Hands-on-Remote

Teaching Module

Automation in Miniature

All materials and contact details can be found on the project websites as well as in the Erasmus+ project profile:
https://sites.google.com/campus.ul.pt/hands-on-remote-language/home
https://erasmus-plus.ec.europa.eu/projects/search/details/2020-1-DE02-KA226-VET-008295

Core development team

- Marion Pellowski and Lorenz Kampschulte, Deutsches Museum, Munich, Germany
- Pedro Reis, Mónica Baptista, Luís Alexandre da Fonseca Tinoca, Lisbon University, Institute of Education, Lisbon, Portugal
- Wojciech Karcz, Adam Zahler, Anna Strzeszewska-Potyrała, Karolina Klimaszewska, Copernicus Science Center, Warsaw, Poland
- Miriam Voss, Mike Kramler, Marion Pellowski, Technical University Munich, Munich, Germany

Disclaimer

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Imprint

Authors: Marion Pellowski, Deutsches Museum and Technical University of Munich, Mike Kramler, Miriam Voss, Technical University of Munich, Munich, Germany

The English version was translated and adapted by
Marion Pellowski, Technical University of Munich and Deutsches Museum,
Annette Beljung, Mike Kramler and Miriam Voß, Technical University of Munich,
both institutions Munich, Germany.

Deutsches Museum, Munich, Germany
Layout & Design: Michał Romański
Print: February 2023

Inhalt

Acknowledgement 3
1 Introduction 4
1.1 General aims 5
1.2 Didactic rationale 6
1.3 Community feeling and digital collaboration 7
1.4 Short summary 8
2 Curriculum fits 11
3 Overview of the module sequence for the teaching module "Automation in Miniature" 12
4 Content: Learning sequence for the teaching module "Automation in Miniature" 13
4.1 Unit 1.a Introduction to the topic 13
4.2 Unit 1.b Introduction to basic circuits and programming with the Arduino 14
4.3 Unit 2.a Online simulation 17
4.4 Unit 2.b Servomotor 18
4.5 Unit 3.a Turntable 20
4.6 Unit 3.b Light barrier 22
4.7 Unit 4. Extensions to the automation process 24
4.8 Unit 5. Collaboration between students 26
4.9 Unit 6. Digital collaboration 27
4.10 Unit 7. Automation in its social context 28
4.11 Unit 8. Other applications of the turntable 30
4.12 General notes on video conferencing and hands-on work in remote situations 32
5 Possibilities for modular uses – different routes 34
Route 1 34
Route 2 34
6 Appendix: Insights into the interactive multimedia page 35

Acknowledgement

We would like to express our sincere thanks to the teachers at the Municipal Vocational School for Information Technology and IT Systems Integration in Munich for their cooperation and advice and for their constant and always constructive support, which was extremely helpful for the development of this teaching module. We would like to thank the young people who tested the module with us for their feedback and input, which was incorporated into the optimization of the module. Sincere thanks also go to all the teachers who participated in our needs assessment for their very valuable input and comments on this project.

1 Introduction

Teaching module
Automation in miniature

Students protram a small production plant and set it in motion. The automatic filling system is madde of simple materials an can be taken home by the student. Students can wortk together as a team, even in remote situations.

Main teaching module

Unit 1a

Brief introduction to the topic

Aim: Establishing references to everyday life and professional environments.
Task: Showing videos of various applications of turntables in automation.

Unit 1b

Introduction to the Arduino

Aim: Introduction to programming with Arduino IDE, using a first circuit design.
Task: Programming of different-colored LEDs, learning basic commands and common programming errors.

Unit 2a

Online Simulation

Aim: Getting to know the online simulation environment "Wokwi", which enables the exchange of code even in distance learning.
Task: Finding mistakes in a Wokwi-debugging file and observing the effects of changes on simulated LEDs.

Unit 2b

Servomotor

Aim: Learning about the characteristics and functions of servomotors.
Task: Using servomotors to move functional parts of the production plant.

Unit 3a

Turntable

Aim: Construction of the central part of the production plant.
Task: Setting up the turntable, adjusting the drive motor and letting the turntable move by a quarter.

Unit 3b

Light barrier

Aim: Using sensors for automated production, discussion of measurements with sensors.
Task: Using the light-dependent resistor (LDR) as a sensor in a reflex light barrier to control the production process.

Teachers can change the sequence of the units and start with 3a/3b and follow-up with 2a/2b.

Additional / optional units

Unit 4

Extensions to the automation process

Aim: Transfer of previously acquired knowledge to new elements. Scope for students' own projects.
Task: Free choice for the installation of other functional parts like
- LED status indicator
- Entrance Slide
- Ejector

Unit 5/6

Collaboration / Digital Collaboration

Aim: Promoting collaboration among students, including teamwork and a sense of community even in distance learning.
Task: Coupling of turntables of different teams; using light transmission to induce physical effects in remote situations.

Unit 7

Automation in its social context

Aim: Exploring and discussing the impact of automation processes on society.
Task: Discussing advantages and disadvantages of automation using an online-whiteboard, and considering aspects like energy demand, waste, etc.

Unit 8

Other uses of the turntable

Aim: Encouraging further engagement with the turntable.
Task: Using the turntable as a sorting machine detecting differences in brightness or color; using the turntable as light-induced sound player measurnig brightness.

1.1 General aims

The module “Automation in Miniature” allows students to program a mini production plant as a real mechanical system, considering effects and solving problems that would not occur in a purely virtual environment. To create a functioning production process, the interplay between the programming and the mechanical functional parts of the mini production plant is central. As for distance learning, the general idea of this module is to replace a high quality and expensively purchased classroom product with a simpler version that teachers can give to their students to take home.

Using the example of the mini production plant, the students learn about the structure and operation of a complex technical system. By controlling servomotors as actuators and using a light sensor to control the filling process, they program and control the small production plant. They program text-based with a microcontroller, the Arduino, which they can also use to record measurement values. The students test their programs on the real example of the production plant and apply troubleshooting during the tests. In doing so, they must also take environmental influences into account (e.g. the influence of ambient light on the measurement values of the light sensor).

Teachers can adapt the module to different levels of programming skills. Essentially, students need only a few lines of code for the basic programming of the automation processes. Due to its concept, the module is also suitable for an introduction to object-oriented programming (definition of objects, structure of objects, use of reality-based classes and objects). Teachers can also use it to transfer existing knowledge of object-oriented programming to an application example. However, programming the entire automation process is very well possible even without an understanding of object-oriented programming.

The topic of automation and its societal consequences affect many vocational students directly as they will be working with such production systems in one way or another. In a final lesson, students are encouraged to discuss and reflect on societal issues related to automation in order to consider their own future careers in a broader context.

In view of the high social relevance of the topic, the students deal with various advantages and disadvantages of automation for our society, including ecological and work-related aspects.

1.2 Didactic rationale

The project employs easy-to-use online tools like an online simulation environment and an online whiteboard to facilitate distance learning. Teachers can introduce these tools as well as the hands-on experiments in both face-to-face and distance-learning settings. This creates a common ground for teaching, regardless of the teaching situation.

Video conferencing for an introduction to the topic and working with the mini production system is central in distance learning, especially at the beginning, and remains important for keeping in touch with students and supporting them in their work. Different forms are possible: videoconferencing with the whole class, videoconferencing with subgroups in breakout rooms and being available as support at fixed times for requests from students. The module also uses video conferencing to foster collaboration between students (s. 1.3).

After an introduction to the basics, students can continue working independently using a multimedia instruction page. They can decide for themselves how to proceed. The teacher has the role of being the contact person for any problems that may arise. The use of the multimedia instruction page allows students to work independently with the materials and facilitates work in both face-to-face and remote situations. The combination of online tools, various forms of videoconferencing, and the multimedia instruction page allows the use of the module for pure distance learning as well as hybrid instruction. Teachers act as learning facilitators in this process, whether remotely or on-site in the classroom.

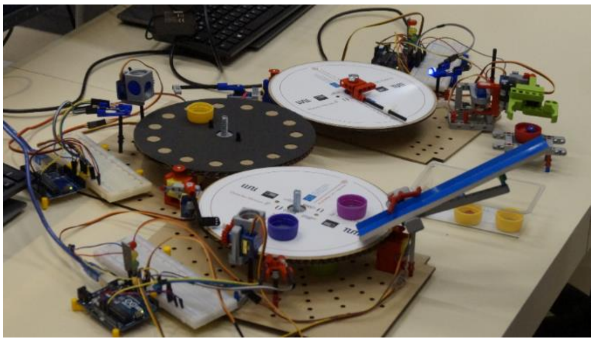
Automation is an important part of modern societies and determines a large part of our lives. It is therefore easy to refer to real automation processes and to students' everyday lives. With the automation process, references to various occupational fields are possible, too. Compared to industrially manufactured production systems for schools, the mini production plant consists of simple materials and the parts are relatively easy to replace. This is crucial for making the material kits available to students in distance learning. In addition, it seemed important to us to make the set so attractive that students might be inspired to continue working with it in their free time.

For use in a very heterogeneous European context, the degree of complexity can be adapted to the respective target group through flexible use and redesign of the mini production plant for other applications.

For more details on development and implementation, see the accompanying Guidebook. According to the categorization made in the Guidebook (chapter 2), this module combines the approaches of local and real experiments, which require sending or picking up physical materials, and remote and virtual experiments, which can give a better basis for conceptual understanding.

1.3 Community feeling and digital collaboration

Teamwork with hands-on experiments has great potential to foster a sense of community among students. Some tasks in this module are designed to promote collaborative learning, especially when students are asked to make connections between their different production plants and have them work together.



Students can connect the different teams' turntables, using the ejector to transfer cups from one turntable to another.

The coupling of the turntables is possible, even if none of the teams has built up the automation process completely. In this case, the students just use a different functional part on each turntable and establish a common transfer mechanism between their turntables.

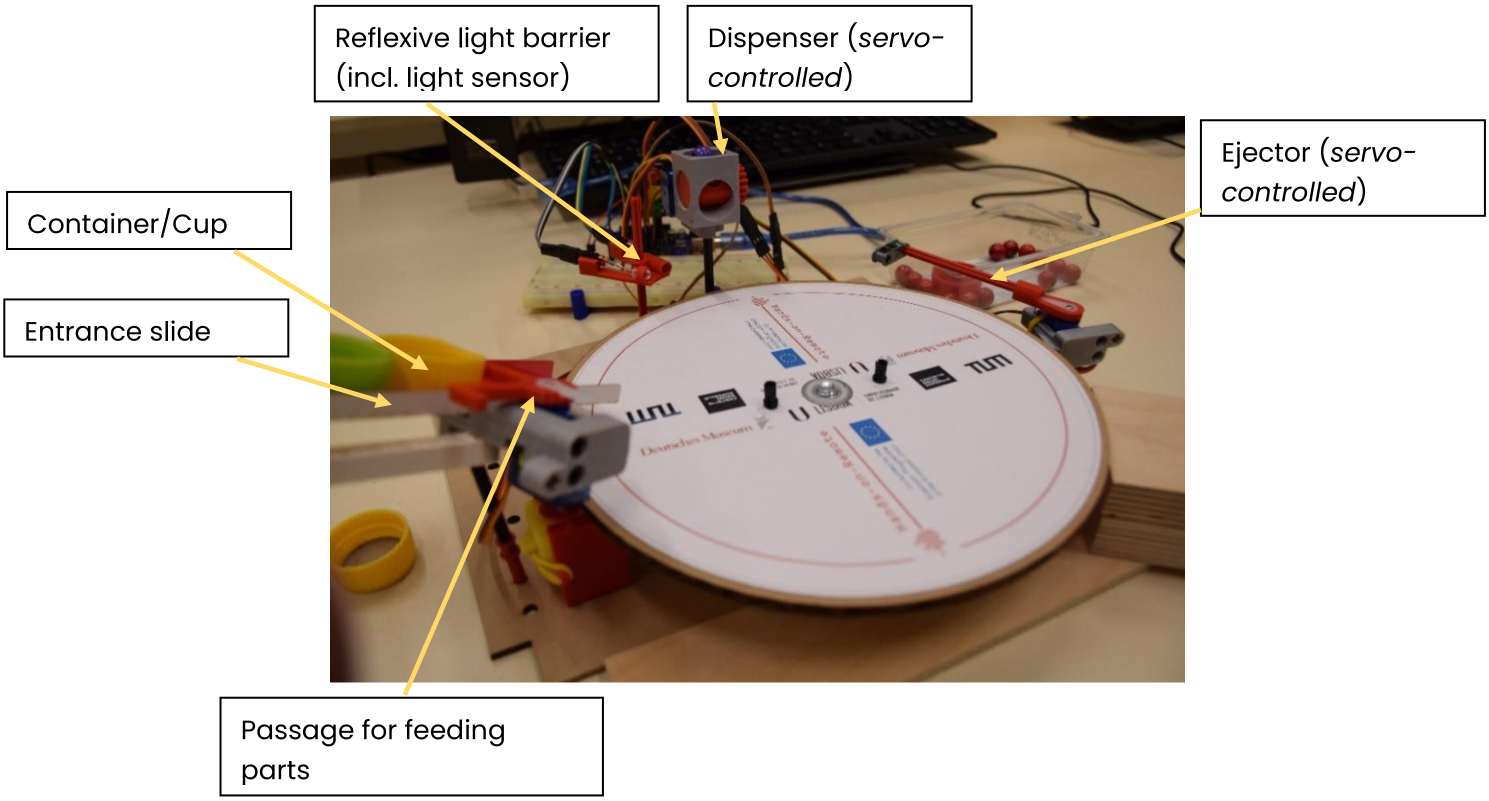
The online simulation environment allows students to share their code and work together on the programming tasks. In this way, close cooperation remains possible even in distance learning.

In order to take advantage of the motivational effects of collaboration and to work against isolation, this module involves a special type of collaboration: Students can interact with each other via light transmission in video conferences, moving parts of another team's turntable at a different location. On the first student's laptop, an LED shines on the webcam for the video conference. In front of the second student's laptop, the light sensor is mounted in a way that it can detect the signal. In response to the light signal, a servo is set in motion at the second student's location and can move a part of this student's production plant.

The concept is to use this approach to create a sense of inclusion despite physical distance - and simply to trigger joy. It is a very simple approach, but it has the great advantage of being easy to understand, low-cost and very quickly implementable. The approach builds on teachers' prior knowledge and that makes it very adaptable even for teachers with little experience with technical media in remote teaching.

1.4 Short summary

The teaching module "Automation in Miniature" enables students to program a small production plant themselves and set it in motion. A traffic light module as status indicator, a light barrier and a turntable serve as hands-on elements, which together with various moving functional parts become a mini production plant.



The students can control the turntable as a filling system - in which the light barrier detects a container on the turntable, the status display indicates a change of state by changing from red to green and a dispenser is used to fill something into the corresponding container. The production process is sensor-dependent in the way that a reflex light barrier detects a passing cup and then triggers the filling process.

Structure of units

Teachers can choose different approaches and progressions of the lessons. The teaching units build on each other, but allow for a division into a beginner and an advanced program. At the beginning, some motivating insights into the industrial application of turntables put the school class in a fictional factory. For their mini manufacturing plant, teams can choose what they "produce" with their own fictional scenario.

As an introduction to programming with the Arduino, the students first switch the LEDs of the status display on and off. For this purpose, they work with both a real and a virtual traffic light module. For the work with the production line, the teams all start together and, for example, set the turntable in motion and build and program the dispenser. After that, the students can decide which additional components they want to add to their manufacturing process.

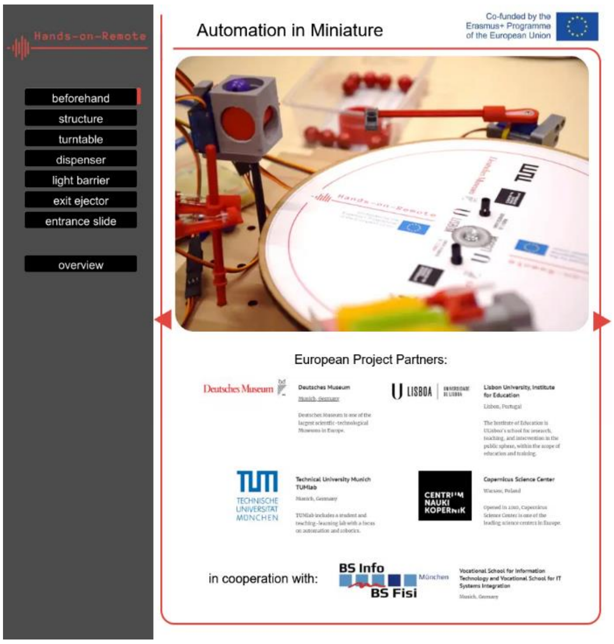
Programming

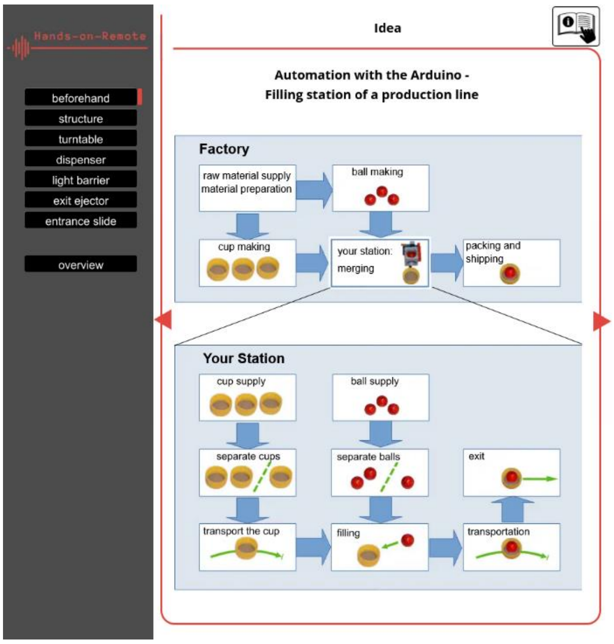
The difficulty level can also be set at the programming level: If a teacher wants to use the module for an entry level, students enter only a few programming commands in a largely predefined programming. If a teacher wants to use the module for advanced programming, students have additional tasks, such as defining classes.

Materials for the Teaching Module

The teaching Module “Automation in Miniature” includes the following materials, which can be accessed on the project’s download page https://sites.google.com/campus.ul.pt/hands-on-remote-en/downloads or on the project page of the Deutsches Museum: https://www.deutsches-museum.de/forschung/forschungsinstitut/projekte/detailseite/erasmus-hands-on-remote

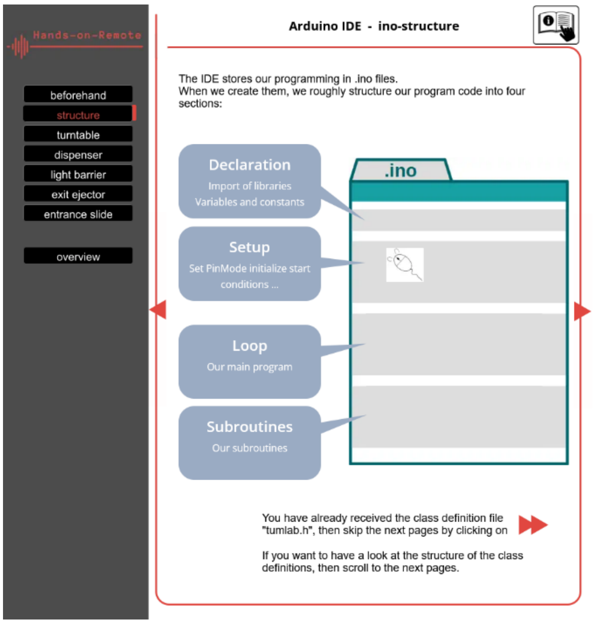
- Presentation file for teacher, *presentation file for teacher EN (pdf)* and a video about a circuit, *video_circuit (mp4)*
- Interactive multimedia tutorial site for students, step-by-step instructions for the independent work of the students with the mini production line, *interactive tutorial site for students EN (html)*

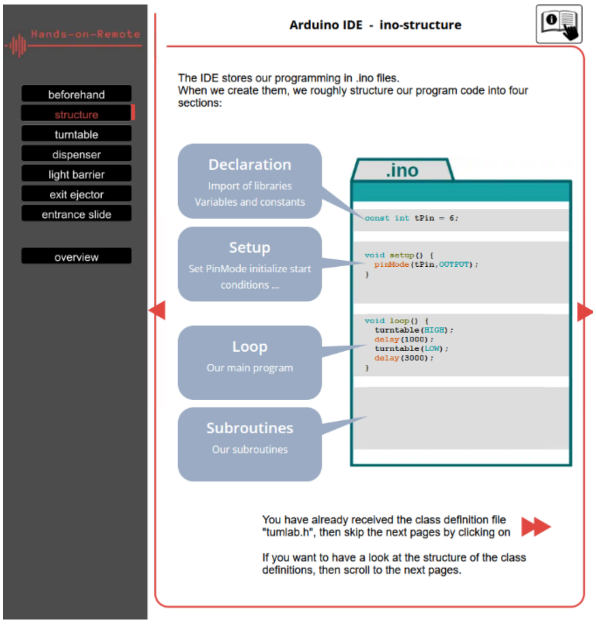




- Worksheets for the unit "Automation in a social context" and link to the online whiteboard, *worksheets automation in its social context EN (pdf)*
- 3D print and laser cut files for the material kit, *Material kit (folder with list, stl- and lasercut files)*
- A file listing common errors and difficulties, *Troubleshooting_Automation_in_Miniature (pdf)*

The description of the individual steps for setting up and programming the entire production plant is beyond the scope of this teaching module. This information can be found in the accompanying files mentioned above.





For the mini production plant, the instructions for building and programming are summarized on the interactive multimedia tutorial site. Screenshots in the appendix provide an initial insight into the structure and design of the instructions page (see chapter 6, Appendix: Insights into the interactive multimedia page). This will give a better idea of how to work with the Arduino and the mini production plant remotely.

2 Curriculum fits

The design of the module, in particular the mini production plant and the accompanying materials, is based on needs analyses with teachers from various vocational schools. The module has links to the curriculum of various training professions and to the lessons of the technical secondary school. Professions for which different parts of the module can play a role are, for example, computer scientist, electronic technician, automation technician, industrial electrician, software developer.

The learning unit on societal aspects of automation has links to applied ethics (ethics of technology). Keywords for the topics are e.g. systems and processes, complex technical systems, control technology and object oriented programming. Connections to the curriculum generally exist for grades 10-13.

3 Overview of the module sequence for the teaching module "Automation in Miniature”

Unit 1.a Introduction to the use of turn tables in automation and the goal of miniaturizing an automation process. *(15 min)*	Unit 1.b Introduction to working with the Arduino using a first circuit design and programming example with different colored LEDs. *(30-45 min)*
Unit 2.a Introduction to an online simulation environment for programming the Arduino to be able to work on programming in a distance-learning situation as a team. Design of a simulated circuit for a servomotor and programming of the servomotor. *(at least 15 min)*	Unit 2.b Introduction to servomotors, including circuitry and programming a real servomotor *(30 min)*
Unit 3.a Setting up the turntable, circuitry and programming of the drive motor. *(20 min)*	Unit 3.b Operation of the light barrier, including circuitry, programming and display of values on the plotter. *(25 min)*

Unit 4. The students work independently and can choose which components for the turntable they add to the automation process (s. table below). *(At least 45 min)*

Short version	Variation	Variation	Variation	Long version
Turntable Light barrier Dispenser	Turntable Light barrier Dispenser LED status indicator for the dispenser	Turntable Light barrier Dispenser Ejector	Turntable Light barrier Dispenser Ejector Entrance slide	Turntable Light barrier Dispenser Ejector Entrance slide LED status indicators for the actuators

Unit 5. Collaboration between students: Coupling of the turntables of different teams *(at least 45 min)*

Unit 6. Digital collaboration: Using light transmission to induce physical effects in remote situations *(flexible duration)*

Unit 7. Automation in its social context *(At least 45 min)*

Unit 8. Possible other uses of the turntable (s. table below) *(At least 45 min)*

Sorting machine	Light-induced sound player
Turntable Light barrier LED status indicator Brightness/Color Ejector for sorting	Turntable with paper overlays in a black and white pattern Light sensor as measuring head Loudspeaker (not included in the material kit)

4 Content: Learning sequence for the teaching module "Automation in Miniature”

4.1 Unit 1.a Introduction to the topic

Topic: Different real-life examples of automation processes

Objective: Brief introduction of the students to the subject, establishing the basic idea to miniaturize an automation process

Duration: approx. 15 min

Background/Introduction to the idea:

Many vocational students will come into contact with automation processes in their working lives - either because they might program production lines themselves in the future or because they will work with such systems - for example, as food technicians.

Overall, automation offers many real-world references.

The idea of a mini automation process goes back to a larger production line in the TUMlab. In this student lab, the corresponding "production process" is programmed and automated by students. Teachers have repeatedly expressed the wish whether there is not also the possibility of a production line in smaller format, which the students can program at school - or now also at home. That was the origin of the idea of using the Arduino to enable a small-scale automation process.

Form: Online via video conference or in the classroom

Materials needed for the teacher: PowerPoint presentation, including links to YouTube videos

Materials needed for the students: None

Content:

Teacher activities:

Teachers provide insights into real-life examples of automated industrial plants with excerpts from various Youtube videos. The videos are about turntables, which play an important role in automation processes. One of these videos shows someone topping a pizza using a turntable and an automated process. This idea can relate to the students' own everyday life. The introduction establishes the idea of using a turntable instead of a straight conveyor belt for the miniaturized automation process. In principle, you can use the turntable in various forms:

- either as a conveyor disc of an automation system with several work steps
- or as a single turntable, e.g. as a single processing station of a complete production line, such as a filling station.

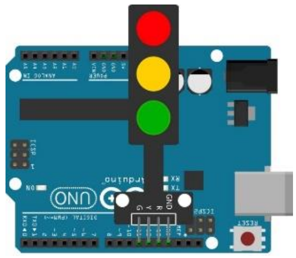
Student activities:

Students watch the videos

4.2 Unit 1.b Introduction to basic circuits and programming with the Arduino

Topics: Circuit design, basic programming commands (Arduino IDE), common programming errors

Objective: The students get to know the microcontroller actively for the first time. They learn basic working methods for programming with a microcontroller. For an easy start, students work with LEDs or the traffic light module, which can later serve as a status display for automation processes.



Background/Introduction to the idea: Microcontrollers are built into many everyday technological devices, even though they are often invisible. A microcontroller requires both circuitry and programming for its functions, giving teachers the opportunity to introduce both topics. For students, it is motivating that they can produce a direct output with their work. If students do not achieve the desired result, they or their teachers can quickly identify and correct errors.

Duration: approx.30 min, with use of the video on electric circuits and trying out the DC Virtual Lab, approx. 45 min

Materials needed for the teacher: PowerPoint presentation

Materials needed for the students: Arduino, LEDs or traffic light module, breadboard, Arduino IDE. If teachers want to use them: Video on circuits, portable files on a USB stick.

Important to note: The Arduino is connected to the computer via a USB hub. Each time a circuit is built, the Arduino is disconnected from the USB hub. This procedure is used to protect the computer in case of a possible short circuit.

Form: Online via video conference or in a computer lab at their school

Content:

Teacher activities:

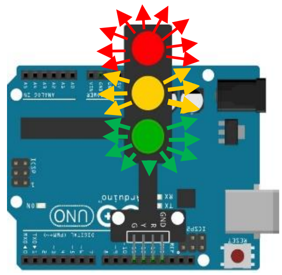
- Teachers familiarize the students with the Arduino and the basic steps of Arduino projects.
- Teachers introduce the text-based programming environment of the Arduino IDE.
- Depending on the learning level of the students:
- Teachers remind students of the basic circuitry to make a LED light up.
- Teachers guide students in building the circuit on the Arduino with a LED or the traffic light module.
- Teachers ask the students about their progress and ask to see the flashing LEDs.
- Teachers assist students with technical problems using a list of common errors and difficulties.

- In distance learning, teachers let the students share their screen in the video conference to help with programming problems.
- Teachers connect the programming of the traffic lights module to automation by teaching their students about the use of LEDs/traffic lights as status indicators in automation systems.

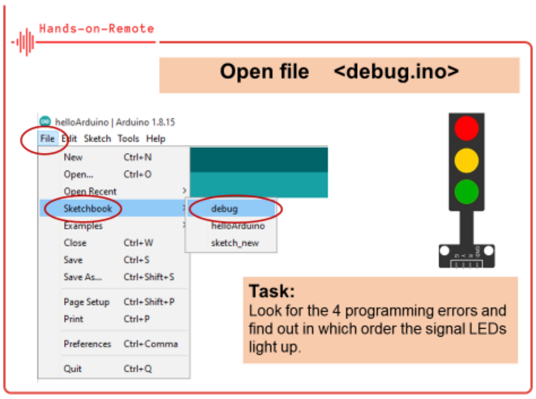
Student activities:

- Students watch a video on circuits and build their own circuit online with the Circuit Construction Kit: DC Virtual Lab (https://phet.colorado.edu/sims/html/circuit-construction-kit-dc-virtual-lab/latest/circuit-construction-kit-dc-virtual-lab_en.html).
- Students build their first real Arduino circuit with an LED or the traffic light module.
- Students connect the Arduino and upload the first code to let an LED blink.
- The first programming task for the students is to change the blinking time of one LED.
- Students overcome technical problems using a list of common errors and difficulties.
- If necessary, students discuss their programming with their teachers.

- The task for the students is now to let 3 LEDs flash at the same time. Teachers can switch to object-oriented programming at this point: 3 LEDs = several objects of the same kind. In object-oriented programming, you can therefore combine them in one class.



Students who already have in-depth knowledge of object-oriented programming can define their own classes. All others can use predefined class definitions. The use of predefined class definitions simplifies the programming for the entire further process.



- Students deepen their knowledge of the programming language by searching for typical semantics/syntax errors in a pre-made debugging file.

4.3 Unit 2.a Online simulation

Teachers can either continue in this order or deal with the turntable (3.1) and the light barrier (3.2) first.

Topics: Use of an online simulation environment for detection of common programming errors and code sharing, first programming of a servo motor

Objectives: Students get to know the online simulation environment "Wokwi" and learn how to use it. This can become particularly important when they want to share code within a team in a distance-learning situation. They transfer code between Wokwi and the Arduino IDE.

Background/Introduction to the idea: Teachers have expressed a desire to use the same methods in face-to-face and online classes. Teachers can use the Wokwi online simulation both on-site and in distance learning. Students can use it to consolidate their acquired knowledge of troubleshooting in the programming environment. Since direct effects are also visible in the online simulation, the motivating factor of seeing a direct output of one's own work can be maintained for students in the online environment as well. The transition to the simulation with the debugging file to the traffic light module with the same errors allows students to recognize real objects in the simulation. In addition, students can add a new element to their code and later transfer that knowledge to the real Arduino. Furthermore, the online simulation offers great practical advantages for the exchange of codes.

Duration: approx. 15 min

Materials needed for the teacher: PowerPoint presentation

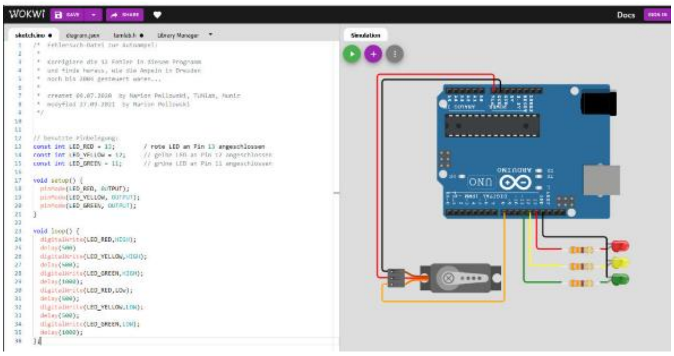
Materials needed for the students: Pre-made debugging Wokwi-file (https://wokwi.com/projects/328915985532715602), Internet connection

Form: Students work independently with the Wokwi-file, either online and with teacher support via video conferencing or in a computer lab at their school.

Content:

Teacher activities:

- Distributing the Wokwi-file and assigning the task



Student activities:
- Students find mistakes in a Wokwi-debugging file and can watch the effects of their changes on simulated LEDs.
- Students extend the simulation with the use of a servomotor.

4.4 Unit 2.b Servomotor

Topics: Getting to know a servo motor and its function, programming the angular positions of a servo motor.

Objectives: Students learn about the properties and functions of servomotors. In the miniaturized production plant, the student teams will use servomotors as actuators to move gears or barriers, for example. In general, the use of servomotors as actuators is quite common in miniature automation processes or in model making.

Background/Introduction to the idea: Three servomotors move essential components of the mini production line: one servomotor forms the passage of the entrance slide, one servomotor is the passage for the balls at the dispenser, and one servomotor moves the ejector. The students can thus transfer their programming knowledge of the servomotor to different applications. If teachers want to introduce it, the students can learn that different real objects of the same type are, in programming terms, different objects of the same class - an important insight in object-oriented programming.

Duration: approx. 30 min

Materials needed for the teacher: PowerPoint presentation

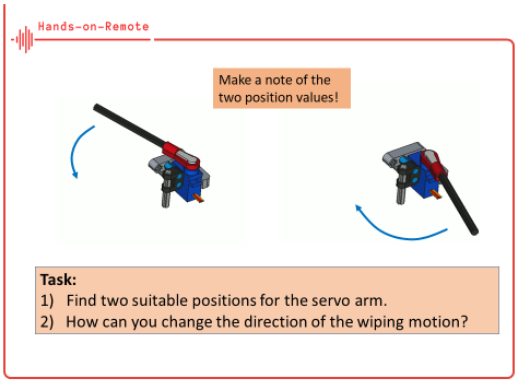
Materials needed for the students: Wokwi simulation (https://wokwi.com/projects/328915985532715602), Arduino IDE, Multimedia HTML-manual, Arduino, servomotor, breadboard, wires. If teachers want to use them: portable files on a USB stick.

Form: Online or in a computer lab at their school

Content:

Teacher activities:

- Introduction: What can a servomotor do? What do we want to use it for?



Student activities:

- Students can choose to work with the dispenser, barrier, or servo arm first. The servomotor only changes between two positions; it switches between rest position and active position.
- Students may copy new necessary code first into the simulation, then into the Arduino IDE or vice versa.
- Students build a circuit with a real servo motor and the Arduino and try to find suitable rest or active positions.

4.5 Unit 3.a Turntable

Topics: Apply practical skills to assemble mechanical parts, build another simple circuit, and insert new program code to create a new object.

Objectives: Students get hands-on practice as they set up the turntable and adjust the drive motor. Students may have to solve small mechanical problems during assembly, e.g. to find the right position for the drive motor. They build another circuit with the Arduino and add new code to move a new element.

Background/Introduction to the idea: The turntable is central to the miniaturized automation process, as it enables the transport of objects between the various stations of the automation process.

Form: Students work independently under the guidance of the multimedia HTML-manual, either online and with teacher support via video conferencing or in a computer lab at their school.

Duration: approx. 20 min

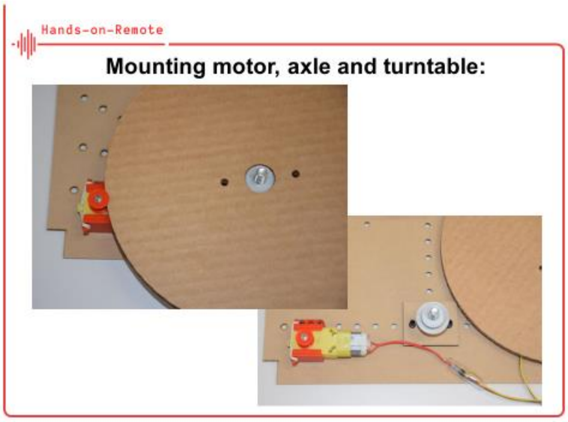
Materials needed for the teacher: PowerPoint presentation, including links to YouTube videos

Materials needed for the students: Multimedia HTML-manual, Arduino IDE, base plate, turntable, drive motor with friction wheel, motor holder, rubber ring, Bolt with 2 nuts as axle, nuts, black lego pins, Arduino, breadboard, wires

Content:

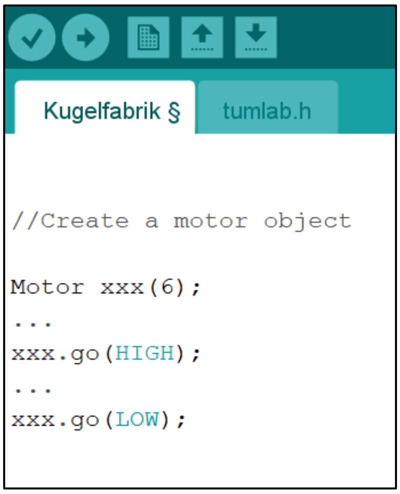
Teacher activities:

- Short introduction to the assembly of the turntable using the PowerPoint presentation



Student activities:

- Students mount the turntable and the drive motor onto the base plate.
- Students expand the circuit of the Arduino and connect the drive motor.



- Students add just a few lines of new code for the drive motor. They create a motor object and turn the motor on or off. The code needed to switch the motor on or off is very similar to the code needed to let a LED blink, so students are able to transfer previously acquired knowledge. Their task is to let the turntable move by a quarter.

4.6 Unit 3.b Light barrier

Topics: Circuitry of the reflex light barrier, Light dependent resistor (LDR) as brightness sensor, relationship between resistance values and light intensity, sensor-dependent control of processes.

Objective: Students identify the need for sensors for the automated control of a production plant. They learn how a reflex light barrier works and apply it as a sensor to control their "production process".

Duration: approx. 25 min

Materials needed for the teacher: PowerPoint presentation, including links to YouTube videos

Materials needed for the students: Multimedia HTML-manual, Arduino IDE, already assembled turntable, LDR + LED as reflex light barrier, Arduino, breadboard, wires (already assembled circuit on Arduino and breadboard)

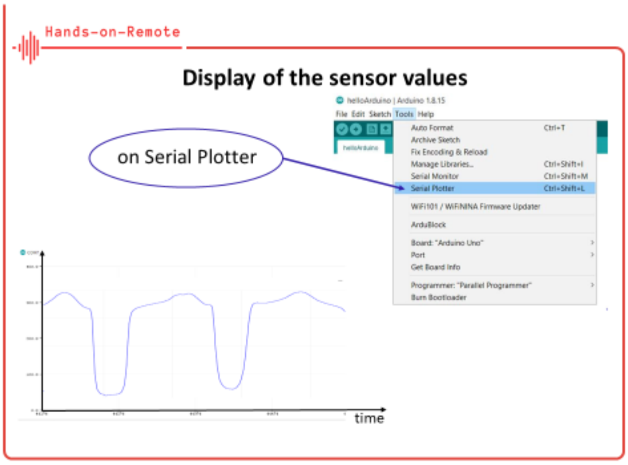
Background/Introduction to the idea: Light barriers are frequently used elements in everyday life, e.g. when closing suburban train doors. The students learn about a possible application for automation processes with the mini manufacturing plant. Students will gain insight into how the light-dependent resistor works and how measured values can be influenced by even small deviations in brightness. Teachers can use these measurements to remind students of the difficulties of obtaining accurate and meaningful measurements and discuss with student teams what this means for sensor-dependent processes using their mini manufacturing facility as an example.

Form: Students work independently under the guidance of the multimedia HTML-manual, either online and with teacher support via video conferencing or in a computer lab at their school. In an online setting, students might work together in breakout rooms.

Content:

Teacher activities:

- Teachers show a typical use of reflex light barriers in automation processes.
- Teachers introduce the light dependent resistor (LDR) as a brightness sensor.



- Teachers assist students in using the light barrier and generating the measured value plots.
- Teachers are available for student queries, e.g. via video conferencing.

Student activities:

- Students learn about the properties of a reflex light barrier using the multimedia HTML-instructions.
- Students mount the reflex light barrier on the base plate.
- Students expand the Arduino circuit and connect the LDR/reflex light barrier to the Arduino.
- Students plot the sensor values they obtain from the LDR and interpret the graphics. They can also track changes under different lighting conditions.
- Students use the reflex light barrier to detect cups in front of it and stop or start the turntable accordingly.
- Students add lines of new code for the light barrier, especially the waiting command 'waitForCan' - only when the sensor detects a cup, the program continues

4.7 Unit 4. Extensions to the automation process

Topics: Transfer of previously acquired knowledge to new elements

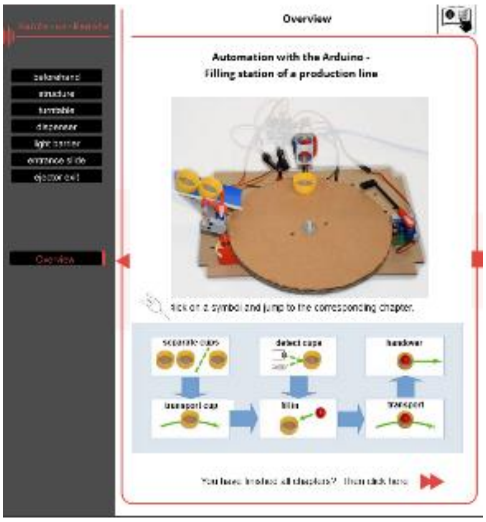
Objective: Students work independently and can choose which components for the turntable they add to their automation process. By doing this, they can transfer previously acquired knowledge about programming, circuits or automation processes and apply it to their new tasks. It also gives students an often-desired freedom to conduct their own experiments.

Duration: flexible, at least 45 min

Materials needed for the teacher: If needed, list of common errors and difficulties

Materials needed for the students: Multimedia HTML-manual, Arduino IDE, already assembled turntable, LDR + LED as reflex light barrier, Arduino, breadboard, wires, as material for the students to choose: dispenser, ejector, entrance slide, LEDs or traffic light module

Background/Introduction to the idea: Students can obtain an automation process with multiple functional elements.



Form: Students work independently under the guidance of the multimedia HTML-manual, either online and with teacher support via video conferencing or in a computer lab at their school. In an online setting, students might work together in breakout rooms.

Content:

Teacher activities:

Teachers are available for student queries, e.g. via video conferencing.

Student activities:

Students work with the multimedia HTML-manual to add more components to their automation process. From the overview page of the multimedia HTML-manual, students can click to go to the different components that interest them. Students can choose themselves which components they want to add.

Short version	Variation	Variation	Variation	Long version
Turntable Light barrier Dispenser	Turntable Light barrier Dispenser LED status indicator for the dispenser	Turntable Light barrier Dispenser Ejector	Turntable Light barrier Dispenser Ejector Entrance slide	Turntable Light barrier Dispenser Ejector Entrance slide LED status indicators for the actuators

This unit can be carried out flexibly in different variants (s. table above), depending on the time and content requirements of the teacher. With the different variants, teachers can also take into account different programming skills and previous experience of the students.

4.8 Unit 5. Collaboration between students

Topics: Extension of programming knowledge, collaboration/teamwork

Objective: To promote collaboration between students

Duration**:** at least 45 min

Materials needed for the teacher: None

Materials needed for the students: Mini production plants already set up in the teams

Necessary knowledge: Student teams need to have programmed at least one functional part of the automation process; ideally students in different teams programmed different parts or the whole automation process.

Form: Online or in a computer lab at their school

Content:

Teacher activities:

The teacher initiates and supports the collaboration of the students and their teamwork.

Student activities:

The task of the students is to couple their turntables across teams. The students make decisions about how they want to proceed in coupling their turntables and then jointly program the necessary steps.

4.9 Unit 6. Digital collaboration

Topics: Measurements with sensors, sensor-dependent triggering of different actors of the automation process, even at remote places

Objective: To foster a sense of community between students

Duration: flexible

Materials needed for the teacher: None

Necessary knowledge: Students need to have programmed sensor-dependent processes with actors beforehand (Unit 3.2).

Form: Online or in a computer lab at their school

Content:

Teacher activities:

The teachers introduce the idea of a communication process via light transmission and support the students in adjusting their sensors.

Student activities:

Students program one of the actors of the automation process and set it in motion, if the light sensor detects an increase in brightness. The first student positions the reflex light barrier, so that the LED shines on the laptop camera when it is switched on. The second student positions the reflex light barrier, so that the light sensor can detect the light transmitted by the first student in a video conference setting. By switching on the light, the first student sets the actor of the second student in motion. The second student can then send a light signal towards the third student and so on.

4.10 Unit 7. Automation in its social context

Teachers can either continue in this order or deal with other applications of the turntable (8.) first.

Topics: Consumer behavior, dependence on manufactured goods, side effects of consumption (energy demand, waste, transportation, etc.), advantages and disadvantages of automation processes

Objective: Students explore various potential impacts of automation processes on society. They reflect on their own consumer behavior.

Duration: Flexible, at least 45 min

Materials needed for the teacher: Access to the online whiteboard that the teacher can distribute to students; question cards to hand out to the students.

Materials needed for the students: Question cards, online whiteboard, and internet connection. If teachers want to use them: already assembled turntables

Form: Online or in a computer lab at their school

Content:

Teacher activities

The teacher hands out the cards to the students and/or leads a group discussion.

Introduction and transition to the new topic with a congratulation for the achievement and asking, what the students would like to indulge in after finishing their tasks. If teachers can directly implement an answer, they are welcome to incorporate that into their lessons.

The teacher discusses the answers with the students and distinguishes whether consumption of manufactured products is part of the proposed treats or not. If all answers are based on consumption, the teacher hands out the second card, which asks whether a treat without consumption would also be possible. He discusses with the students the side effects of high consumption (energy costs, waste, transportation, etc.).

Hands-on-Remote

Automation in miniature

Congratulations on your first automated production!

Now you have really achieved something, you can treat yourself to something nice......

What would you like to treat yourself to now?

Feel free to discuss in the team what you can think of and if you like, write it down here....

Hands-on-Remote

Automation in miniature

In your ideas of what you want to indulge in, is there anything that has nothing to do with consumption or consumer products?

Can you imagine treating yourself without needing or consuming anything? What would that be?

Depending on the course of the discussion and the chosen focus, the teacher asks further questions. The teacher familiarizes the students with the online whiteboard and explains their assignment to the students. The teacher can let the students choose a topic they want to explore in more depth. If teachers want to play with it they can use a paper overlay with suggested topics for the turntable and let the students choose their topic this way.

Student activities:

- Students discuss what they would like to indulge in and reflect on their consumption behavior together with the teacher.
- Students can look at the turntable they just built and recognize that it is also a "manufactured" thing: Which parts here come from an automation process, which seem to be self-made?
- Students can begin to discuss what advantages and disadvantages automation processes can have. They consider what applies the other way around when things are self-manufactured. Students collect their spontaneous ideas and thoughts they associate with automation and its social consequences on an online whiteboard. They are welcome to add new materials and new links between topics.
- Students can now choose a topic they would like to explore in more depth and discuss it in teams.

4.11 Unit 8. Other applications of the turntable

Topics: Use brightness values to distinguish objects or play sounds

Objective: Both students and teachers brought up the idea that it would be nice to find another application for the same materials to continue working with the material kit and try something new. Some of them were also willing to tinker further with the material kit in their spare time. Students can continue to work very independently here.

At the same time, these other applications may represent alternatives if teachers and students do not want to get too involved in programming.

Duration: Flexible, at least 45 min

Materials needed for the teacher: If needed: Printer to print grayscales made by the students.

Materials needed for the students: Arduino, breadboard, wires, Arduino IDE, already assembled turntable, light barrier

Additionally for the Sorting machine: Ejector and LED status indicator

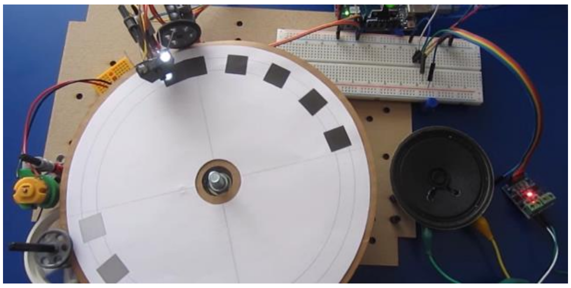
Additionally for the light-induced sound player: Paper on cardboard in disc size to put it firmly on the turntable, a printer to print the black and white pattern or the grayscale on the paper disk, a buzzer, a loud speaker

Sorting machine	Light-induced sound player
Turntable Light barrier LED status indicator Brightness/Color Ejector for sorting	Turntable with paper overlays in a black and white pattern Light sensor as measuring head Loudspeaker (not included in the material kit)

Form: Independent work of the students

Content:

- *Sorting machine*: Use the ejector to sort different types of objects that differ, for example, in brightness or color.



- *Light-induced sound player*: If you place a paper with a black and white pattern on the turntable, you can use the light sensor to convert this brightness information into tones and simple rhythms for a simple sound output device. You can also use a printed grayscale for the same effect.

Student activities:

- Students program the mini production line as a sorting machine and distinguish cups by color or brightness.
- The students design gray-scale patterns for the turntable. The Arduino recognizes the brightness values and plays the corresponding sounds. The students can also assign sounds to brightness values themselves in the programming.

4.12 General notes on video conferencing and hands-on work in remote situations

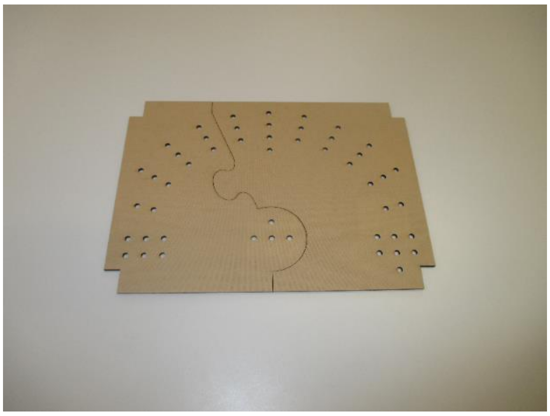
Data protection

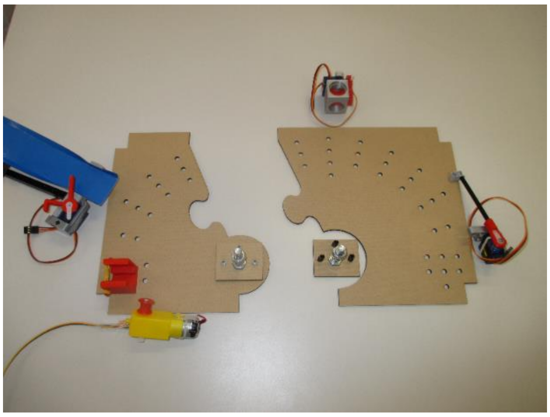
School should clarify questions about the necessary data processing during video conferencing - e.g. the use of screen sharing or the use of the camera - in accordance with the GDPR. In close consultation with students, teachers can conduct lessons without the camera on; however, if students turn on the camera, it is much easier to see how students are progressing, and it can significantly improve the group feeling. One option teachers can offer is to have students point the camera at their experiments without being in the picture themselves - but this option will not be available to all students.

Managing the materials

The first part of the project sequence, namely the sections 1.a Introduction to the topic to 2.b Servomotor, can easily be done in an online learning environment. For this, teachers need a suitable video conferencing system that allows them to show an online presentation and, if necessary, set up breakout rooms where students can work independently in teams. The proportion of practical parts that the students use for their work at home consists here of an Arduino, a breadboard, cables, a traffic light module or LEDs and a servo, i.e. a comparatively manageable set of materials. However, in a full remote setting this requires that all students in a class receive such a set. If this is not possible, the teacher can instead deepen the work with the online simulation environment Wokwi. However, this will be at the expense of the competencies to learn how to handle a microcontroller, skills that the students will still need for the later units.

When it comes to the turntable, the implementation becomes more complicated. Ideally, each student would receive a complete material kit to work with at home. If this is not possible (whether because of the cost, the effort required for production, the difficulty of borrowing or shipping the materials), there is need for an alternative.





An alternative may be to divide the base plate of the turntable and give the students different parts so that they can work as a team to create the automation process together.

5 Possibilities for modular uses – different routes

If the prerequisites are different, it may be helpful to use the teaching module differently. Two alternative routes are suggested below, there are probably more possibilities.

Route 1

Route 1 continues to rely on the automation process. However, the automation process is reduced to the transport of cups with the turntable, the use of the servomotor of the dispenser and a sensor-dependent control of the production process.

1.1 Introduction to the use of turntables in automation and the goal of miniaturizing an automation process.
1.2 Introduction to working with the Arduino using a first circuit design and programming example with different colored LEDs.
2.1 Introduction to servomotors, including circuitry and programming a real servomotor. Using the servomotor of the dispenser.
2.2 Setting up the turntable, circuitry and programming of the drive motor.
2.3 Operation of the light barrier, including circuitry, programming and display of values on the plotter. Using the light sensor to control the "production process", namely stopping the turntable if there is a cup in front of the dispenser.
3. Automation in its societal context

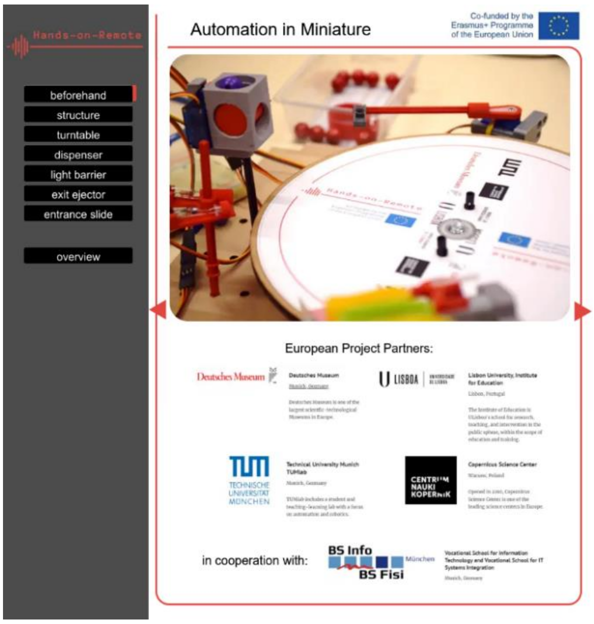
Route 2

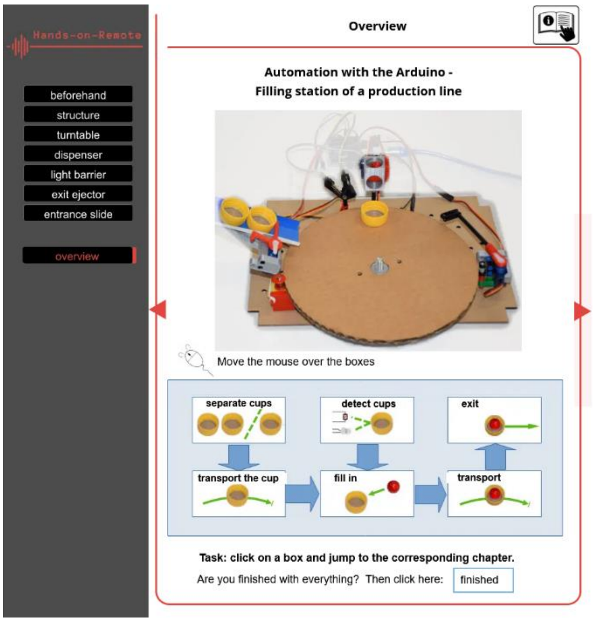
Route 2 allows you to use the turntable in a completely different way, namely only as light-induced sound player. Therefore, only the basic steps are necessary to work with the Arduino, to set the turntable in motion and to work with the light barrier.

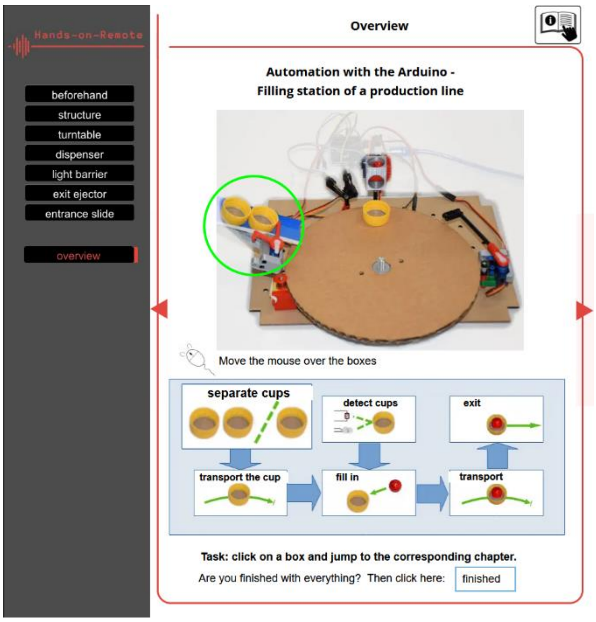
1.1 Introduction to a self-chosen topic
1.2 Introduction to working with the Arduino using a first circuit design and programming example with different colored LEDs.
2.1 Setting up the turntable, circuitry and programming of the drive motor.
2.2 Operation of the light barrier, including circuitry, programming and display of values on the plotter.
3. Use of the turntable as light-induced sound player

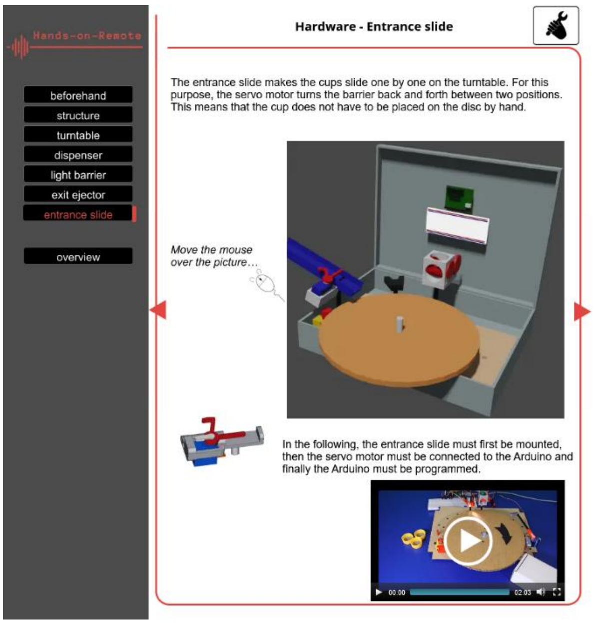
6 Appendix: Insights into the interactive multimedia page

For a first and quick impression, the screenshots show exemplary step-by-step instructions with pictures of the animations, videos and mouse over effects for setup, wiring and programming of the entrance slide and light reflex barrier. To get a better impression, it may be advisable to zoom in to a larger view.









Hardware – mount the Entrance slide

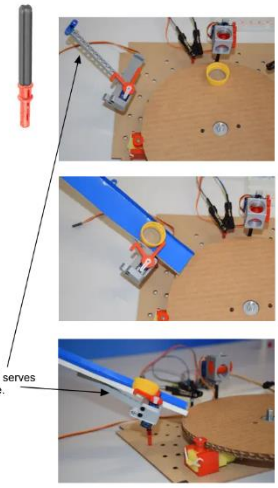
Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview

Mount the entrance slide with the gate:

1. Fasten the entrance slide with the stand in the base plate.

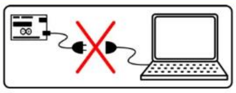
2. Push the slide into the guide.

3. The Lego arm (beam) serves as a support for the slide.



Wiring – Entrance Slide

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview



Please disconnect the Arduino from the computer before changing the wiring!

Work on the Arduino only in a de-energized state!

Now the servo of the gate of the entrance slide must be connected to the Arduino

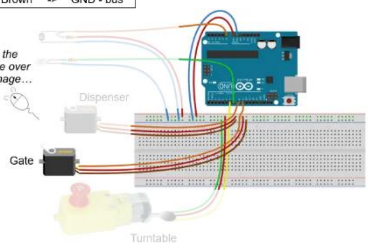
Use the supply bus (red and blue) on the breadboard again:

	at the arduino:		at the breadboard:
red wire:	V in - Pin	->	red power bus
blue wire:	GND - Pin	->	blue ground bus

Connect the servo of the entrance slide (gate) to:

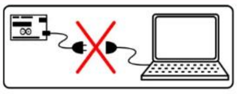
Orange	->	Pin 9
Red	->	V in - bus
Brown	->	GND - bus

Move the mouse over the image…



Wiring – Entrance Slide

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview



Please disconnect the Arduino from the computer before changing the wiring!

Work on the Arduino only in a de-energized state!

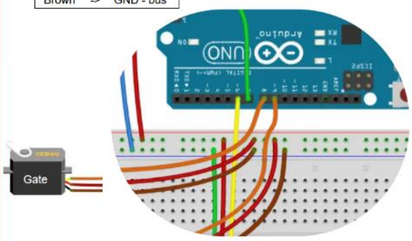
Now the servo of the gate of the entrance slide must be connected to the Arduino

Use the supply bus (red and blue) on the breadboard again:

	at the arduino:		at the breadboard:
red wire:	V in - Pin	->	red power bus
blue wire:	GND - Pin	->	blue ground bus

Connect the servo of the entrance slide (gate) to:

Orange	->	Pin 9
Red	->	V in - bus
Brown	->	GND - bus



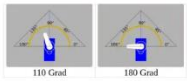
Hardware - Gate-Servo

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview

The barrier of the entrance slide (=gate) is moved by a servo.

Servos are small geared motors that can usually …

Move the mouse over the images…



For the barrier we need two positions:
-> barrier closed (= rest position, holds back all cups
-> barrier open (= pass position one cup slides onto the disc)

The angled pointer…

Click into the picture to start the animation…

The degrees of both positions must be determined by trial and error.

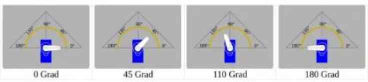
By the way: The angle value of the rest position is bigger than the angle value of the pass position.

Hardware - Gate-Servo

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview

The barrier of the entrance slide (=gate) is moved by a servo.

Servos are small geared motors that can usually move only half a turn, between 0° and 180°. You send the servo the desired number of degrees and the servo moves to the position



For the barrier we need two positions:
-> barrier closed (= rest position, holds back all cups
-> barrier open (= pass position one cup slides onto the disc)

The angled pointer…

Click into the picture to start the animation…

The degrees of both positions must be determined by trial and error.

By the way: The angle value of the rest position is bigger than the angle value of the pass position.

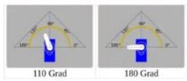
Hardware - Gate-Servo

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview

The barrier of the entrance slide (=gate) is moved by a servo.

Servos are small geared motors that can usually …

Move the mouse over the images…



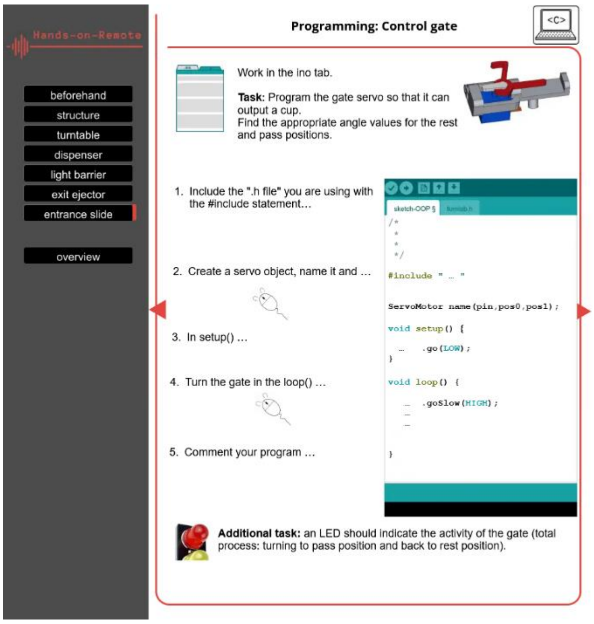
For the barrier we need two positions:
-> barrier closed (= rest position, holds back all cups
-> barrier open (= pass position one cup slides onto the disc)

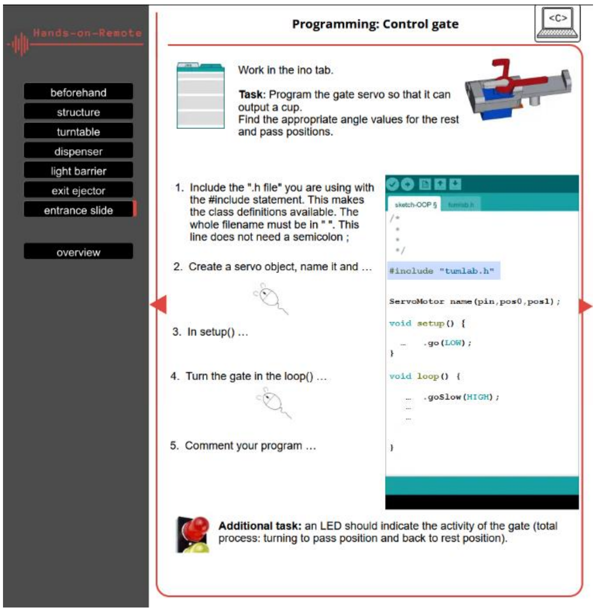
The angled pointer of the gate prevents further cups from slipping through when opening the barrier.

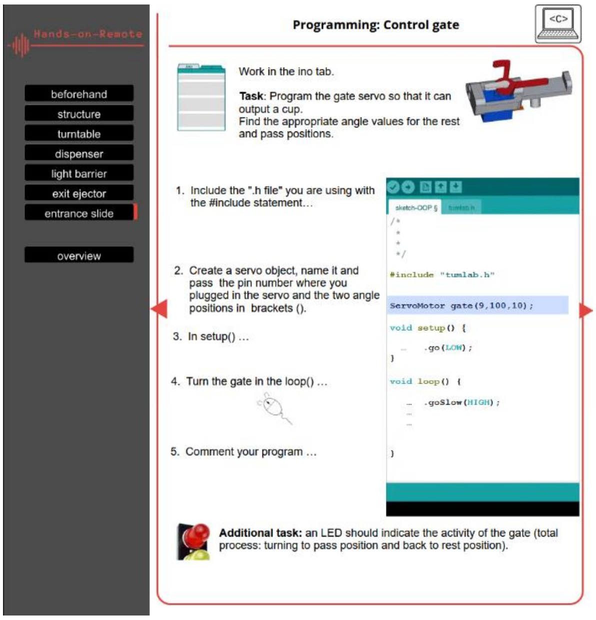
Click into the picture to start the animation…

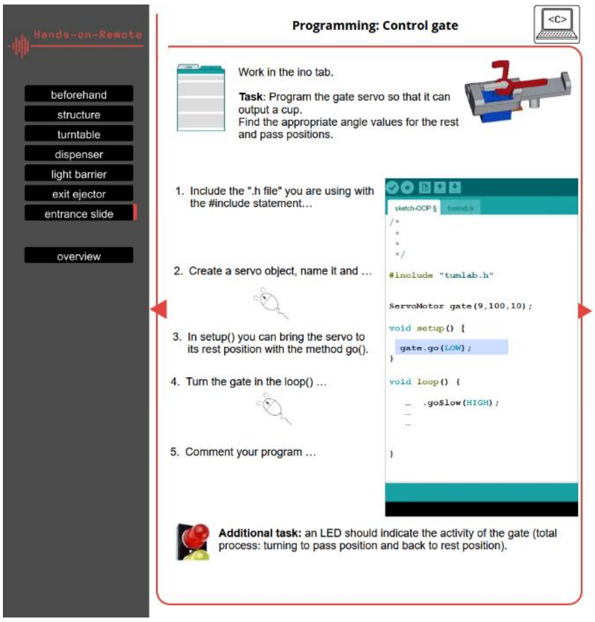
The degrees of both positions must be determined by trial and error.

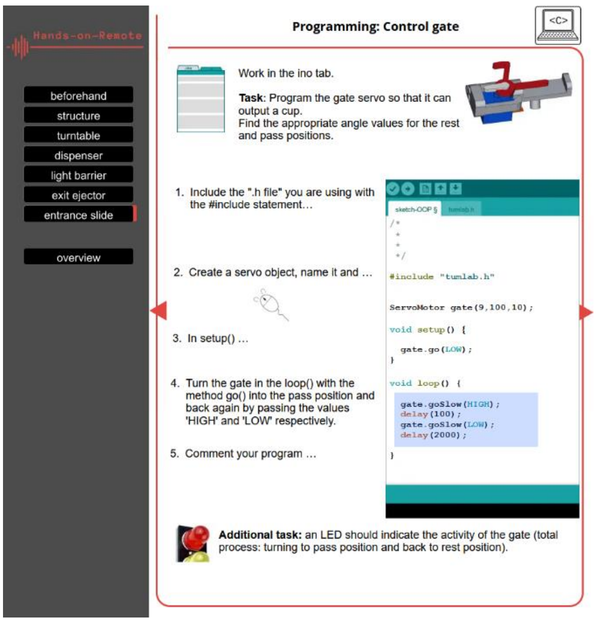
By the way: The angle value of the rest position is bigger than the angle value of the pass position.

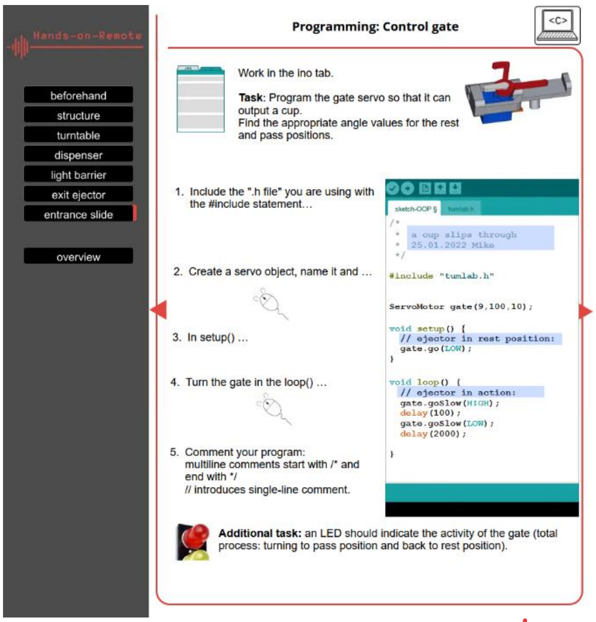

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: Control gate
Work in the ino tab.
Task: Program the gate servo so that it can output a cup.
Find the appropriate angle values for the rest and pass positions.
1. Include the ".h file" you are using with the #include statement…
2. Create a servo object, name it and …
3. In setup() …
4. Turn the gate in the loop() …
5. Comment your program …
Additional task: an LED should indicate the activity of the gate (total process: turning to pass position and back to rest position).

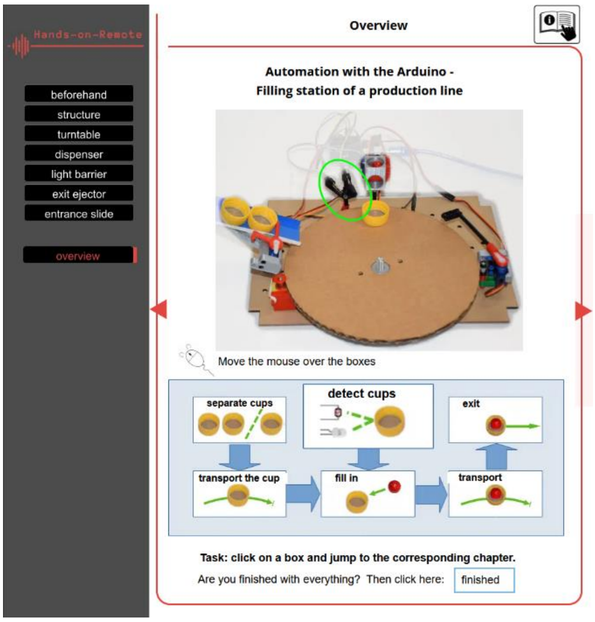

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: Control gate
Work in the ino tab.
Task: Program the gate servo so that it can output a cup.
Find the appropriate angle values for the rest and pass positions.
1. Include the ".h file" you are using with the #include statement. This makes the class definitions available. The whole filename must be in " ". This line does not need a semicolon ;
2. Create a servo object, name it and …
3. In setup() …
4. Turn the gate in the loop() …
5. Comment your program …
Additional task: an LED should indicate the activity of the gate (total process: turning to pass position and back to rest position).


Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: Control gate
Work in the ino tab.
Task: Program the gate servo so that it can output a cup.
Find the appropriate angle values for the rest and pass positions.
1. Include the ".h file" you are using with the #include statement…
2. Create a servo object, name it and pass the pin number where you plugged in the servo and the two angle positions in brackets ().
3. In setup() …
4. Turn the gate in the loop() …
5. Comment your program …
Additional task: an LED should indicate the activity of the gate (total process: turning to pass position and back to rest position).

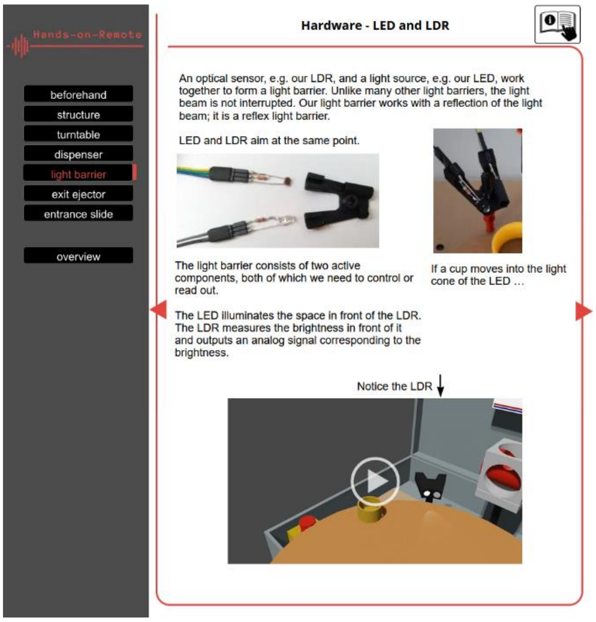

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: Control gate
Work in the ino tab.
Task: Program the gate servo so that it can output a cup.
Find the appropriate angle values for the rest and pass positions.
1. Include the ".h file" you are using with the #include statement…
2. Create a servo object, name it and …
3. In setup() you can bring the servo to its rest position with the method go().
4. Turn the gate in the loop() …
5. Comment your program …
Additional task: an LED should indicate the activity of the gate (total process: turning to pass position and back to rest position).


Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: Control gate
Work in the ino tab.
Task: Program the gate servo so that it can output a cup.
Find the appropriate angle values for the rest and pass positions.
1. Include the ".h file" you are using with the #include statement…
2. Create a servo object, name it and …
3. In setup() …
4. Turn the gate in the loop() with the method go() into the pass position and back again by passing the values 'HIGH' and 'LOW' respectively.
5. Comment your program …
Additional task: an LED should indicate the activity of the gate (total process: turning to pass position and back to rest position).

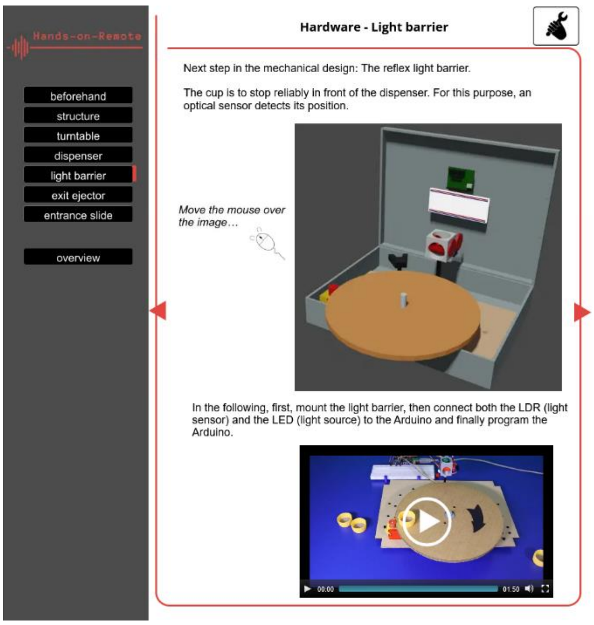

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: Control gate
Work in the ino tab.
Task: Program the gate servo so that it can output a cup.
Find the appropriate angle values for the rest and pass positions.
1. Include the ".h file" you are using with the #include statement…
2. Create a servo object, name it and …
3. In setup() …
4. Turn the gate in the loop() …
5. Comment your program:
multiline comments start with /* and end with */
// introduces single-line comment.
Additional task: an LED should indicate the activity of the gate (total process: turning to pass position and back to rest position).

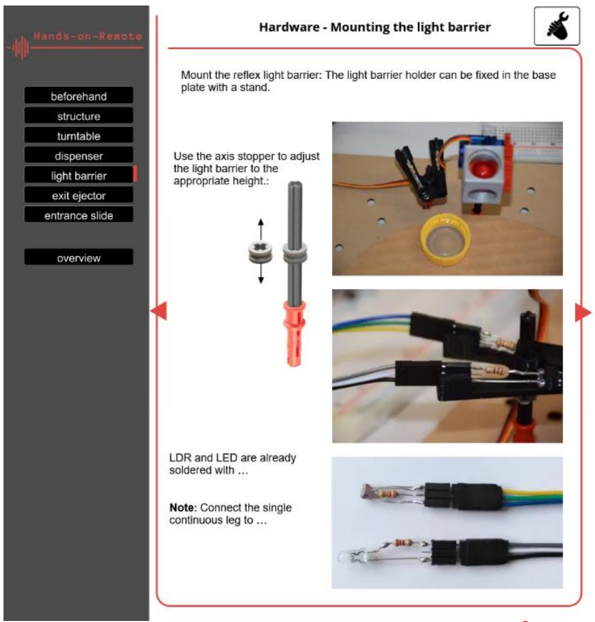

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Overview
Automation with the Arduino -
Filling station of a production line
Move the mouse over the boxes
separate cups
detect cups
exit
transport the cup
fill in
transport
Task: click on a box and jump to the corresponding chapter.
Are you finished with everything? Then click here:
finished

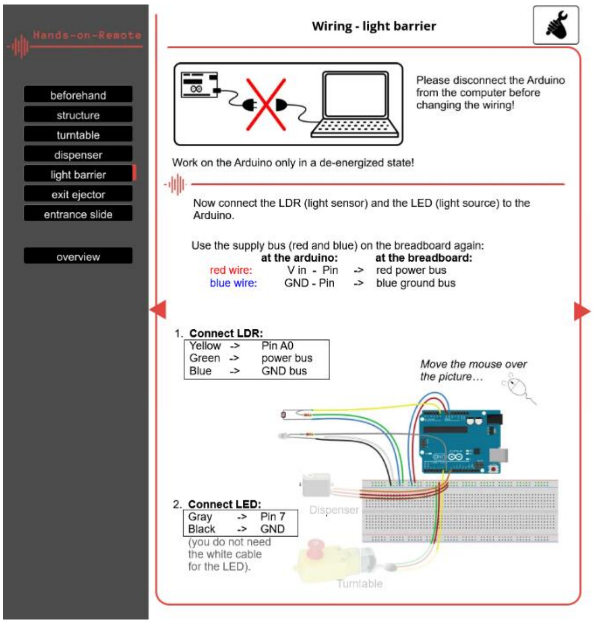

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Hardware - LED and LDR
An optical sensor, e.g. our LDR, and a light source, e.g. our LED, work together to form a light barrier. Unlike many other light barriers, the light beam is not interrupted. Our light barrier works with a reflection of the light beam; it is a reflex light barrier.
LED and LDR aim at the same point.
The light barrier consists of two active components …
If a cup moves into the light cone of the LED …
The LED illuminates …
The LDR measures …
Notice the LDR

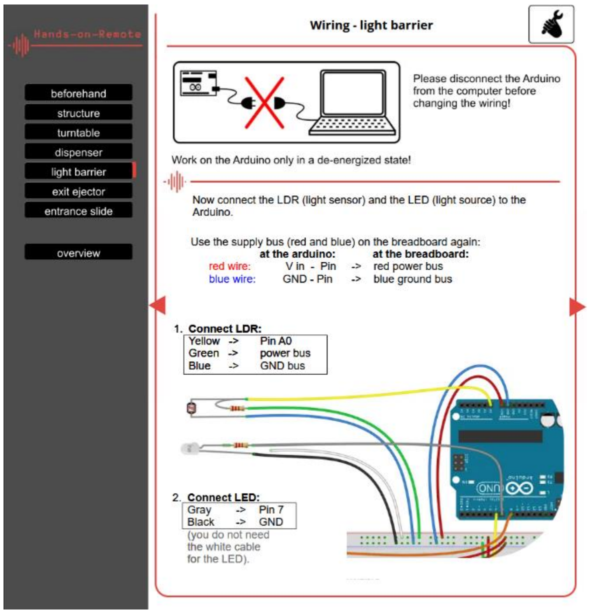

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Hardware - LED and LDR
An optical sensor, e.g. our LDR, and a light source, e.g. our LED, work together to form a light barrier. Unlike many other light barriers, the light beam is not interrupted. Our light barrier works with a reflection of the light beam; it is a reflex light barrier.
LED and LDR aim at the same point.
The light barrier consists of two active components, both of which we need to control or read out.
If a cup moves into the light cone of the LED …
The LED illuminates the space in front of the LDR.
The LDR measures the brightness in front of it and outputs an analog signal corresponding to the brightness.
Notice the LDR

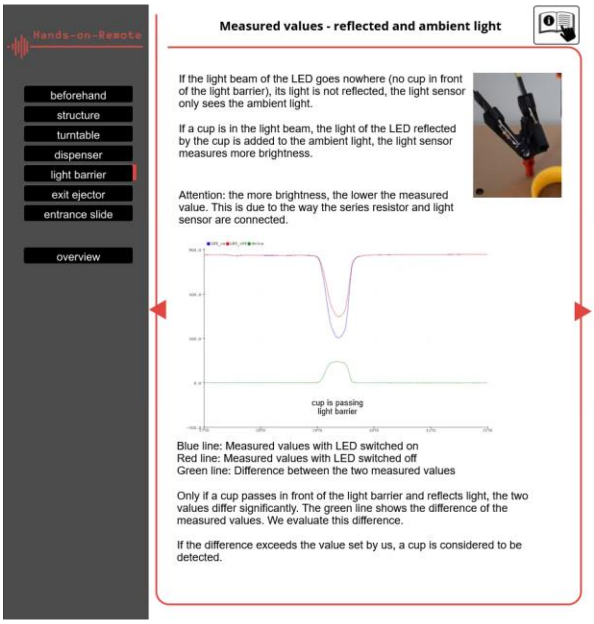

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Hardware - LED and LDR
An optical sensor, e.g. our LDR, and a light source, e.g. our LED, work together to form a light barrier. Unlike many other light barriers, the light beam is not interrupted. Our light barrier works with a reflection of the light beam; it is a reflex light barrier.
LED and LDR aim at the same point.
The light barrier consists of two active components …
If a cup moves into the light cone of the LED, it reflects its light and the LDR registers the increased brightness.
The LED illuminates …
The LDR measures …
Notice the LDR

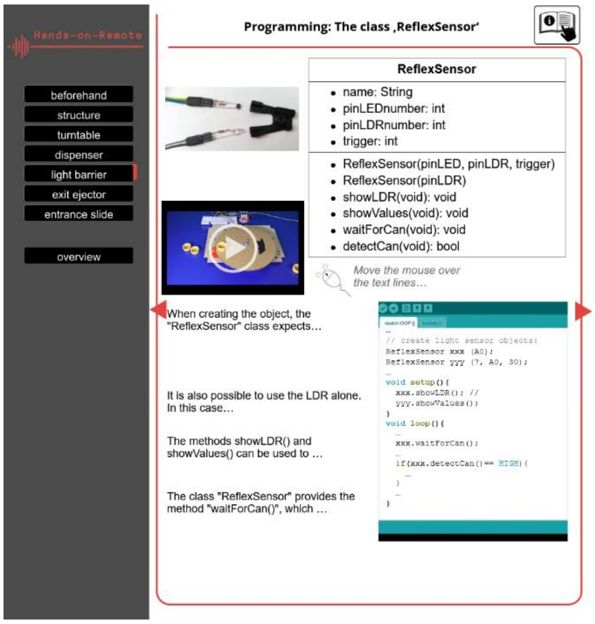

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Hardware - Light barrier
Next step in the mechanical design: The reflex light barrier.
The cup is to stop reliably in front of the dispenser. For this purpose, an optical sensor detects its position.
Move the mouse over the image…
In the following, first, mount the light barrier, then connect both the LDR (light sensor) and the LED (light source) to the Arduino and finally program the Arduino.

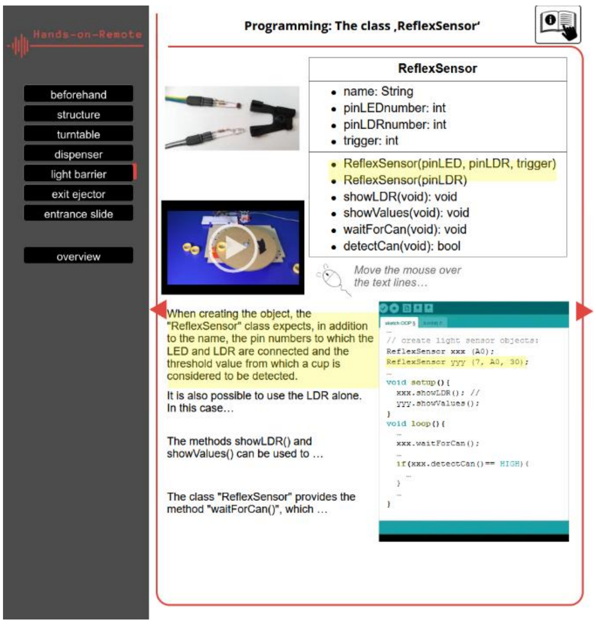

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Hardware - Mounting the light barrier
Mount the reflex light barrier: The light barrier holder can be fixed in the base plate with a stand.
Use the axis stopper to adjust the light barrier to the appropriate height.:
LDR and LED are already soldered with …
Note: Connect the single continuous leg to …

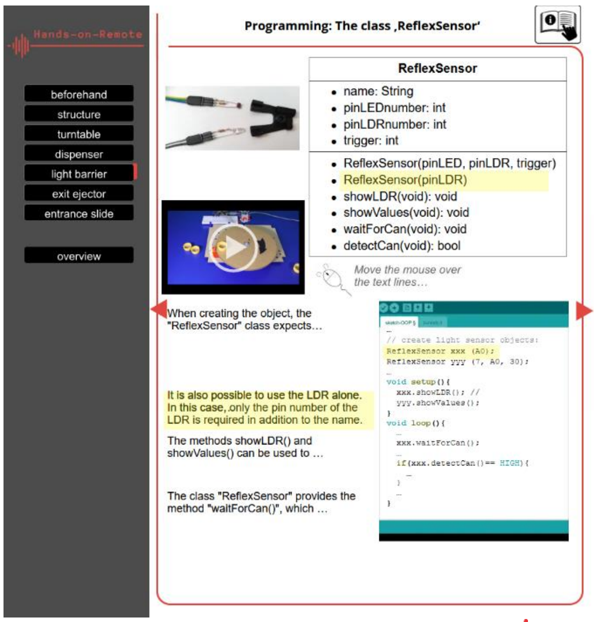

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Wiring - light barrier
Please disconnect the Arduino from the computer before changing the wiring!
Work on the Arduino only in a de-energized state!
Now connect the LDR (light sensor) and the LED (light source) to the Arduino.
Use the supply bus (red and blue) on the breadboard again:
at the arduino: at the breadboard:
red wire: V in - Pin -> red power bus
blue wire: GND - Pin -> blue ground bus
1. Connect LDR:
Yellow -> Pin A0
Green -> power bus
Blue -> GND bus
Move the mouse over the picture…
2. Connect LED:
Gray -> Pin 7
Black -> GND
(you do not need the white cable for the LED).
Dispenser
Turntable

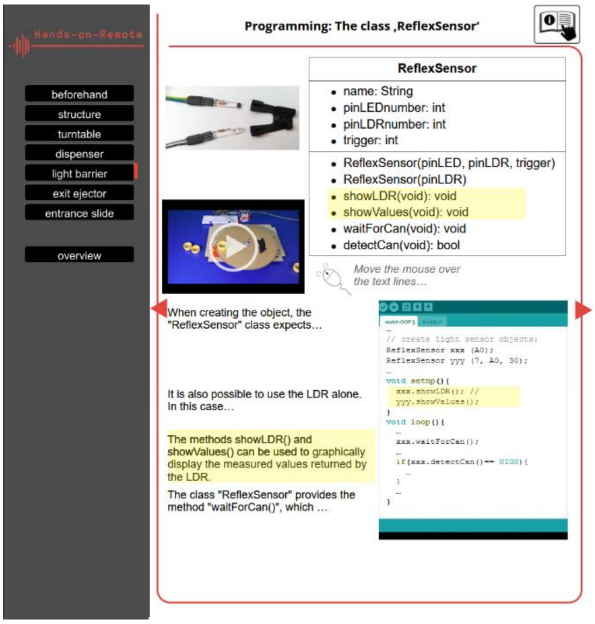

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Wiring - light barrier
Please disconnect the Arduino from the computer before changing the wiring!
Work on the Arduino only in a de-energized state!
Now connect the LDR (light sensor) and the LED (light source) to the Arduino.
Use the supply bus (red and blue) on the breadboard again:
at the arduino: at the breadboard:
red wire: V in - Pin -> red power bus
blue wire: GND - Pin -> blue ground bus
1. Connect LDR:
Yellow -> Pin A0
Green -> power bus
Blue -> GND bus
2. Connect LED:
Gray -> Pin 7
Black -> GND
(you do not need the white cable for the LED).

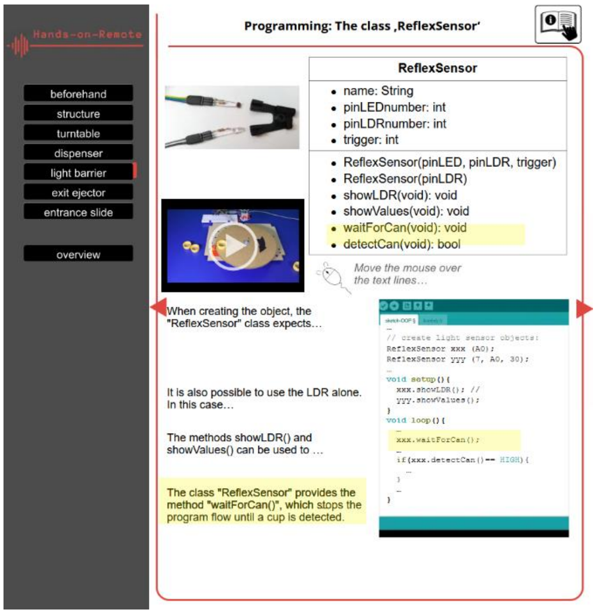

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Measured values - reflected and ambient light
If the light beam of the LED goes nowhere (no cup in front of the light barrier), its light is not reflected, the light sensor only sees the ambient light.
If a cup is in the light beam, the light of the LED reflected by the cup is added to the ambient light, the light sensor measures more brightness.
Attention: the more brightness, the lower the measured value. This is due to the way the series resistor and light sensor are connected.
cup is passing light barrier
Blue line: Measured values with LED switched on
Red line: Measured values with LED switched off
Green line: Difference between the two measured values
Only if a cup passes in front of the light barrier and reflects light, the two values differ significantly. The green line shows the difference of the measured values. We evaluate this difference.
If the difference exceeds the value set by us, a cup is considered to be detected.

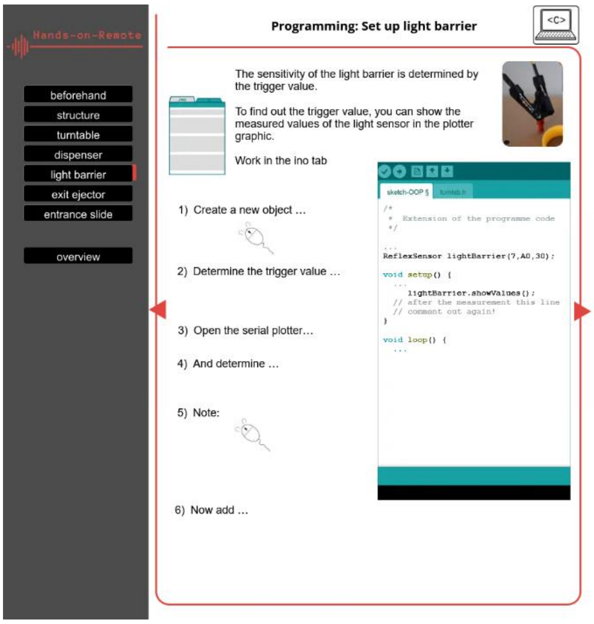

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: The class ‚ReflexSensor'
ReflexSensor
name: String
pinLEDnumber: int
pinLDRnumber: int
trigger: int
ReflexSensor(pinLED, pinLDR, trigger)
ReflexSensor(pinLDR)
showLDR(void): void
showValues(void): void
waitForCan(void): void
detectCan(void): bool
Move the mouse over the text lines…
When creating the object, the "ReflexSensor" class expects…
It is also possible to use the LDR alone. In this case…
The methods showLDR() and showValues() can be used to …
The class "ReflexSensor" provides the method "waitForCan()", which …
// create light sensor objects:
ReflexSensor xxx (A0);
ReflexSensor yyy (7, A0, 30);
void setup(){
xxx.showLDR(); //
yyy.showValues();
}
void loop(){
xxx.waitForCan();
if(xxx.detectCan()== HIGH){
}
}

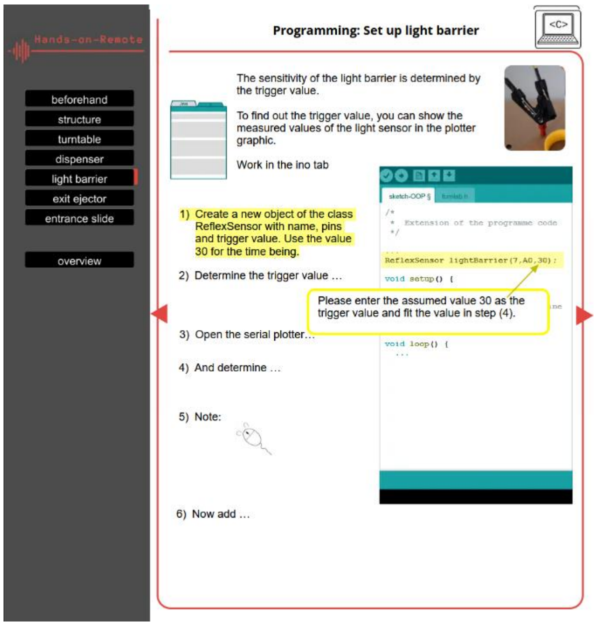

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: The class ‚ReflexSensor'
ReflexSensor
name: String
pinLEDnumber: int
pinLDRnumber: int
trigger: int
ReflexSensor(pinLED, pinLDR, trigger)
ReflexSensor(pinLDR)
showLDR(void): void
showValues(void): void
waitForCan(void): void
detectCan(void): bool
Move the mouse over the text lines…
When creating the object, the "ReflexSensor" class expects, in addition to the name, the pin numbers to which the LED and LDR are connected and the threshold value from which a cup is considered to be detected.
It is also possible to use the LDR alone. In this case…
The methods showLDR() and showValues() can be used to …
The class "ReflexSensor" provides the method "waitForCan()", which …
// create light sensor objects:
ReflexSensor xxx (A0);
ReflexSensor yyy (7, A0, 30);
void setup(){
xxx.showLDR(); //
yyy.showValues();
}
void loop(){
xxx.waitForCan();
if(xxx.detectCan()== HIGH){
}
}

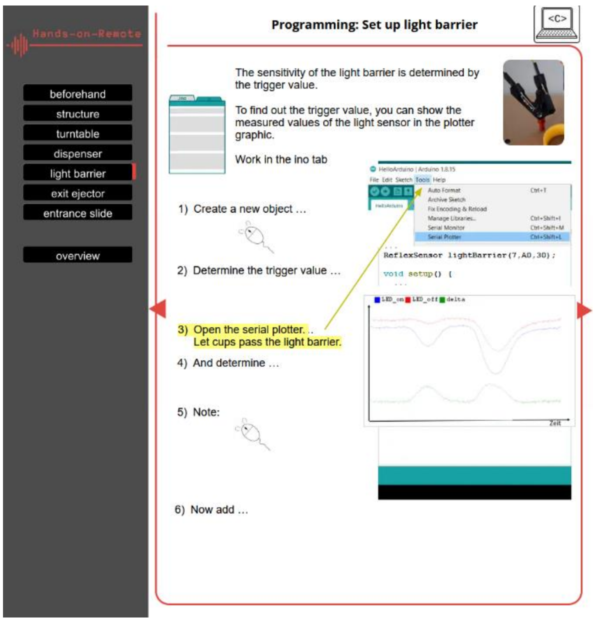

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: The class ‚ReflexSensor'
ReflexSensor
name: String
pinLEDnumber: int
pinLDRnumber: int
trigger: int
ReflexSensor(pinLED, pinLDR, trigger)
ReflexSensor(pinLDR)
showLDR(void): void
showValues(void): void
waitForCan(void): void
detectCan(void): bool
Move the mouse over the text lines…
When creating the object, the "ReflexSensor" class expects…
It is also possible to use the LDR alone. In this case, only the pin number of the LDR is required in addition to the name.
The methods showLDR() and showValues() can be used to …
The class "ReflexSensor" provides the method "waitForCan()", which …
// create light sensor objects:
ReflexSensor xxx (A0);
ReflexSensor yyy (7, A0, 30);
void setup(){
xxx.showLDR(); //
yyy.showValues();
}
void loop(){
xxx.waitForCan();
if(xxx.detectCan()== HIGH){
}
}

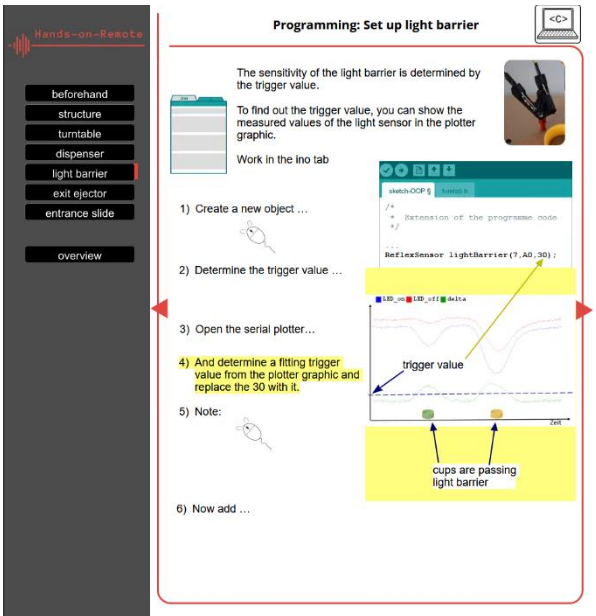

Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: The class ‚ReflexSensor'
ReflexSensor
name: String
pinLEDnumber: int
pinLDRnumber: int
trigger: int
ReflexSensor(pinLED, pinLDR, trigger)
ReflexSensor(pinLDR)
showLDR(void): void
showValues(void): void
waitForCan(void): void
detectCan(void): bool
Move the mouse over the text lines…
When creating the object, the "ReflexSensor" class expects…
It is also possible to use the LDR alone. In this case…
The methods showLDR() and showValues() can be used to graphically display the measured values returned by the LDR.
The class "ReflexSensor" provides the method "waitForCan()", which …


Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: The class ‚ReflexSensor'
ReflexSensor
name: String
pinLEDnumber: int
pinLDRnumber: int
trigger: int
ReflexSensor(pinLED, pinLDR, trigger)
ReflexSensor(pinLDR)
showLDR(void): void
showValues(void): void
waitForCan(void): void
detectCan(void): bool
Move the mouse over the text lines…
When creating the object, the "ReflexSensor" class expects…
It is also possible to use the LDR alone. In this case…
The methods showLDR() and showValues() can be used to …
The class "ReflexSensor" provides the method "waitForCan()", which stops the program flow until a cup is detected.


Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: Set up light barrier
The sensitivity of the light barrier is determined by the trigger value.
To find out the trigger value, you can show the measured values of the light sensor in the plotter graphic.
Work in the ino tab
1) Create a new object …
2) Determine the trigger value …
3) Open the serial plotter…
4) And determine …
5) Note:
6) Now add …
ReflexSensor lightBarrier(7,A0,30);
void setup() {
lightBarrier.showValues();
// after the measurement this line
// comment out again!
void loop() {


Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: Set up light barrier
The sensitivity of the light barrier is determined by the trigger value.
To find out the trigger value, you can show the measured values of the light sensor in the plotter graphic.
Work in the ino tab
1) Create a new object of the class ReflexSensor with name, pins and trigger value. Use the value 30 for the time being.
2) Determine the trigger value …
3) Open the serial plotter…
4) And determine …
5) Note:
6) Now add …
Please enter the assumed value 30 as the trigger value and fit the value in step (4).
ReflexSensor lightBarrier(7,A0,30);
void setup() {
void loop() {


Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: Set up light barrier
The sensitivity of the light barrier is determined by the trigger value.
To find out the trigger value, you can show the measured values of the light sensor in the plotter graphic.
Work in the ino tab
1) Create a new object …
2) Determine the trigger value …
3) Open the serial plotter. Let cups pass the light barrier.
4) And determine …
5) Note:
6) Now add …
ReflexSensor lightBarrier(7,A0,30);
void setup() {


Hands-on-Remote
beforehand
structure
turntable
dispenser
light barrier
exit ejector
entrance slide
overview
Programming: Set up light barrier
The sensitivity of the light barrier is determined by the trigger value.
To find out the trigger value, you can show the measured values of the light sensor in the plotter graphic.
Work in the ino tab
1) Create a new object …
2) Determine the trigger value …
3) Open the serial plotter…
4) And determine a fitting trigger value from the plotter graphic and replace the 30 with it.
5) Note:
6) Now add …
ReflexSensor lightBarrier(7,A0,30);
trigger value
cups are passing light barrier

Deutsches Museum

INSTITUTO DE EDUCAÇÃO
—
ULISBOA

CENTRUM NAUKI KOPERNIK

TUM
Technische Universität München

All materials are available at

https://erasmus-plus.ec.europa.eu/projects/search/details/2020-1-DE02-KA226-VET-008295

The Hand-on Remote project was funded in the Erasmus+ KA226 Partnerships for Digital Education Readiness program (2020-1-DE02-KA226-VET-008295)

Co-funded by the Erasmus+ Programme of the European Union

Hands-on-Remote