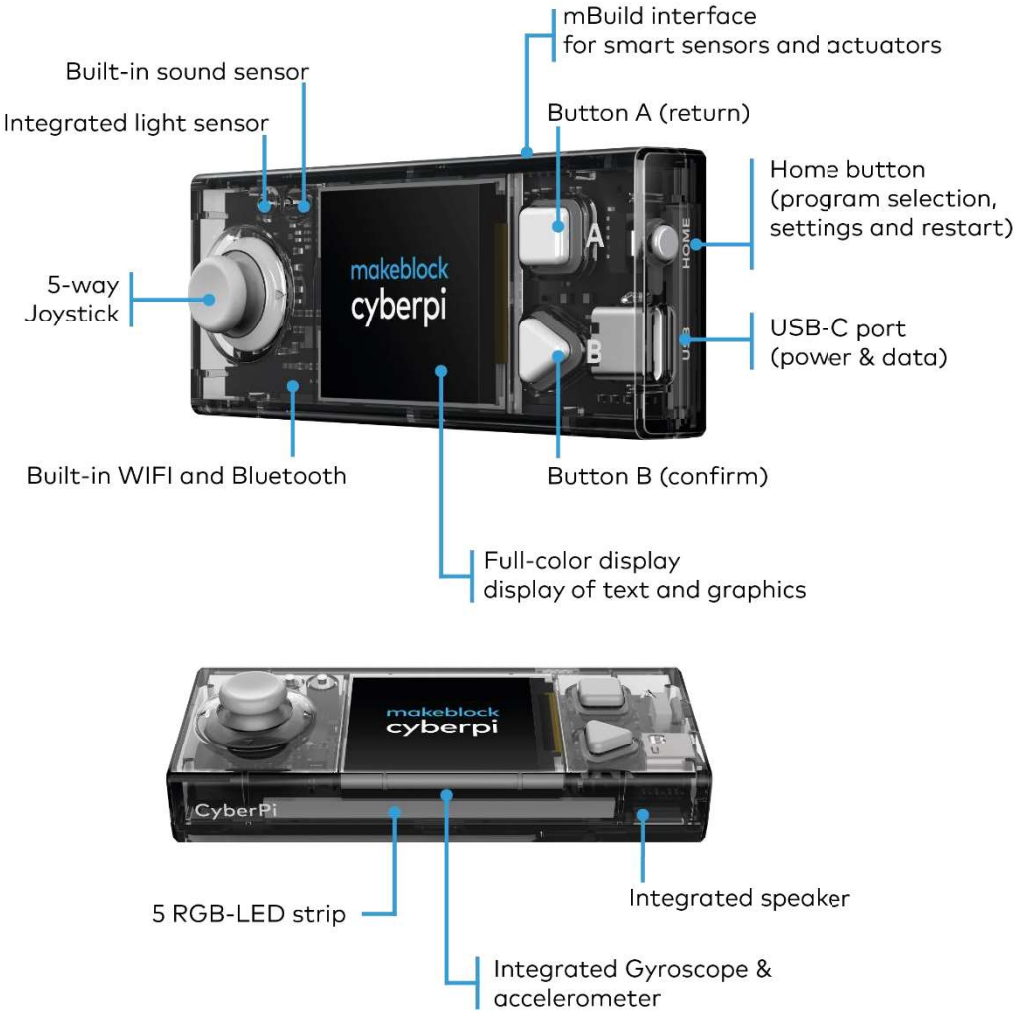
Sensors that collect data are found in many different places in everyday life. Much more than you probably think at first. For example, with the help of sensors the exterior lighting of your house is turned on automatically when it gets dark outside. Or the heating is automatically turned off when it gets too hot in the classroom. The amount of outdoor light and the heat in the classroom is all recorded using sensors. Can you and your classmates think of any other examples?

2. What is the CyberPi?

To get the mBot2 to work, you need to write a computer program. You do that in the code editor mBlock. The computer program you write consists of a series of commands that the mBot2 has to execute.

Mounted on the mBot2 there is a CyberPi. A CyberPi is a small, programmable microcomputer. You give the CyberPi commands using the code blocks in mBlock. The CyberPi passes these commands back to the mBot2. You can remove the CyberPi from the mBot2.

The CyberPi has many different functions, such as a microphone, a speaker and a joystick. There are also many sensors on the CyberPi. Take a look at the image below.



2. Hands-on (10 min)

Step 2: Hands-on

This step consists of four parts:

1. Getting acquainted with the sensors of the mBot2.
2. Extend and test some programming examples of the sensors.
3. Display the data from the sensors.
4. Difference between live mode and download mode.

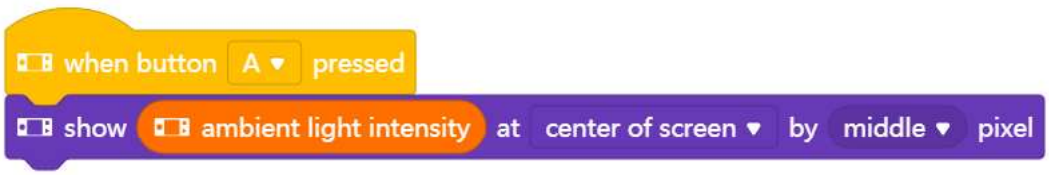
makeblock education

1. Getting to know the mBot2 sensors

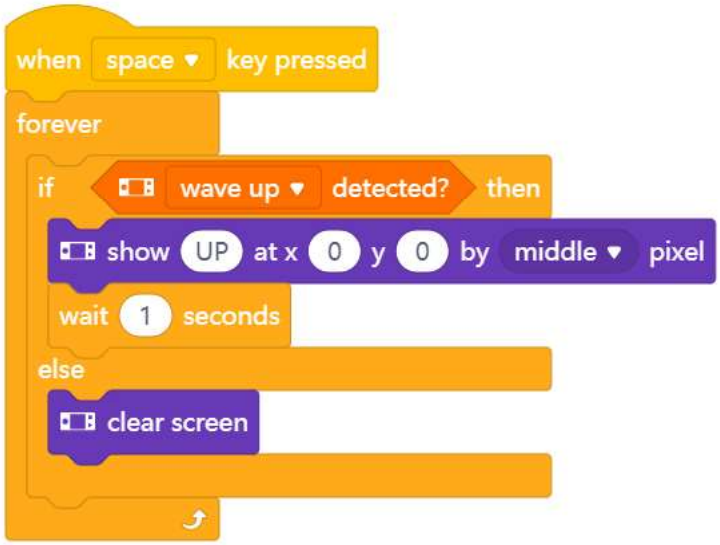
Every robot works with sensors. Sensors can be compared with your senses (=taste, touch, smell, hearing, sight). Through these sensors the mBot2 'sees' its environment. There are different types of sensors that can make the mBot2 'see', such as:

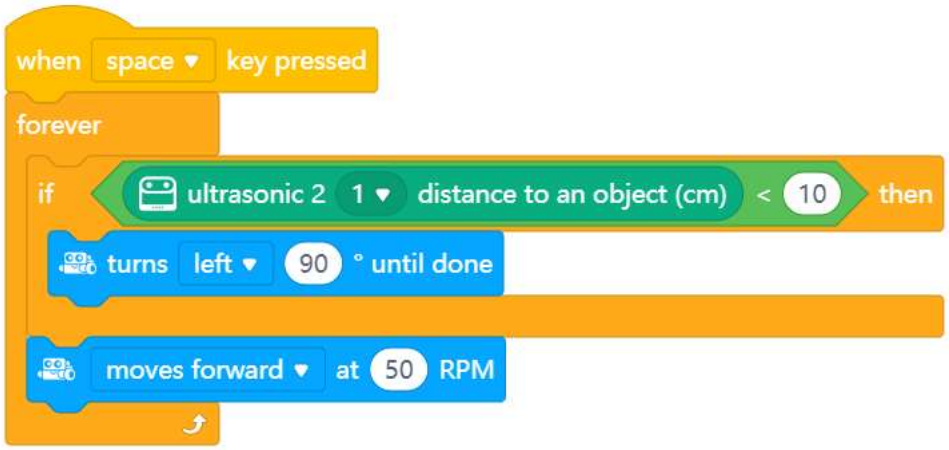
- Light sensor
- Sound sensor
- Gyroscope accelerometer
- Quad RGB sensor
- Timer

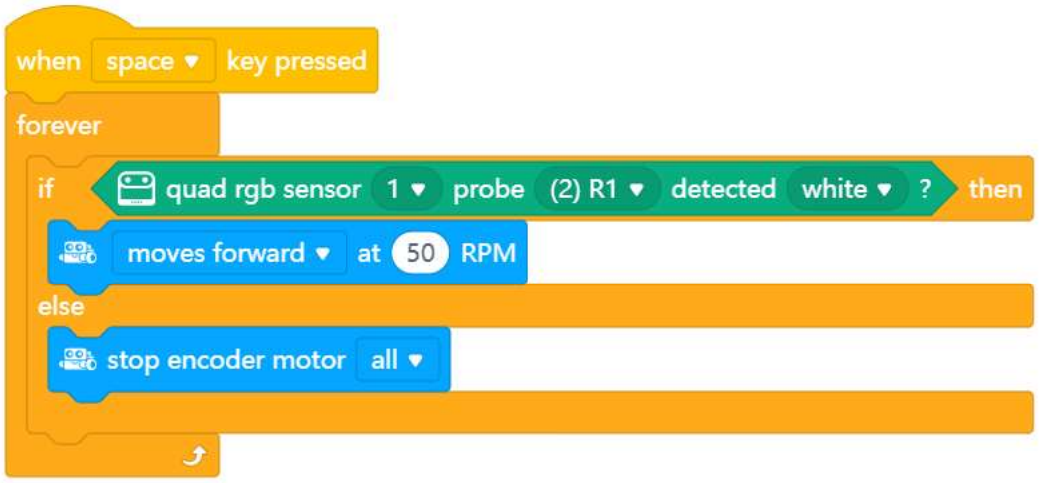
The table below lists the sensors of the mBot2. Each sensor is accompanied by a brief explanation and programming example.

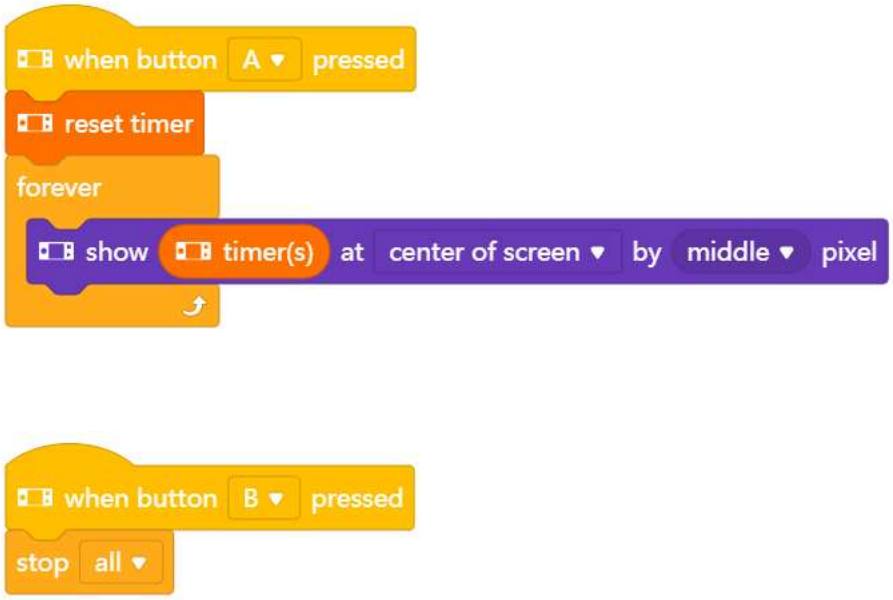
<table>
<tr><th>Sensor type</th><th>What does this sensor do?</th></tr>
<tr><td rowspan="3">Light sensor</td><td>Light has a certain strength (=light energy). The light sensor is a device that converts light energy into electrical energy.
A light sensor is often used, for example, to:
• Automatically adjust the brightness of the display of smartphones to the environment;
• To control the lighting in homes or to automatically switch on headlights of vehicles
In the programming example below, the light energy of the environment is shown on the display of the CyberPi. You can extend the programming example so that the mBot2 stops driving, for example, when the light energy of the environment is below a certain value, as when it gets dark.</td></tr>
<tr><td>Programming example</td></tr>
<tr><td> 
</td></tr>
</table>

Sensor type	What does this sensor do?
Sound sensor	Sound is a mechanical vibration that propagates in waves. In the air these are pressure and density fluctuations. If the vibration is in the audible range (between 16 and 20000 vibrations per second) and sufficiently intense, we can hear it as tones or sounds. The power of the sound is called the sound intensity. The sensor in the microphone converts the sound into an electrical signal that can be evaluated in terms of pitch and power. A sound sensor is often used for the following things: • Telephone calls, sound recording • Voice-controlled assistance systems, such as on a cell phone or in a Smart Home In the programming example below, the sound intensity of the environment is shown on the display of the CyberPi. You can extend the programming example so that the mBot2 constantly measures the sound intensity during a tour of the classroom and shows it on the display.
	Programming example
	

Sensor type	What does this sensor do?
Gyroscope & accelerometer	A gyroscope measures tipping movements, more precisely the speed of turning/tipping movements. An accelerometer measures the change in velocity. These sensors can be implemented as micromechanical components on an electronic component. Both sensors provide different information about the position in space. A gyroscope is often used to, for example: • Keep ships stable on the high seas or • Balance a Segway or a "hoverboard" so that you don't fall over too quickly with it Acceleration sensors, in turn, indicate whether a smartphone is facing up or down, or whether a vehicle is having an accident (violent, very rapid change in speed) - in order to then trigger the airbags. In the programming example below, the tilt movement of the mBot2 is shown on the display. You can extend the programming example so that the mBot2 constantly measures and records the tipping motion on the display during a lap around the classroom. You will learn more about this sensor and its capabilities in Lesson 6.
	Programming example
	

Sensor type	What does this sensor do?
Ultrasonic sensor	Sound, which is vibration in the form of variations in density and pressure, propagates in air at a constant speed in the form of waves (speed of sound in air approx. 334 m/s). The higher the vibration, the higher the sound appears - up to a limit above which people can no longer perceive this sound. These very high vibrations are called ultrasound. Since the speed of sound in the air is known, sound can be used to determine distances to objects; to do this, one emits a sound and measures the time until this sound is reflected back from the object. This reflected sound wave is also called an echo. Ultrasounds usually are used for this purpose. For example, an ultrasonic sensor is often used: • for an imaging procedure, e.g. in pregnancy (each pixel is a distance measurement) or • in the control of robots to prevent collisions In the programming example below, the ultrasonic sensor is used to prevent the mBot2 from driving into an obstacle. When the mBot2 is less than 10 cm from an obstacle, the robot makes a 90° turn to the left and then simply continues driving. You can extend this programming example so that the mBot2 moves randomly through a classroom without bumping into tables and chairs. You'll learn more about this sensor and its capabilities in Lesson 4.
	Programming example
	

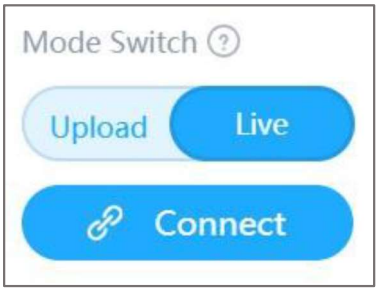
Sensor type	What does this sensor do?
Quad RGB Sensor	The Quad RGB sensor consists of four individual light and color sensors. They measure the intensity of light entering the sensor from red, green and blue areas of the light spectrum. This allows the sensor to detect colors of objects directly in front of it, like markings on the ground, and also allows the robot to follow a black line for orientation. A Quad RGB sensor is widely used in applications such as: • show a warehouse robot a way through the warehouse (also different colors for different paths), or • ensure that exactly the right color is used during painting In the programming example below, the mBot2 drives forward when the robot sees the color white. You can extend the programming example so that the mBot2 drives a route through the classroom, stops, or turns based on different colors. You will learn more about this sensor and its capabilities in Lesson 5.
	Programming example
	

Sensor type	What does this sensor do?
Timer	The timer is a kind of stopwatch that tells the time in seconds since the CyberPi was turned on or reset. This counter can also be set to zero with a command to make time measurements easier. For example, you can use the timer to have a race of several mBots against each other and measure exactly how fast they are at the finish line. The following programming example shows you how to set and use the timer. Pressing the 'A' key resets the timer to zero and then shows it permanently in the display. You can end the display and the program as a whole by pressing 'B'. Try to extend the programming example so that the timer starts counting when the mBot2 starts moving.
	Programming example
	

2. Testing and extending programming examples for the sensors

In the table above, there is a programming example for each sensor. You are going to recreate these programming examples in mBlock and test them. Think of an extension for two of the programming examples. The table above already mentions an extension suggestion for every programming example. Maybe you know a much nicer extension!

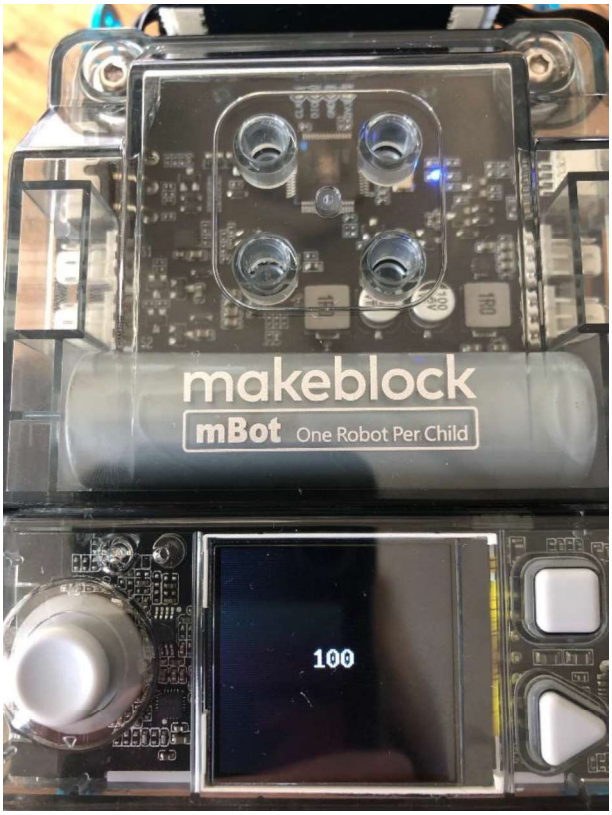
While writing a computer program, you can immediately test what you are creating. You do this by using live mode. You select live mode by moving the Mode switch to the right. Take a look at the image below.



3. Displaying the data from the sensors

In testing the programming examples, you have seen that each sensor shows something on the CyberPi's display. For example, the light sensor records the light intensity of the environment. The timer registers the time, and the gyro sensor keeps track of which way the mBot2 tilts. Everything that a sensor registers is called data.

You can show the data from a sensor on the display of the mBot2. In the programming examples you have seen that you can display the data with, for instance, a number or a text.



You can display the data from the sensors in different ways on the display of the mBot2. You can use different code blocks for this. These code blocks can be found in mBlock under the category 'Display'.
In the diagram below you see three examples of how to show data on the CyberPi's display. Reproduce the programming examples and test in live mode. See what happens on the display of the mBot2.

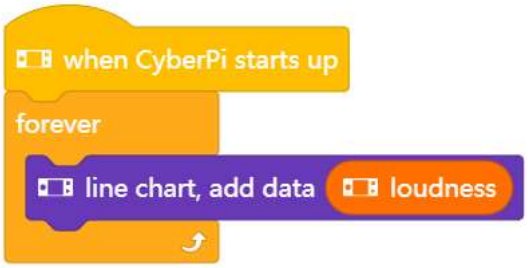
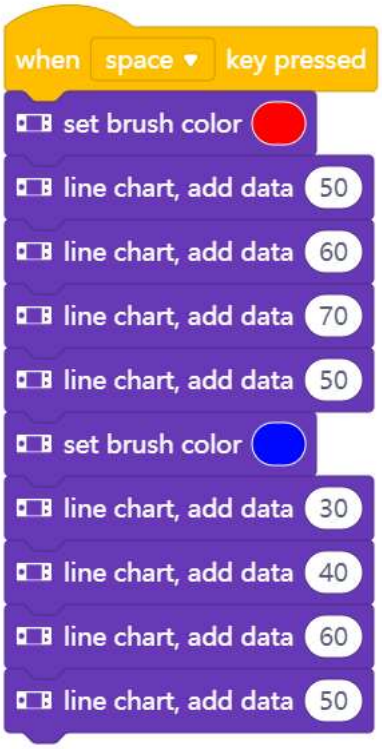
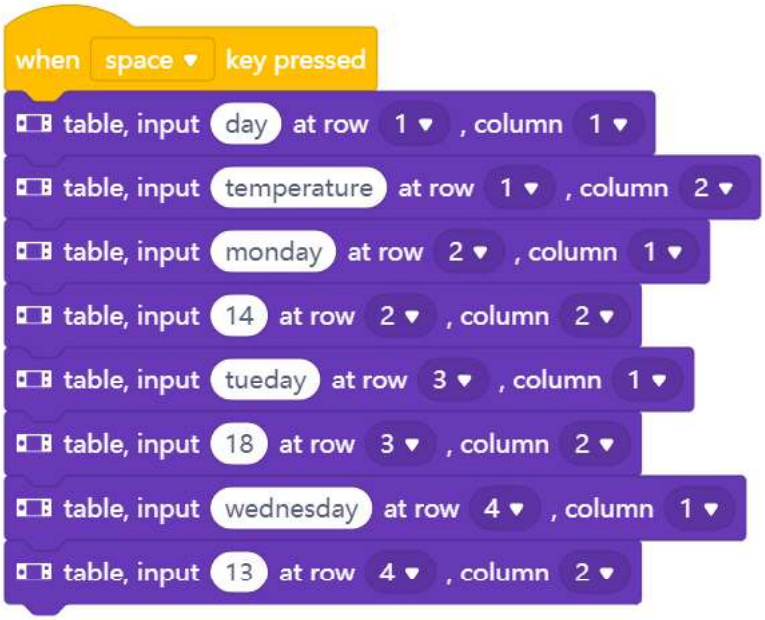
	Programming example
Line diagram	
Line diagram (color)	

Table	

4. Live mode and upload mode

You need to transfer the computer program you build in mBlock to the CyberPi.

The CyberPi puts the mBot2 to work according to the commands you wrote in mBlock. While writing a computer program, you can directly test what you are making. You do that by using live mode. You select live mode by moving the Mode switch to the right. You must not disconnect the CyberPi from the computer while testing in live mode!

The CyberPi also has an upload mode. In upload mode, the computer program is transferred to the CyberPi. The program is stored on the CyberPi until you replace it with another program. You can simply unplug the CyberPi.

In the diagram below you can see the differences between live mode and upload mode.

Mode	Differences explained
Live mode	• Programs run on your computer (=your PC or laptop). They will not be stored on the CyberPi. • The CyberPi must remain connected to your computer. • The mBlock code editor must always be open on your computer. • This mode is used for programming in the stage.
Upload mode	• Programs run on the CyberPi instead of on your computer and are also stored on the CyberPi. • The CyberPi does not need to be connected to your computer. • The mBlock code editor does not need to be open on your computer. • A program that you upload to the CyberPi is saved on the CyberPi until you replace it with another program.

3. Trying out (25 min)

Step 3: Trying it out

You have already learned a lot about the sensors of the mBot2 and how to show the data from the sensors on the display of the CyberPi. You are now going to collect data yourself using the sensors of the mBot2 and show it on the CyberPi's display. You will do this while driving the mBot2 around the classroom or school. What data you collect and how you show it on the CyberPi's display is up to you.

Use the knowledge you gained in 'Step 2' of this lesson. Of course, you can do plenty of experimenting yourself with the different programming possibilities in mBlock.

When thinking about this assignment, it is helpful to use the following step-by-step plan. Do you have an idea of what you want to make? If so, first discuss with your teacher whether this is feasible.

	Explanation
Step 1: What do you want to do?	• What route do you want the mBot2 to drive? • What data do you want the mBot2 to collect?
Step 2: What do you need?	• What do you need in addition to the mBot2?
Step 3: What code blocks do you need to make the mBot2 drive?	• How will you make the mBot2 drive? • What data will you have the mBot2 collect? • What code blocks will you use? • Make a brief description on how your program works (pseudocode/natural language, flowchart or UML) • If you need further explanation, you can discuss with your fellow students, the teacher, or do a research on the topic. There is help available for every coding block in mBlock as well.
Step 4: In what way do you want to show the data from the sensors on the display?	• How do you want to show the data on the display? • What code blocks will you use? • Make a brief description on how your program works (pseudocode/natural language, flowchart or UML) • If you need further explanation, you can discuss with your fellow students, the teacher, or do a research on the topic. There is help available for every coding block in mBlock as well.
Step 5: Testing and implementation	• Is the first version ready? Test it! During the testing round, write down areas of improvement. • Work on the improvement points until the mBot2 does exactly what you had in mind. • Successful? Film the end result and ask your teacher if you can post it on social media with the hashtag #mbot2.

4. Wrap-up (5 min)

Step 4: Wrap-up

What data did your mBot2 collect? Did it go well? Did you get to see all the data on the display of the CyberPi?

In this lesson, you learned about the different sensors on the mBot2 and where you might encounter them in everyday life. You know how to program these sensors and how to show the data on the display. You also learned about the difference between the upload and live modes of mBlock.

It is now time for a brief reflection. Think on your own and discuss with the group:

- What do you think worked out well?
- What could be better?
- Which parts of the lesson did you find easy and which did you find more difficult?
- What would you like more explanation about?
- Who could help you with that?