Klaus Trimborn, Dipl.-Eng. Claas Niehues, Micaela Krieger-Hauwede

STEM Coding Ultimate

Robotics, AI, and Modern Technology – Discover Them in a Realistic and Hands-On Way

Klett MEX

fischertechnik

STEM Coding Ultimate

2-4 students

224 components incl. spare parts bag and simple re-sorting

10 models with 13 lessons and **96+ hours teaching material**

Incl. TXT 4.0 controller, 2 × encoder motors, 2 × mini push buttons, 5 × LEDs, photo transistor, NTC resistor, track sensor, ultrasonic sensor, USB camera, 9-V rechargeable battery with USB-C charging port and charging cable

More information, quick start guides, and video tutorials are available here:
→ www.fischertechnik.de/en/stem-coding-ultimate

All tasks and building instructions are integrated into the **fischertechnik app, STEM Suite**.
Just download the free app and get started.
The app is available for iOS, Android, Windows, and macOS.

LEGAL NOTICE

fischertechnik
→ www.fischertechnik.de/en/schools

Klett MEX
→ www.klett-mex.de

Project Management: Tobias Schlipf, Ann-Christin Walker, fischertechnik GmbH

Implementation: Tanja Zunder, Klett MEX GmbH

Authors: Klaus Trimborn, Dipl.-Eng. Claas Niehues, Micaela Krieger-Hauwede

Edited by: Micaela Krieger-Hauwede, Leipzig

Design: Tanja Bregulla, Orangepixel

Didactic concept, pedagogical consulting: Klett MEX GmbH

CONTENTS

MEANING OF THE ICONS

Model level of difficulty:

easy

moderate

difficult

Programming level of difficulty:

easy

moderate

difficult

Lesson:

with app

without app

Action- and problem-oriented construction and programming made clear for upper secondary school!

Tips for teachers

STEM Coding Ultimate

- **Complete everyday tasks in information technology and robotics**
- **Skill orientation based on the 4C model:
  Creativity, collaboration, communication, and critical thinking**
- **Blockly and Python programming with the versatile TXT 4.0 controller**
- **224 components, incl. spare parts bag and simple re-sorting**

Teaching material
for more than 96 lessons

Number of students
2–4 per assembly kit

Time required
All basic tasks can be solved in 2–4 double lessons.

Number of models
10 (with 13 lessons)

Learning objectives
- Expand fundamentals of computer science and robotics
- Understand how actuators and complex sensors work
- Acquire basic knowledge of neural networks and AI programming

Age group
Upper secondary school

LEARNING MATERIAL

Digital skills and a basic understanding of more complex programming are now part of the advanced knowledge that students in upper secondary school must acquire. With STEM Coding Ultimate this advanced knowledge can be acquired through motivating and didactically prepared everyday examples.

In an independent, app-guided way, students learn – starting with a manageable basic task and moving on to real-life applications – how to implement more complex circuit technology challenges and programming tasks. The two basic tasks **Parking garage 1: Parking barrier** and **Traffic light control 1: Signal system** can be flexibly expanded into team tasks in **Parking garage 2: Entrance and exit** and **Traffic light control 2: Intersection**, as well as into advanced programming tasks in **Parking garage 3: License plate recognition** and **Traffic light control 3: AI phase control**.

Four lessons on three vehicle models make it possible, depending on the students' performance level, to address issues related to autonomous driving (simple driving practice, digital or analog control of an AGV, AI-based line following using a neural network).

The individual lessons in STEM Coding Ultimate address the questions: How does one learn to equip a versatile TXT 4.0 controller with sensors and actuators? How can several TXT 4.0 controllers be linked together to solve complex tasks? How can an automatic syringe pump be used for fine dosing of medication? How can a sensor-controlled automatic access gate be programmed? How can a simple parking barrier be expanded through teamwork into a parking system with separate entrance and exit? How can license plate recognition with automatic parking fee calculation be implemented? How does the control of a simple traffic signal system work? How are two traffic signal systems coordinated into an intersection? What are the basics of autonomous driving? Which sensors are used in this process? How can neural networks and AI be applied in traffic light control and autonomous driving?

PRODUCT DESCRIPTION

Through ten real-life models and independently realizable lessons, students learn step by step and problem-oriented to implement exciting tasks from the field of information technology. Two of the models can be expanded into team tasks, where students acquire basic knowledge in linking two TXT 4.0 controllers. In addition, two of the models offer lessons on the use of neural networks and AI. With the help of a versatile TXT 4.0 controller, a range of practical sensors and actuators, an intuitive app, and high-quality functional fischertechnik building blocks, students in upper secondary school work on tasks that they know from their own everyday life. The hands-on and problem-based learning approach helps students develop subject-specific, methodological, personal, and social skills at different levels.

CURRICULUM ALIGNMENT

The references to the educational plans/core curricula are based on the common reference framework for natural sciences, information technology, and technology (GeRRT). This forms the basis for cross-state comparability of technology and computer science skills, as well as for the development of corresponding curricula, textbook creation, design of school teaching materials, or the development of extracurricular teaching and learning concepts.

The common reference framework is divided into five areas of competence. These five areas form the basis of the projects.

Understanding technology:

- Analyze and record a complex technical system functionally, hierarchically, and structurally.
- A simple example shows that every technological development and use of technology has consequences for society.

Using technology:

- Select a technical system typical for everyday use based on relevant technical criteria.
- If necessary, adequately analyze the malfunctioning of the technical system and identify and implement possible steps to solve the problem.

Developing technology:

- Break down a technical problem into sub-problems by analyzing requirements and the situation and assess the technical feasibility of solving the sub-problems.
- Develop or refine appropriate solutions for sub-problems, also based on existing solutions.
- Develop plans for partial solutions by adapting existing plans or developing your own partial solutions using digital tools.
- Use the partial solutions to develop the overall solution with the required (also digital) tools, devices, materials, and machines.

Evaluating technology:

- Evaluate a simple technical situation according based on individual criteria.

Communicating technology:

- Purposefully procure technical representations and evaluations for the technical system.
- Read and understand an existing technical description and instructions for the system in the appropriate language and/or images.
- Describe the technical system with standard-oriented technical representations as well as your own technical solutions using technical terminology adapted to the target audience.

MAIN LEARNING OBJECTIVES

- Achieve everyday tasks in information technology and robotics.
- Transfer model concepts to common applications.
- Practice and enhance project and group work in a team.
- Develop professional, methodological, personal, and social skills.
- Learn about program sequences and complex computer science structures.
- Learn Blockly or Python programming for a versatile TXT 4.0 controller.
- Understand and apply the functionality and networking of actuators and sensors.
- Preprofessional orientation in mathematics, natural sciences, (information) technology.

INSTRUCTIONS FOR WORKING WITH THE MATERIALS

The basic working and connecting techniques of fischertechnik as well as the functionality and structure of a program are reviewed in the first introductory lesson on analog sensors. Thirteen lessons for ten models are available. Three of the lessons can be expanded into team tasks or complex programming tasks. The materials can also be used to develop independent solutions to topics from everyday life. Information on implementation, media, organizational forms, or differentiation is provided for all models.

The materials for teachers are structured according to the following general pattern:

- key questions on the topic,
- presentation of the teaching idea,
- description of the task,
- everyday relevance,
- subject reference within the curriculum,
- lesson plan, possibly divided into consecutive lessons,
- methodological-didactic notes with options for differentiation and motivational aspects,
- programming skills,
- notes on additional materials,
- functional descriptions of the individual models, and
- material lists.

The tasks for the students are also structured according to a recurring pattern:

- storytelling for introduction,
- introductory and advanced tasks on the topic,
- design and construction phase of the project,
- instructions for transferring the program code to the TXT 4.0 controller,
- detailed development and explanation of the program code,
- step-by-step commissioning of the model in parallel,
- troubleshooting assistance,
- options for differentiation, and
- a section with information on related topics or curiosities to conclude the project.

More information, quick-start guides, and video tutorials are available here:
→ www.fischertechnik.de/en/stem-coding-ultimate

INSTRUCTIONS FOR WORKING WITH THE APP

All tasks and building instructions are integrated into the fischertechnik “STEM Suite” app.

Just download the free app and get started. The app is available for iOS, Android, Windows, and macOS.

The projects and tasks in the app are worked on independently by the students according to the pattern described above. The given order of the tasks and solutions enable all students to achieve a functioning result (model) and gain a sense of accomplishment.

MODEL 1

Analog sensors

Are the parameters correct?

KEY QUESTIONS:

- **How can sensors be used to measure resistance, temperature, and distances? *(Communication and information)***
- **Which sensors can be used to monitor production effectively? *(Collaboration)***
- **Under what conditions should the system report deviations from the norm? *(Critical thinking)***
- **What must be taken into account so that the sensor system can be used for different application scenarios and the system functions as robustly as possible? *(Creativity)***

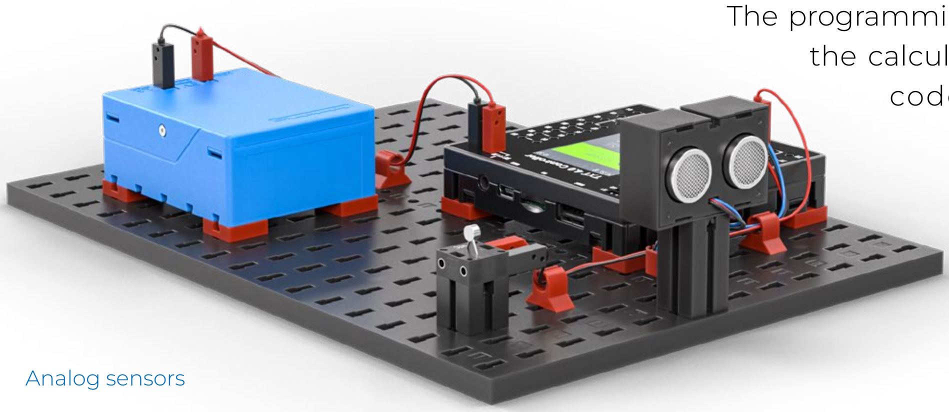
THE TEACHING CONCEPT AT A GLANCE

Grade level:	11–13
Time required:	2 double lessons (expandable up to 3 DL)
Degree of difficulty:	Model
	Programming
Model type:	Tabletop model with analog sensors

MODEL DESCRIPTION / TASK

The students plan and implement an experimental setup for experiments with analog sensors. Starting with the simple resistance display in the interface test, they develop their own graphic display on the screen of the TXT 4.0 controller and measure resistance and temperature.

The programming is expanded to include the calculation and display of color codes (three- or four-digit, differentiation 1 and 2) for layer resistors.



EVERYDAY RELEVANCE

Measuring resistances and temperatures is familiar to the students from science classes or from their everyday life.

In particular, embedding the topic in a realistic application context has a high motivational value.

The topic could be integrated into preprofessional orientation with regard to IT-related occupational fields. Here, the automated recording of physical quantities is used in many areas. Automated heating and/or filling is used not only in industry but increasingly also in the home environment.

SUBJECT REFERENCE

Information technology:	Advanced programming, conditional loops, functions, graphic displays
Physics:	Resistance in circuits, temperatures, distance measurements
Technology:	Stable construction, construction technology
Mathematics:	Calculation of terms, working with units

LESSON PLAN

Introductory phase

Classroom discussion

- Announce the topic.
- Ask what the essential characteristics of quality assurance are in the filling/production of additives.
- Ask about measured quantities and scenarios in which automatic measuring filling systems are used.
- Discuss possible applications of the scenarios collected (e.g. coffee machine, kitchen machine, dishwasher).
- Determine requirements for the experimental setup.

Support, if necessary

- Show sensors and components from the kit, use presentation media if needed.

Planning Phase

Classroom discussion

- The procedure for building the model and the target function are developed jointly.
- The work steps in the app are specified or discussed.

Partner or individual work	• The students familiarize themselves with the app and load the corresponding task. • The students define meaningful functions of a measurement using the sensors connected to the TXT 4.0 controller. • The students use the app to create the requirements list for the experiment to be built.
Optional: Partner or group work	• The students sketch possible setups for the measurement using the sensors (resistances, ultrasonic sensor). • The students discuss the results in the group and choose a design.

Construction Phase

Partner or individual work	• The students use the app for the experimental setup. The app guides them step by step through the construction.

Programming Phase

Partner or individual work	• The students write the program for the measurements to be carried out with the experimental setup. The app guides them step by step through the programming task; help is available in the app. • The program is transferred to the TXT 4.0 controller after each differentiation step.

Experimentation and Test Phase

Partner or group work	• The system is put into operation. • Possible malfunctions in the functional sequence must be found and eliminated. • Troubleshooting is possible with suggestions in the app. • Possible optimizations of the hardware and the programming can be carried out.

Final Phase

Optional: Presentation and allocation of differen-tiations	• Students eligible for differentiation may be addressed by the teacher if necessary. The graphical display of the color coding of resistors is presented. • The app provides concrete ideas for interested students, including the display of three- or four-digit color codes on the display of the TXT 4.0 controller.
Discussion in plenary	• Project debriefing in class. • Clarification of future possible applications in everyday life

METHODOLOGICAL AND INSTRUCTIVE TIPS

Differentiation options

Depending on the duration of the lesson series and the abilities of the students,

- the result of resistance measurement can be displayed not only as a number but also on a scale display as well as a three- or four-digit color code,
- the experimental setup can be modified to perform temperature measurements on selected objects,
- the experimental setup can be changed to measure and display the filling levels of drinking vessels during filling.

Motivational aspects

The topics "automatic filling" and "quality monitoring" are familiar to all students from everyday life. In many households, coffee and washing machines, along with many other smart applications, have long been part of everyday life. Partially autonomous cooking machines are increasingly used in meal preparation.

PROGRAMMING SKILLS

- Program start
- Infinite loop **repeat forever**
- Integration of sensors
- Programming of the display on the TXT 4.0 controller
- Loop **repeat while**
- Command **wait**
- Use of variables and their change
- Working with subprograms

ADDITIONAL MATERIALS

Optional download:
- Circuit diagram
- Building instructions

- Drawing media (paper, whiteboard, or projection screen).

FUNCTIONS OF THE MODEL AND THEIR TECHNICAL SOLUTIONS

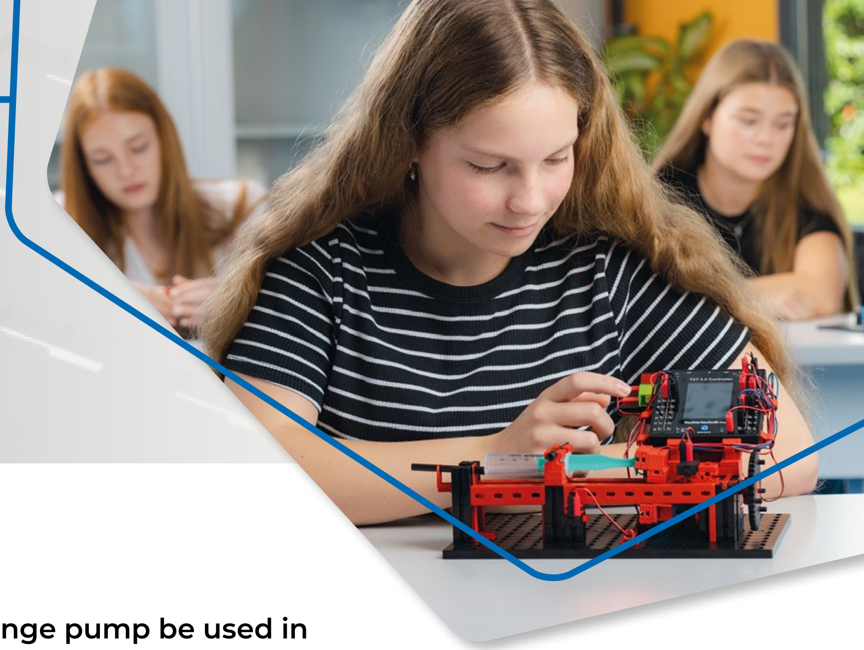
Function of the sensors/actuators	Technical solution
Measurement of resistances	Measurement of conductivity (quality criterion) of the additive
Measurement of temperatures	Monitoring of filling temperature
Measurement of distances	Monitoring of filling amounts
Differentiation options for advanced lessons	• Supplementing the numerical display with a scale display • Integration of color codes and/or other resistance types • Output of an alarm in case of under- or overtemperature • Measurement of distances and calculation of the difference (minimum – maximum)

MATERIAL LIST

Sensors	Function
1 On/Off push button on the TXT 4.0 controller	Switch on/start measurement/ stop measurement
1 NTC resistor	Resistance measurement, temperature determination
1 ultrasonic sensor	Distance measurement

Actuators	Function
1 display of the TXT 4.0 controller	Visualization of measurement results (as digits or as graphic)

MODEL 2



Syringe pump

It's all about the dose!

KEY QUESTIONS:

- **Where can an automatic medical syringe pump be used in everyday life? *(Communication)***
- **Which functions must the syringe pump control reasonably fulfill? *(Collaboration)***
- **Under what conditions should the system move the syringe? *(Critical thinking*)**
- **What must be considered so that the control can be used at different locations and the system functions as robustly as possible? *(Creativity)***

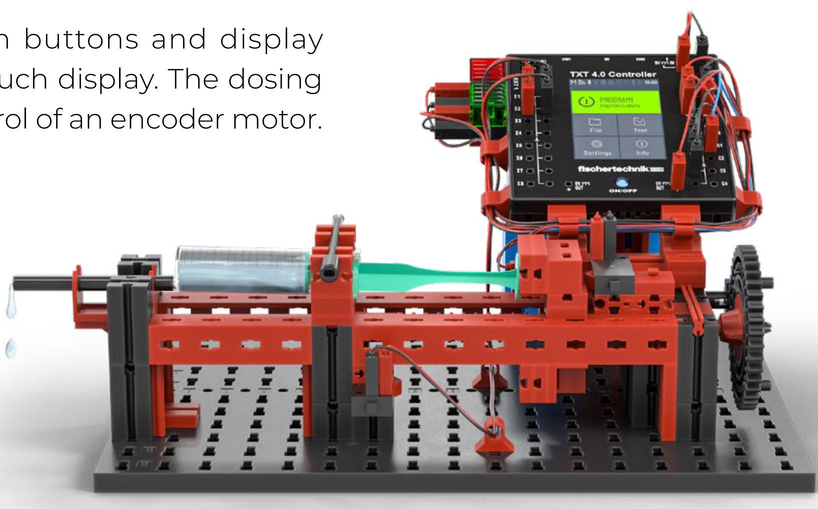
THE TEACHING CONCEPT AT A GLANCE

Grade level:	11–13
Time required:	3 double lessons (expandable up to 6 DL)
Degree of difficulty:	Model
	Programming
Model type:	Tabletop model for syringe pumps

MODEL DESCRIPTION / TASK

The students plan and implement a medical syringe pump for the fine dosing of medication.

Control elements such as selection buttons and display elements are programmed on the touch display. The dosing is precisely achieved by stepwise control of an encoder motor.



Syringe pump

Two LEDs provide light signals during dosing operation (green) or in the end position and if the fluid volume is insufficient for a dose (red). Via touch buttons, various functions such as *Start*, *Reverse*, *Dose preselection*, and *Injection time* are implemented step by step. From the number of motor revolutions, the currently injected dose as well as the remaining quantity of medication in the syringe are determined and displayed. If the remaining quantity is insufficient for a pre-selected dose, the start is blocked. Likewise, the motor stops when the syringe is fully pushed in. In both cases, the red LED blinks.

When prompted to change the syringe and automatically at program start, the syringe drive returns to the start position and stops automatically.

EVERYDAY RELEVANCE

Medical syringe pumps are precision dosing devices that administer medications or fluids in precisely controlled amounts over a specific period of time. Significantly, they are frequently used in medicine, for example in anesthesia, intensive care, or pain therapy.

In addition, syringe pumps can be used for precise dosing in biological or chemical experiments and manufacturing processes.

SUBJECT REFERENCE

Information technology:	Advanced programming, conditional loops, functions
Physics:	Change of movement
Technology:	Stable construction, construction technology
Mathematics:	Conditional counting, volume and time calculation
Biology, Chemistry:	Experiments with dosing tasks: In chemical and biological experiments, the syringe pump can be used across subjects to perform precise dosing.

LESSON PLAN

Introductory phase		
	Classroom discussion	• Announce the topic. • Ask about the essential features of the syringe pump.
	Support, if necessary	• Show sensors, actuators and components from the assembly kit, use presentation media if necessary.

Planning Phase	
Classroom discussion	• The procedure for building the model and the target function are developed jointly. • The work steps in the app are specified or discussed.
Partner or individual work	• The students familiarize themselves with the app and load the corresponding task. • The students define meaningful functions of an automatic syringe pump. • The students use the app to create the requirements list for the syringe pump to be built.
Optional: Partner or group work	• Optionally, the students sketch possible setups of the syringe pump with additional sensors (end switch) and actuators (LED). • The students discuss the results in the group and choose a design.
Construction Phase	
Partner or individual work	• The students use the app to build the syringe pump. The app guides them through the program in short steps.
Programming Phase	
Partner or group work	• The students write the program for the syringe pump. The app guides them step by step with open questions through the programming task. • The app offers assistance. • The program is transferred to the TXT 4.0 controller after each differentiation step.
Experimentation and Test Phase	
Partner or group work	• The syringe pump is put into operation. • Possible malfunctions in the functional sequence must be found and eliminated. • Potential troubleshooting is possible based on suggestions in the app. • Possible optimizations of the hardware and the programming can be carried out.
Final Phase	
Discussion in plenary	• Project debriefing in class. • Clarification of future possible applications in everyday life

METHODOLOGICAL AND INSTRUCTIVE TIPS

Technical explanation
In medical practice, professional syringe pumps work with highly precise step dosing. Instead of allowing a liquid to flow continuously, the desired volume flow is achieved by repeatedly delivering the smallest individual doses (e.g. 0.01 ml or even 0.001 ml). This principle ensures maximum accuracy and safety, especially with potent medications in intensive care or emergency medicine.

Didactic advantages
In the model, a microdose of 0.1 ml is preset. Dosing in such discrete steps can be implemented particularly clearly in programming terms, as it can be easily translated into loops or time-controlled routines. The dosing speed can be easily adjusted by changing either the number or the frequency of microdoses – such considerations encourage an active understanding of regulation and control. In addition, the procedure is intuitively understandable for the students: It is more tangible to consider how many individual steps are needed to dispense, for example, 1 ml of liquid than to imagine a continuous flow abstractly.

Differentiation options
Depending on the working speed of the individual students, variations in the minimum dose and the injection speed can be programmed (Differentiation option 1). Furthermore, the touch display can be expanded with a slider that allows the injection time to be set continuously (Differentiation option 2).

Motivational aspects
Working with syringe pumps connects technology with meaningful application – it is not just about motors and sensors, but about how technology can save lives. For boys, but especially for girls, this medical technology context can be particularly motivating because it shows that programming and building are not abstract, but meaningful and socially relevant. Those who tinker here are working on solutions that help people – this makes technology tangible, exciting, and valuable.

PROGRAMMING SKILLS

- Program start
- Infinite loop **repeat forever**
- Integration of sensors and actuators
- Programming of the display on the TXT 4.0 controller
- Integration of buttons on the touch display
- Loop **repeat n times**
- Command **wait**
- Use of variables and their change
- Working with subprograms
- Working with threads
- Working with events

Optional download:
- Circuit diagram
- Building instructions

ADDITIONAL MATERIALS

- Drawing media (paper, whiteboard, or projection screen).

FUNCTIONS OF THE MODEL AND THEIR TECHNICAL SOLUTIONS

Function of the sensors/actuators	Technical solution
Execution of a rotation of an encoder motor	Controlling the drive motor for a rotation with a fixed step size
Light signals red/green LED	Controlling the LED for operation and error messages
Calculation and display of information on the touch display	Programming and evaluation of amounts and times

MATERIAL LIST

Sensors	Function
1 On/Off push button on the TXT 4.0 controller	1. Switching on the syringe pump 2. Emergency stop of the syringe pump
2 mini push buttons	Detect the start and end positions

Actuators	Function
1 encoder motor	Movement
2 LEDs (1 × red, 1 × green)	Status display

MODEL 3

Gate model

Access granted?

KEY QUESTIONS:

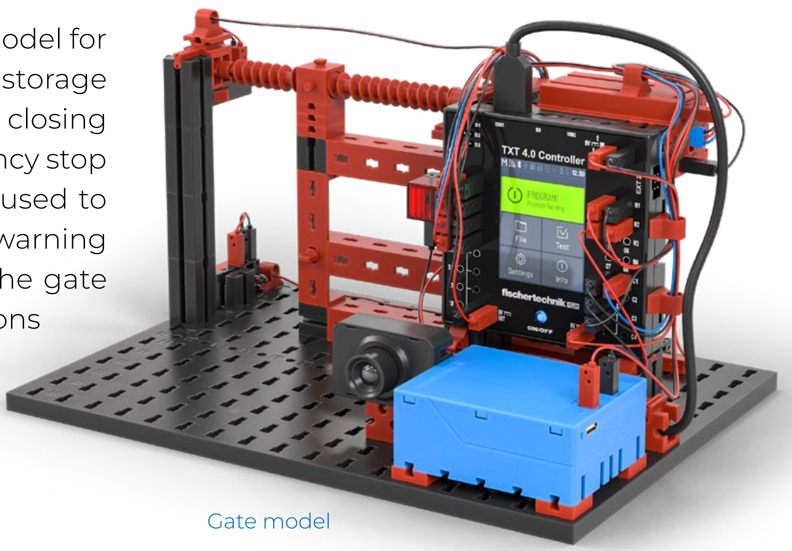
- How can end stops, interruptions, and distances be measured with sensors? *(Communication and information)*
- Which sensors can be used to meaningfully monitor and control gate operation? (Collaboration)
- Under what conditions should the system prioritize user protection over function? *(Critical thinking)*
- What must be considered so that the sensors can be used for different application scenarios and the system functions as robustly as possible? *(Creativity)*

THE TEACHING CONCEPT AT A GLANCE

Grade level:	11–13
Time required:	2 double lessons (expandable up to 8 DL)
Degree of difficulty:	Model
	Programming
Model type:	Gate model for access control

MODEL DESCRIPTION / TASK

The students plan and implement a gate model for sensor-based access control of a fictitious storage room. Starting from a simple opening and closing of the gate model controlled by an emergency stop switch, in a second step end switches are used to protect the mechanics, and in a third step warning LEDs are used to visualize the activity of the gate drive. A simple graphic display of the buttons *Open*, *Close*, and *Emergency Stop* is to be programmed on the display of the TXT 4.0 controller and put into operation.



Gate model

To avoid human error in monitoring the closing process and the associated pressing of the "Emergency Stop" button, closing control is to be automated by a light barrier that detects possible obstructions when the gate model closes and automatically triggers an emergency stop. At the same time, an optical signal in the form of permanently lit red LEDs indicates the stop of the closing process. Releasing the gate to close again should only be possible when there is no longer any obstruction in the gate and the "Open" button on the display of the TXT 4.0 controller has been pressed. Compared to the first solution with end switches, this automatic safety circuit represents a significant advancement in the development of automation of the gate model.

In the next step, automatic opening of the gate upon the approach of a person from the inside should be made possible through the use of an ultrasonic sensor. It is assumed that the legality of the desire to open has already been ensured when entering the storage room. If a person or a vehicle approaches the gate from the inside up to 20 cm, the gate should automatically initiate the opening process. Again, the LEDs should blink during the movement of the gate. Access to the storage area from outside should be restricted through the integration of color recognition using the USB camera. If a green object approaches, the gate should open automatically; with any other color, it should either remain locked or close.

An extended access control should be implemented via programming of a simple code lock. For this purpose, 7 buttons are placed on the display of the TXT 4.0 controller. Six of them are used to enter the code, and another to transmit the code and initiate the opening process if the code entry was correct. The code lock is a simple one, where the order of entries or multiple entries are not taken into account. It should only check whether keys have been pressed. However, it is left up to the students to program their own more complex code locks to protect the storage room.

EVERYDAY RELEVANCE

Embedding the topic in a realistic application context has high motivational value. Integration of the topic into preprofessional orientation could take place with regard to mechatronics or IT-related occupational fields. Here, automated opening and closing is used in many areas. Automatic opening of doors or roller shutters is used not only in industry but also increasingly in the home environment.

SUBJECT REFERENCE

Information technology:	Advanced programming, conditional loops, functions, coding, graphic displays
Physics:	Light barrier, distance measurement with ultrasonic sensor
Technology:	Stable construction, control of actuators, reading sensors, construction technology
Mathematics:	Calculation of terms

LESSON PLAN

Introductory phase

Classroom discussion

- Announce the topic.
- Ask about the essential requirements for the automatic opening or closing of a gate or door model.
- Ask about scenarios in which automatically opening and closing gate/door systems are used.
- Discuss possible applications of the scenarios collected (e.g. garage door, yard gate, access gate, sliding door, roller shutter).
- Determine requirements for the experimental setup.

Support, if necessary

- Show sensors, actuators and components from the assembly kit, use presentation media if necessary.

Planning Phase

Classroom discussion

- The procedure for building the model and the target function are developed jointly.
- The work steps in the app are specified or discussed.

Partner or individual work

- The students familiarize themselves with the app and load the corresponding task.
- The students define meaningful functions of the control with the TXT 4.0 controller.
- The students use the app to create the requirements list for the gate model to be built.

Optional: Partner or group work	• The students sketch possible setups of the gate model themselves and construct it freely. • The students discuss the results in the group and choose a design.

Construction Phase

Partner or individual work	• The students use the app to build the gate model. The app guides them through the program in short steps.

Programming Phase

Partner or individual work	• The students write the programs for the control tasks to be carried out with the experimental setup. The app guides them step by step through the programming task. • The app offers assistance. • The program is transferred to the TXT 4.0 controller after each differentiation step.

Experimentation and Test Phase

Partner or group work	• The experimental setup is put into operation. • Possible malfunctions in the functional sequence must be found and eliminated. • Troubleshooting is possible with suggestions in the app. • Possible optimizations of the hardware and the programming can be carried out.

Final Phase

Optional: Presentation and allocation of differentiation	• Students eligible for differentiation may be addressed by the teacher if necessary. Here, the option of programming an extended code lock is presented.
Discussion in plenary	• Project debriefing in class. • Clarification of future possible applications in everyday life

METHODOLOGICAL AND INSTRUCTIVE TIPS

Differentiation options

Depending on the duration of the lesson series and the abilities of the students:

- the introductory scenario of opening and closing of the gate model without end switches and with manual monitoring may be omitted,
- the experimental setup can be changed and sensors installed/used in other places,
- the code lock can be improved,
- a competition “Who cracks the code?” for the best programming of a code lock can follow (conversion of the gate model into a safe that must either be cracked or protected).
- Further application scenarios can be formulated by the students and implemented along with setup and programming.

Motivational aspects

The topic of automatic control of movement is familiar to all students from everyday life. In many households, automatically opening roller shutters, awnings, or garage doors, along with many other smart applications, have long been part of everyday life. Electronic, code-based door locks are increasingly used for building access control.

PROGRAMMING SKILLS

- Program start
- Integration of sensors
- Integration of actuators
- Integration of the TXT 4.0 controller display
- Integration of inputs via a touch display
- Camera configuration
- Infinite loop **repeat forever**
- Condition **if – do**
- Command **wait**
- Command **wait until**
- Working with subprograms
- Use of events and threads

ADDITIONAL MATERIALS

Optional download:

- Circuit diagram
- Building instructions

- If available, materials for programming code locks can be used.
- Use drawing media (paper, whiteboard, or projection screen).

FUNCTIONS OF THE MODEL AND THEIR TECHNICAL SOLUTIONS

Function of the sensors/actuators	Technical solution
Red LEDs	Control LEDs during gate movement
Light barrier interruption	Prevent gate movement
Ultrasonic sensor	Proximity detection
Differentiation	Design of variously complex gate protections or gate controls (code lock, combination of sensor inputs)

MATERIAL LIST

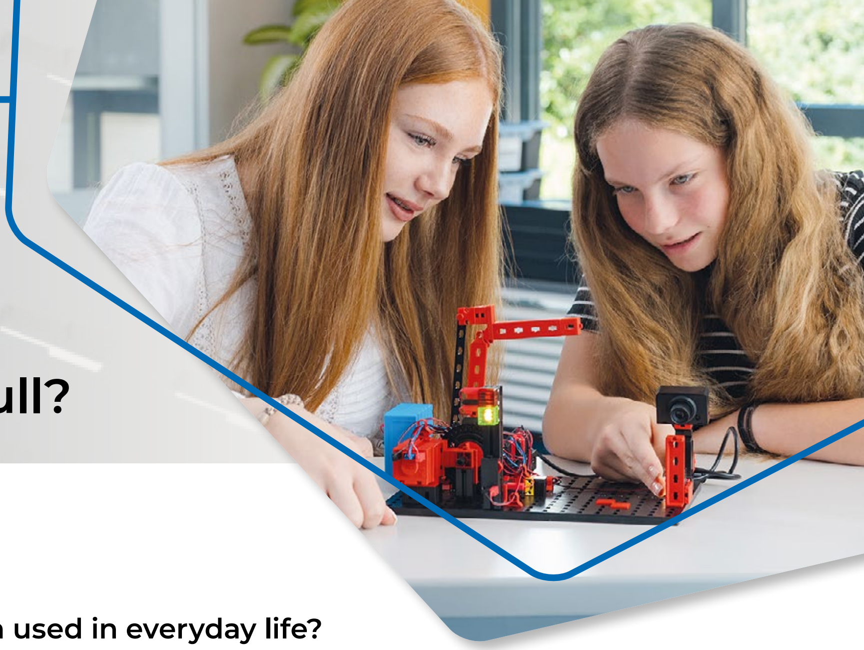
Sensors	Function
1 On/Off push button on the TXT 4.0 controller	1. Switch on the barrier control 2. Emergency stop of the barrier
1 ultrasonic sensor	Proximity detection
1 USB camera	Color recognition
1 photo transistor	Light barrier interruption detection, receiver
2 mini push buttons	End switch for gate control

Actuators	Function
1 TXT 4.0 controller display	Display
2 LEDs (2 × red)	Warning
1 encoder motor	Movement of the gate
1 LED (white)	Illumination of light barrier sensor

MODELS 4 AND 5

Parking garage 1–3

Is the parking garage full?



KEY QUESTIONS:

- **Where is an automatic barrier system used in everyday life? *(Communication)***
- **Which functions must the barrier control reasonably fulfill? *(Collaboration)***
- **Under what conditions should the system open or close? *(Critical thinking)***
- **What must be considered so that the control can be used at different locations and the system functions as robustly as possible? *(Creativity)***

THE TEACHING CONCEPT AT A GLANCE

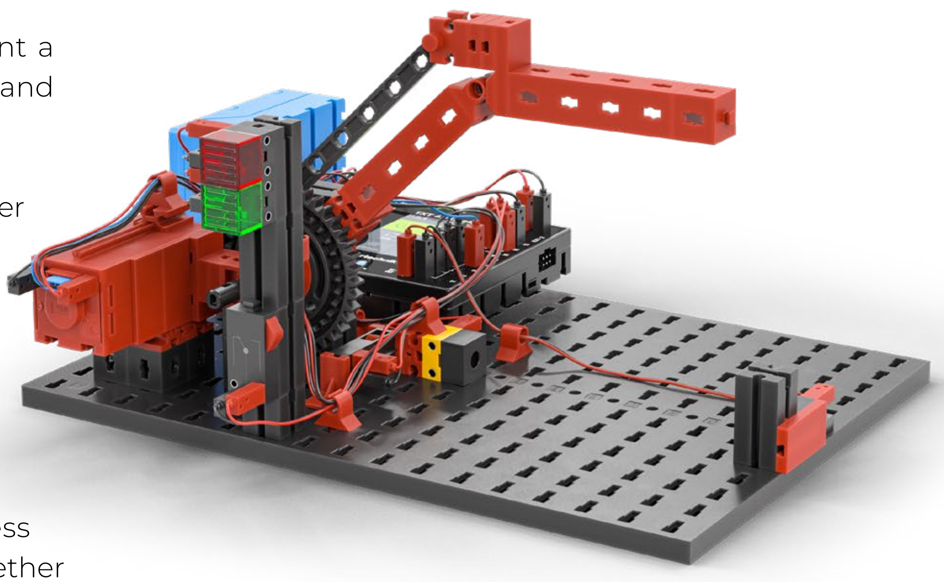
Grade level:	11–13
Time required:	2 double lessons per unit (expandable up to 10 DL)
Degree of difficulty:	Model up to
	Programming up to
Model type:	Tabletop model for parking garage barriers

MODEL DESCRIPTION / TASK

The students plan and implement a parking garage barrier for entry and exit of a parking garage.

The requirement to open the barrier is initially made via a push button (entry) or a button on the TXT 4.0 controller display (exit).

Two LEDs provide light signals for full opening (green) or during the opening and closing process (red). A light barrier monitors whether a vehicle is in the barrier area during



Parking garage barrier

closing. From the number of entries and exits, the number of free parking spaces is determined and displayed on the TXT 4.0 controller display; when full, the red LED blinks (Parking garage 1).

In a second step, two groups join together and link their models and controls so that one barrier serves as the entry and the second as the exit (Parking garage 2).

In the fully developed model, the barrier control is supplemented by optical license plate recognition using the USB camera and automatic parking fee calculation (Parking garage 3).

EVERYDAY RELEVANCE

Barrier-controlled access to parking spaces and garages is familiar to the students from everyday life.

In particular, automatic access through optical license plate recognition has a high motivational value.

The topic could be integrated into preprofessional orientation with regard to IT-related occupational fields. Automated switching by recording physical variables is used here in many fields. Automated object recognition in access technology is becoming increasingly important.

SUBJECT REFERENCE

Information technology:	Advanced programming, conditional loops, functions
Physics:	Change of movement
Technology:	Stable construction, construction technology
Mathematics:	Angles, conditional counting, time and price calculation

LESSON PLAN

Introductory phase

Classroom discussion	• Announce the topic. • Ask what the essential features of barrier control are.
Support, if necessary	• Show sensors, actuators and components from the assembly kit, use presentation media if necessary.

Planning Phase		
	Classroom discussion	• The procedure for building the model and the functions to be achieved are developed jointly. • The work steps in the app are specified or discussed.
	Partner or individual work	• The students familiarize themselves with the app and load the corresponding task. • The students define meaningful functions of an automatically opening and closing barrier. • The students use the app to create the requirements list for the barrier system to be built.
	Optional: Partner or group work	• Optionally, the students sketch possible setups of the barrier with additional sensors (light barrier) and actuators (LED). • The students discuss the results in the group and choose a design.
Construction Phase		
	Partner or individual work	• The students use the app to build the barrier. The app guides them through the program in short steps.
Programming phase of the parking garage barrier (Parking garage 1)		
	Partner or individual work	• The students develop the program for the barrier system. The app guides them step by step with open questions through the programming task. • The app offers assistance. • The program is transferred to the TXT 4.0 controller after each differentiation step.
Experimentation and test phase of the parking garage barrier (Parking garage 1)		
	Partner or individual work	• The parking garage barrier is put into operation. • Possible malfunctions in the functional sequence must be found and eliminated. • Potential troubleshooting is possible based on suggestions in the app. • Possible optimizations of the hardware and the programming can be carried out.

Final/follow-up phase of the parking garage barrier (Parking garage 1)

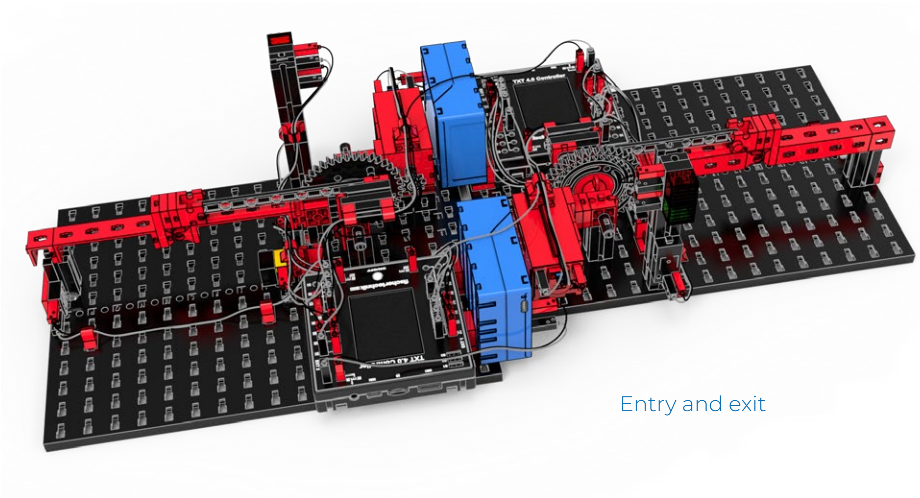
Optional: Presentation and allocation of differentiations

- Students (or the entire group) eligible for differentiation may be addressed by the teacher. Here, the combination of two barriers for separate control of entry and exit is introduced in group work.
- As a further differentiation (or directly after the first program without subsequent group work), the USB camera and license plate recognition and evaluation in the app are offered. The app provides concrete ideas for, e.g., time recording and calculation of parking fees.

Programming phase of entry and exit (Parking garage 2)

Partner or individual work

- The groups decide which barrier model is responsible for entry and which for exit.
- Linking of the two TXT 4.0 controllers according to learning cards
- Division of program adjustments for entry and exit separately (possibly also joint work of both groups)


Entry and exit

Experimentation and test phase of entry and exit (Parking garage 2)

Partner or group work

- Commissioning of the system
- Checking correct communication of the two TXT 4.0 controllers
- If applicable. Troubleshooting and correction

Final/follow-up phase of entry and exit (Parking garage 2)

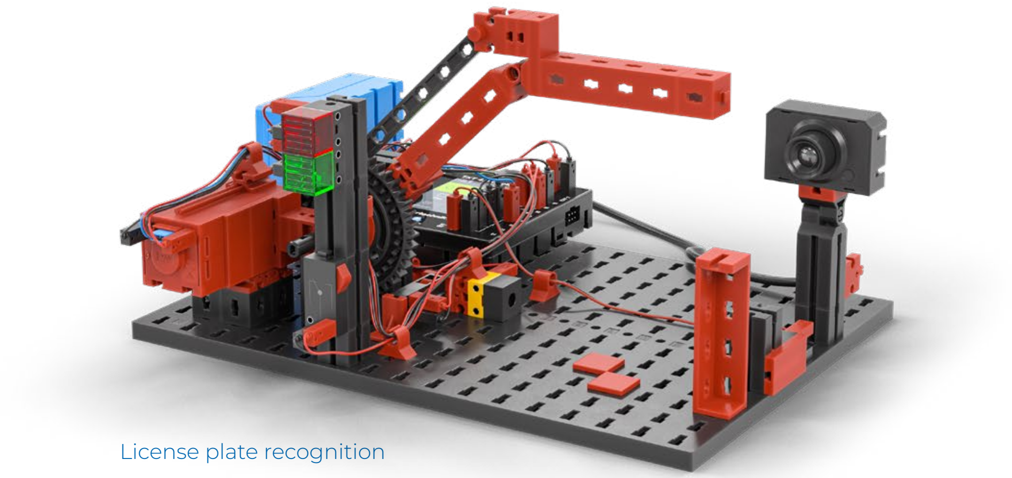
Optional: Presentation and allocation of further differentiation

- Now the USB camera and license plate recognition and evaluation in the app are offered. The app provides specific ideas for, e.g., time recording and calculation of parking fees.

Programming phase of license plate recognition (Parking garage 3)

Partner or individual work

- If necessary, undo the linking of two barriers and continue working in the individual groups
- Connect the USB camera
- Handling of the barcode and its decoding as well as storage in a new program
- Load the program from Parking garage 1
- Design of the TXT 4.0 controller display according to learning cards
- Step-by-step extension of the program: Recording and storing incoming license plates, calculating parking duration and fees of exiting vehicles
- Simulation of the payment process and opening of the barrier, including calculation and display of available parking spaces



License plate recognition

Experimentation and test phase of license plate recognition (Parking garage 3)

Partner or group work

- Commissioning of the system
- Test and, if necessary, troubleshoot the individual programming steps
- Test the overall program
- If applicable. Troubleshooting and correction

Final phase of license plate recognition (Parking garage 3)

Optional: Presentation and allocation of further differentiation

- Groups working particularly efficiently can be given the extension task of storing incoming electric and regular cars in separate tables and, for example, reserving 4 parking spaces for electric cars and 6 parking spaces for regular cars. Parking fees can also be calculated in a differentiated way. Independent work without learning cards is intended here.

Discussion in plenary

- Project debriefing in class.
- Clarification of future possible applications in everyday life

METHODOLOGICAL AND INSTRUCTIVE TIPS

Differentiation options

Depending on the duration of the lesson series and the strengths of the students, the following are possible:

- two groups combine their systems as entry and exit of a parking garage,
- they link two TXT 4.0 controllers (entry/exit) to synchronize counting results,
- in Parking garage 3, they use the USB camera to record license plates for access, calculate parking time for exit, and display parking fees on the display based on the recorded data.

Motivational aspects

The topic of automatic barrier systems is familiar to all students from everyday life. Automatic license plate recognition, which can also be applied to further automation processes, motivates intensive engagement and solution of a corresponding task.

PROGRAMMING SKILLS

- Program start
- Infinite loop **repeat forever**
- Integration of sensors and actuators
- Programming of display outputs
- Programming of touch buttons
- Camera configuration
- Loop **repeat until**
- Commands **wait** and **wait until**
- Counting loops
  count with – from ... to ... – by
- Use of variables and their change
- Working with subprograms
- Working with events and threads

Optional download:

- Circuit diagram
- Building instructions

ADDITIONAL MATERIALS

- Drawing media (paper, whiteboard, or projection screen).

FUNCTIONS OF THE MODEL AND THEIR TECHNICAL SOLUTIONS

Function of the sensors/actuators	Technical solution
Execution of a rotation of an encoder motor	Control the drive motor for exactly one rotation
Light signals red/green LED	Control the LED for free passage and stop
Light barrier interruption Light barrier interruption	Prevent barrier movement Count cars
License plate recognition via USB camera	Open the barrier
Parking garage 2: Connection of two TXT 4.0 controllers	• Separate control of entry and exit barrier • Data exchange between two TXT 4.0 controllers
Parking garage 3: License plate recognition via USB camera	• Barcode decoding • Programming and evaluation of start and end times • Calculation of parking fees

MATERIAL LIST

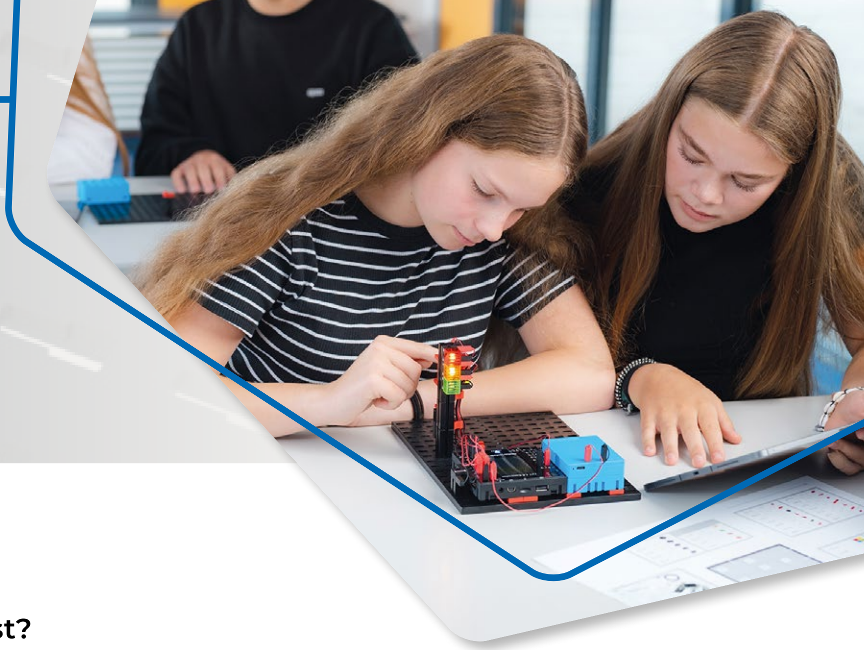
Sensors	Function
1 On/Off push button on the TXT 4.0 controller	1. Switch on the barrier control 2. Emergency stop of the barrier
1 photo transistor	Light barrier interruption detection, receiver
1 USB camera	Barcode recognition
1 mini push buttons	Request for entry
1 mini push buttons	Status query of the barrier (open or closed)

Actuators	Function
1 encoder motor	• Movement • Counting of rotations
2 LEDs (1 × red, 1 × green)	Status display
1 LED (white)	Illumination of light barrier sensor

MODELS 6 AND 7

Traffic light control 1–3

Is traffic flowing?



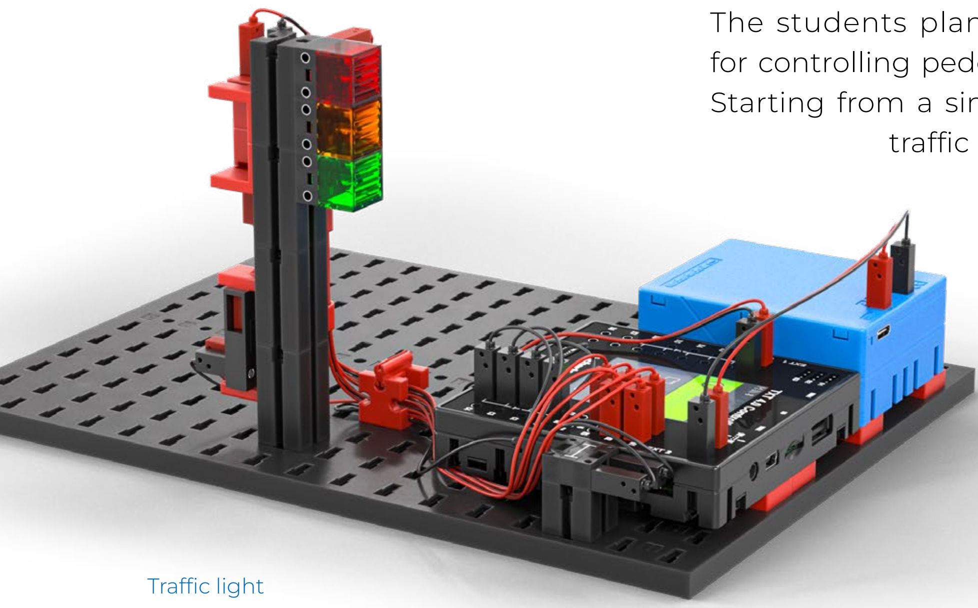
KEY QUESTIONS:

- **What types of traffic flow control exist?** ***(Communication)***
- **How can the traffic flows of motor vehicles and pedestrians be safely directed?** ***(Collaboration)***
- **What waiting times are reasonable for the different traffic participants?** ***(Critical thinking)***
- **How can traffic signals be operated safely in a time- and demand-controlled way?** ***(Creativity)***

THE TEACHING CONCEPT AT A GLANCE

Grade level:	11–13
Time required:	2–3 double lessons per unit (expandable up to 10 DL)
Degree of difficulty:	Model
	Programming up to
Model type:	Tabletop model for traffic light

MODEL DESCRIPTION / TASK


Traffic light

The students plan and implement a traffic light for controlling pedestrian and motor vehicle traffic. Starting from a simple time-controlled pedestrian traffic light, through a crossing traffic light for a main road, up to an intersection with motor vehicle and pedestrian flows as well as AI phase control of a crossing traffic light, the model develops step by step and so do the requirements for construction and programming.

The students learn to understand and read signal layout plans and signal timing plans. From the information in the signal timing plans, they determine possible switching groups that can be activated simultaneously and the associated switching times.

From these groups and times, the students develop a control program that can be tested for suitability using the traffic light model.

EVERYDAY RELEVANCE

Time-controlled activation of actuators is familiar to the students from technical science or computer science lessons or from their everyday life.

Embedding the topic in a realistic application context, which the students know very well from everyday life, provides a high motivational value.

Integration of the topic into preprofessional orientation could take place with regard to traffic engineering or electrical engineering occupational fields, where automated control of actuators is used in many areas. In particular, automated time control is used not only in industry but also increasingly in the home environment, e.g. with smart home applications.

SUBJECT REFERENCE

Information technology:	Advanced programming, conditional loops, functions, graphic displays, neural networks, AI
Physics:	Electrical circuit, movement, measurement of times
Technology:	Stable construction, signal timing plans, construction technology
Mathematics:	Calculation of terms

LESSON PLAN

Introductory phase

Classroom discussion

- Announce the topic.
- Ask what the essential characteristics for controlling traffic flows are.
- Ask for scenarios where crossing problems may occur.
- Discuss possible applications of the scenarios collected (e.g. pedestrian traffic light, vehicle traffic light, other traffic signals).
- Determine requirements for building the models.

Support, if necessary	• Show sensors, actuators and components from the assembly kit, use presentation media if necessary.

Planning Phase

Classroom discussion	• The procedure for building the model and the functions to be achieved are developed jointly. • The work steps in the app are specified or discussed.
Partner or individual work	• The students familiarize themselves with the app and load the corresponding task. • The students define meaningful functions for controlling a traffic light (TL) using the TXT 4.0 controller. • The students use the app to create the requirements list/signal timing table for the current use case.
Optional: Partner or group work	• The students measure the signal times of a pedestrian traffic light located near the school and create an adapted signal timing plan. • The students discuss the results in the group and work out the differences compared to the specified TL.

Construction phase of the traffic light (Traffic light control 1)

Partner or individual work	• The students use the app to build the TL. The app guides them through the program in short steps.

Programming phase of the traffic light (Traffic light control 1)

Partner or individual work	• The students write the program for controlling the TL. The app guides them step by step through the programming task. • The app offers assistance. • Extracting important information for controlling the traffic light from a signal layout and a signal timing plan, as well as defining the switching times and the successive switching states • Clustering of signal groups into switching phases • Programming of a simple time-controlled traffic light for vehicle and pedestrian traffic • The program is transferred to the TXT 4.0 controller after each differentiation step.

Experimentation and test phase of the traffic light (Traffic light control 1)

Partner or group work

- The traffic light model is put into operation.
- Possible malfunctions in the functional sequence must be found and eliminated.
- Troubleshooting is possible with suggestions in the app.
- Integration of a request control for the pedestrian traffic light
- Further development of the program into a day/night control with request control of the pedestrian traffic light
- Possible optimizations of hardware and programming can be carried out.

Final/follow-up phase of the traffic light (Traffic light control 1)

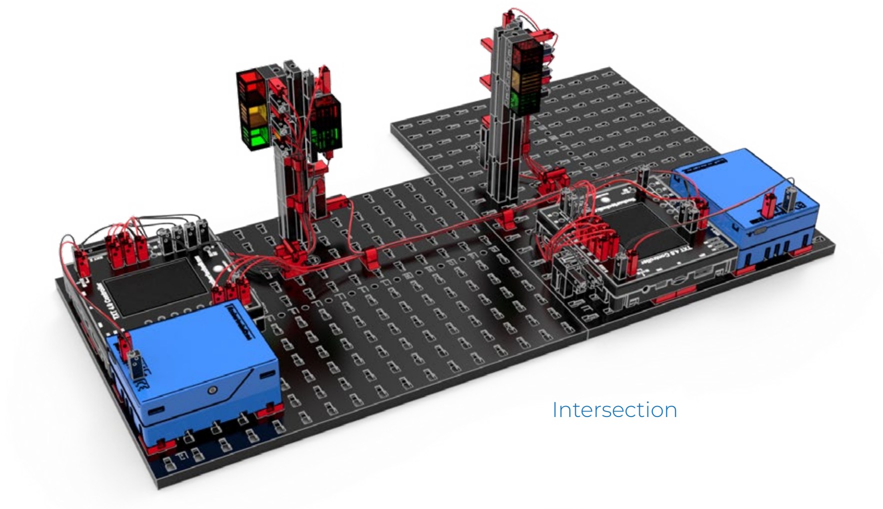
Optional: Presentation and allocation of differentiations

- Students (or the entire group) eligible for differentiation may be addressed by the teacher. Here, the waiting time model can be introduced.

Construction phase of the intersection (Traffic light control 2)

Partner or group work

- Construction of the partial intersection from two traffic light models
- The groups decide which traffic light model is responsible for which direction.



Intersection

Programming phase of the intersection (Traffic light control 2)

Partner or group work

- Linking of the two TXT 4.0 controllers according to learning cards
- Division of work for program adaptation to create complex traffic light control for an intersection (possibly also joint work by both groups)
- Programming of the two TXT 4.0 controllers as master and slave controllers

Experimentation and test phase of the intersection (Traffic light control 2)

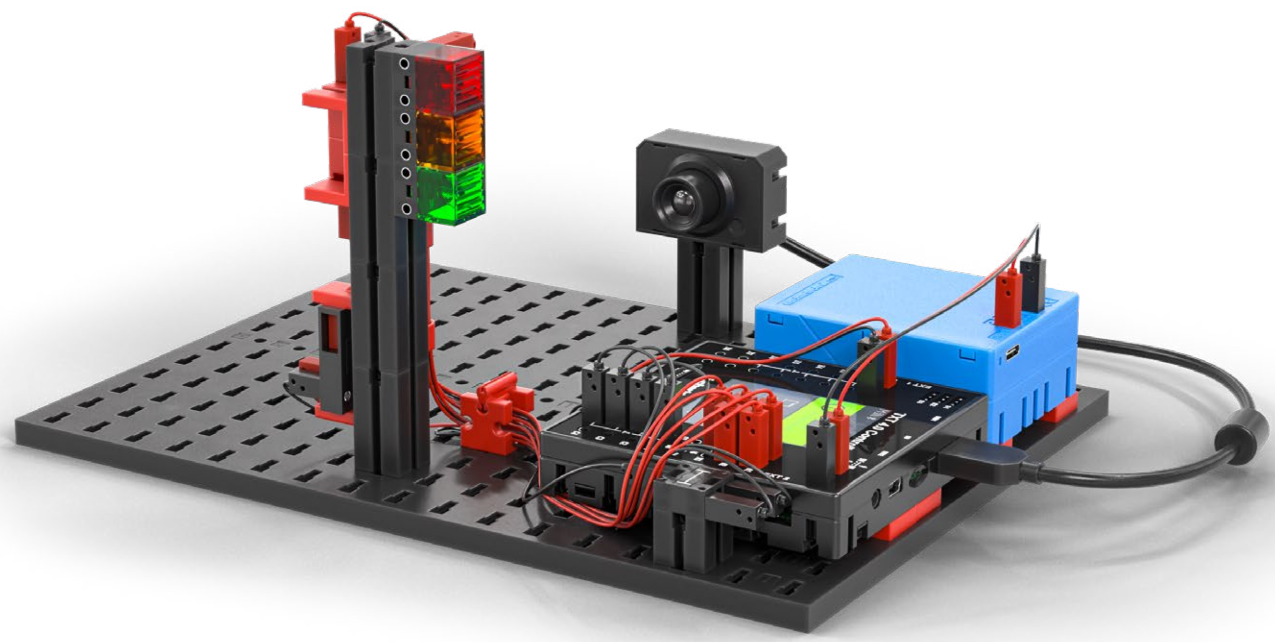
Partner or group work	• The partial intersection is put into operation. • Possible faults in the linking and in the functional sequence must be found and eliminated. • Troubleshooting is possible with suggestions in the app. • Further development of the program into more complex traffic light control is possible. • Possible optimizations of hardware and programming can be carried out.

Final/follow-up phase of the intersection (Traffic light control 2)

Optional: Presentation and allocation of further differentiation	• Students (or the entire group) eligible for differentiation may be addressed by the teacher. Here, the concepts of neural networks and AI are to be presented.

Construction phase of AI phase control (Traffic light control 3)

Partner or individual work	• Integration of the USB camera for object categorization into the individual traffic light (Traffic light control 1)



Traffic light with USB camera

Programming phase of AI phase control (Traffic light control 3)		
	Partner or individual work	• Configuration of the USB camera • Recording of training data for the AI model • Programming of AI-supported color recognition • Adjustment of pedestrian traffic light times according to the recognised crossing participant (wheelchair user or pedestrian) and the associated crossing times
Experimentation and test phase of AI phase control (Traffic light control 3)		
	Partner or group work	• Testing the programming using colored blocks • Testing the programs with simulated traffic participants (wheelchair user, pedestrian)
Final phase of AI phase control (Traffic light control 3)		
	Optional: Presentation and allocation of further differentiation	• Optional: Construction of a full intersection using several construction kits (possible only without AI)
	Discussion in plenary	• Project debriefing in class. • Clarification of future possible applications in everyday life

METHODOLOGICAL AND INSTRUCTIVE TIPS

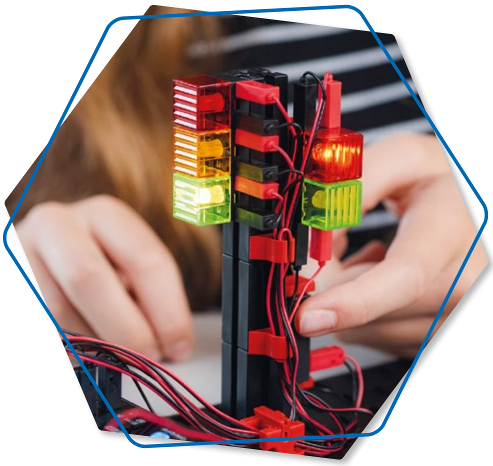
Differentiation options

Depending on the duration of the lesson series and the abilities of the students,

- The complexity of the requirements for the control of the TL is increased not only by the number of traffic signals but also by the number of traffic flows that must be safely crossed.
- A real measurement of the signal times of a TL near the school can be carried out, which could form the basis for modified control of the TL.
- The traffic light model can be extended through group work.
- With the help of the USB camera, recognition of traffic participants and, as a result, an adjustment of the phase lengths due to modified clearance times can be implemented.

Motivational aspects

The topic of automatic control of actuators is familiar to all students from everyday life. In many households, numerous smart applications are already part of daily life. Washing machines or dishwashers that switch on autonomously on a timer, centrally controlled roller shutters, or partially autonomous cooking machines are increasingly being used.



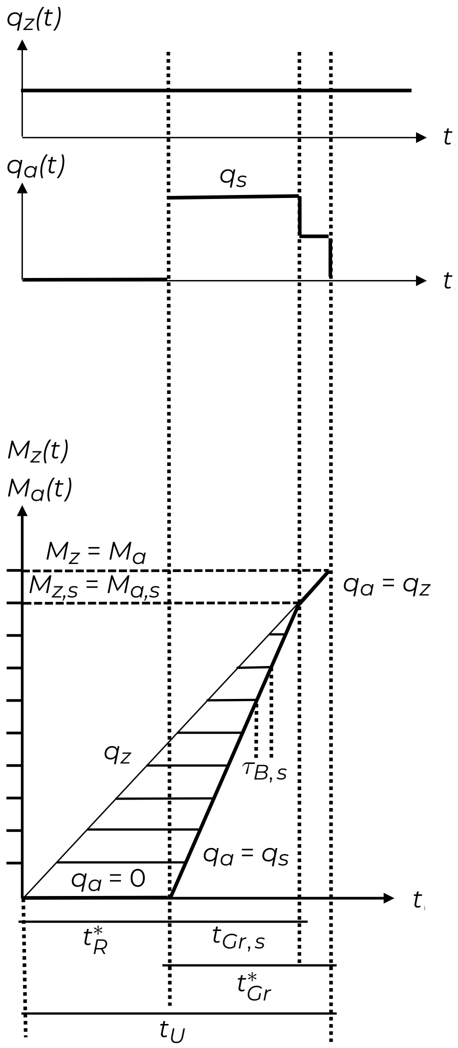
BASICS OF TRAFFIC ENGINEERING

The use of traffic engineering facilities for controlling traffic flows is familiar to both students and teachers.

The dimensioning of the TL represents a central task in traffic engineering to manage traffic flow efficiently and improve traffic safety. Several factors must be taken into account, including both traffic engineering and technical aspects.

First, a comprehensive traffic survey is conducted, in which traffic volumes, vehicle types, pedestrian flows, as well as cross-sections and traffic types are recorded. Such a survey could be carried out in lessons at a nearby signal-controlled intersection with the aim of creating local signal layout plans and signal timing plans. Based on this data, the phase times of the traffic light are determined, which regulate the ratio of green, yellow, and red phases.

The calculation of green times is based on the required throughputs and stopping times, while also considering the reaction times of traffic participants. A simplified waiting time model can be used for this:



q_Z: Inflow

q_a: Outflow

q_S: Saturated outflow

M_a: Number of discharged vehicles

$M_{a,S}$: Number of discharged vehicles up to end $t_{Gr,S}$

M_Z: Number of incoming vehicles

$M_{Z,S}$: Number of incoming vehicles up to end $t_{Gr,S}$

$\tau_{B,S}$: Required time value, gross headway of successive vehicles, when $q = q_S$

t_R^*: Effective blocking time

t_{Gr}^*: Effective release time

$t_{Gr,S}$: Green duration with saturated outflow

t_U: Cycle time

Traffic congestion occurs during the effective blocking time t_R^* with constant inflow q_Z. Congestion reduction takes place during the effective green time t_{Gr}^*. Until the congestion is resolved, the reduction takes place with the saturated outflow q_S (in the time $t_{Gr,S}$). After the congestion has been resolved, outflow q_a = inflow q_Z.

The following applies:

Green duration with saturated outflow: $t_{Gr,S} = \frac{q_Z}{q_S - q_Z} \cdot t_R^*$

Maximum waiting time: $w_{\max} = t_r^* = t_U - t_{Gr}^*$

Sum of waiting times per cycle: $W = M_{Z,S} \cdot \frac{t_R^*}{2}$

Maximum waiting time: $w_{\max} = t_R^* = t_U - t_{Gr}^*$

Average waiting time: $\overline{w} = \frac{t_R^* + t_{Gr,S}}{t_U} \cdot \frac{t_R^*}{2}$

Example: $q_Z = \frac{450\ \text{Fz}}{h}$, $t_U = 90\,\text{s}$, $t_{Gr}^* = 30\,\text{s}$ $\rightarrow$ $t_R^* = (90 - 30)\,\text{s} = 60\,\text{s}$

Green duration with saturated outflow: $t_{Gr,s} = \frac{450}{1800 - 450} \cdot 60\,\text{s} = 20\,\text{s}$

Number of obstructed vehicles: $M_{z,s} = 0.5\,\frac{\text{Fz}}{s} \cdot 20\,\text{s} = 10\,\text{Fz}$ (per cycle)

Maximum waiting time: $w_{\text{max}} = t_r^* = 60\,\text{s}$

Sum of waiting times per cycle: $W = 10\,\text{Fz} \cdot \frac{60\,\text{s}}{2} = 300\,\text{s}$ (per cycle)

Average waiting time: $\overline{w} = \frac{60 + 20}{90} \cdot \frac{60}{2}\,\text{s} = 26.7\,\text{s}$

It follows that with the given traffic situation, 300 s waiting time per cycle can be expected. From an economic perspective, the costs incurred due to non-productive working time are considerable. This results in the obligation to implement traffic light cycles with the shortest possible waiting times.

Another important aspect is the coordination of the TL within a networked traffic system in order to ensure optimal traffic flow along main routes. So-called green waves are calculated for this purpose, adapted to the respective traffic flows.

The technical design also includes the choice of signal transmitters, the positioning of sensors for traffic detection and compliance with safety standards and legal requirements. Overall, the dimensioning aims to minimize waiting times, improve traffic safety, and enable sustainable traffic management.

PROGRAMMING SKILLS

- Program start
- Infinite loop **repeat forever**
- Integration of sensors
- Integration of the TXT 4.0 controller display
- Programming of display outputs
- Programming of the touch button
- Camera configuration
- Loop **repeat until**
- Command **wait**
- Use of variables and their change
- Working with subprograms
- Programming of two TXT 4.0 controllers as master and slave controllers
- Development of neural networks
- Recording of AI training data

ADDITIONAL MATERIALS

Optional download:

- Circuit diagram
- Building instructions

- Drawing media (paper, whiteboard, or projection screen).

FUNCTIONS OF THE MODEL AND THEIR TECHNICAL SOLUTIONS

Function of the sensors/actuators	Technical solution
Traffic light (TL)	Time control of actuators
Demand registration	Adjustment of programming
Crossing simulation	Separation of traffic flows
Color recognition using USB camera	Control of green phase durations
Traffic light control 2: Connection of two TXT 4.0 controllers	Data exchange between two controllers based on the master-slave concept
Traffic light control 3: Color recognition using USB camera	• Development of a neural network • Collection of training data • Implementation of a classification problem using AI
Further differentiation: Adaptation of the TL to a real existing TL	Data collection, model building, phasing, programming, testing

MATERIAL LIST

Sensors	Function
1 On/Off push button on the TXT 4.0 controller	Switching on the traffic light control
1 mini push buttons	Demand registration
1 USB camera	Categorization of traffic participants

Actuators	Function
1 display of the TXT 4.0 controller	Visualization of the program
5 LEDs (2 × red, 2 × green, 1 × yellow)	Traffic light signals

Automated guided vehicle 1–4

Safe navigation in the course

KEY QUESTIONS:

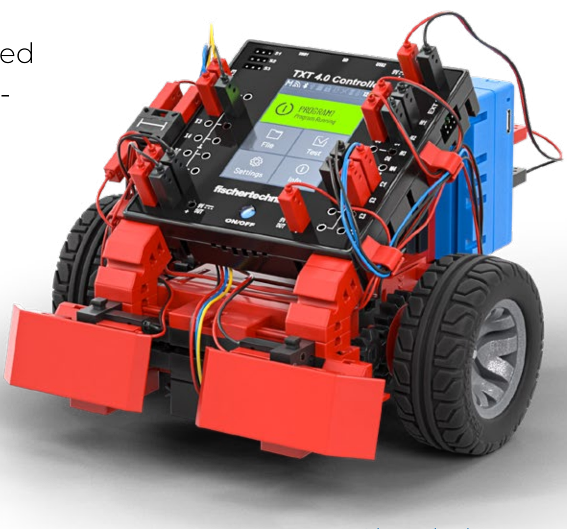
- Which sensors and control methods are suitable for a automated guided vehicle (AGV) in the course? *(Communication)*
- How can different sensor values (IR sensor, ultrasonic sensor, USB camera) be coordinated for safe navigation? *(Collaboration)*
- What compromises are reasonable between speed, precision, and safety? *(Critical thinking)*
- How can the behavior of the AGV be developed from rule-based to AI-supported control? *(Creativity)*

THE TEACHING CONCEPT AT A GLANCE

Grade level:	11–13
Time required:	2–3 double lessons per unit (expandable up to 13 DL)
Degree of difficulty:	Model
	Programming up to
Model type:	Tabletop models for automated guided vehicles

MODEL DESCRIPTION / TASK

The students plan and implement automated guided vehicles (AGV), which they gradually equip with additional sensors and intelligent control. Starting with a base vehicle with encoder motors that performs defined routes and turns as well as initial line following using a track sensor, then advancing to an AGV with an ultrasonic sensor for collision avoidance and a USB camera for interaction with colored surfaces, up to rule loop-controlled or even AI-supported line following, the models develop step by step – and with them the demands on setup, wiring, and programming.



Obstacle detector

The students learn to create state transition diagrams for driving states and to read and use sensor data in a targeted way. From the properties and measurement values of the sensors they determine suitable driving and steering groups, define variables for distance and angle control as well as reaction times for safe course corrections.

From these defined states and parameters, the students first develop rule-based control programs with variables, subprograms, and state logic; then they configure and train a neural network for a regression problem that predicts suitable motor speeds from sensor inputs and is extended through differentiation by the distance sensor. The students test the suitability in driving trials on the course – straight driving, lane keeping, obstacle avoidance, and color surface reactions – and improve their solutions through systematic troubleshooting and speed optimization.

EVERYDAY RELEVANCE

Sensor-based vehicle control is familiar to the students from technology or computer science lessons and their everyday lives. Familiar applications include cruise control and lane keeping assist in cars, e-scooters with sensors, and robotic vacuum cleaners that follow lines and avoid obstacles.

Embedding in a realistic mobility context creates strong motivation, as students can directly recognize parallels to urban traffic and warehouse logistics.

Integration of the topic into preprofessional orientation is appropriate in automotive engineering, electrical engineering, and robotics/automation, where sensor-driven control and automated actuator control are key competencies.

The combination of sensor technology and AI-supported control is encountered by students not only in industry and traffic but also in the home environment, e.g. in smart home applications, intelligent heating, or automatic lighting – everywhere that measured values trigger decisions and movements are reliably executed.

SUBJECT REFERENCE

Information technology:	Advanced programming, conditional loops, functions, state machines, event control, camera integration, P and PD controllers, neural networks, training of a neural network
Physics:	Movement (distance, time, speed), signal processing, ultrasonic time-of-flight measurement and sound propagation, inertia and braking distance, color recognition, control engineering
Technology:	Stable construction, construction technology
Mathematics:	Calculation of terms, scaling, proportionality and linear functions, unit conversion, normalization, regression

LESSON PLAN

Introductory phase

Classroom discussion

- Announce the topic.
- Ask what the essential features are for controlling automated guided vehicles.
- Ask for scenarios in which automated guided vehicles occur.
- Discuss possible applications.
- Determine requirements for building the models.

Support, if necessary

- Show sensors, actuators and components from the assembly kit, use presentation media if necessary.

Planning Phase

Classroom discussion

- The procedure for building the model and the target function are developed jointly.
- The work steps in the app are specified or discussed.

Partner or individual work

- The students familiarize themselves with the app and load the corresponding task.
- The students define meaningful functions for controlling an AGV using the TXT 4.0 controller.
- The students use the app to create the sensor list for the current use case.
- The students prepare the course. The driving surface should be as smooth as possible.

Note: The driving experiments should preferably take place either on the floor or on a surface with edge boundaries, in order to avoid the vehicle models falling off.

Optional: Partner or group work

- The students discuss the sensors that can be used and sketch possible setups.
- The students discuss their experiences with automated or autonomous driving in the group.

Construction phase for vehicle training (AGV 1)

Partner or individual work	• The students use the app to build the obstacle detector. The app guides them through the program in short steps. • The students check the wiring of the sensors using the interface test.

Programming phase for vehicle training (AGV 1)

Partner or individual work	• The students develop a program for simple straight driving as well as for targeted turns with the AGV. • The students calculate the necessary encoder pulses and make individual adjustments to the conversion factors *"pulses per cm"* and *"pulses per degree"* according to the conditions of the AGV. • The students extend the AGV with simple obstacle detection and create a simple state transition diagram to analyze the programming task. • The students develop a program for simple line following using digital control. • The app guides them step by step with open questions through the programming task. • The app offers assistance. • The program is transferred to the TXT 4.0 controller after each differentiation step.

Experimentation and test phase for vehicle training (AGV 1)

Partner or group work	• The obstacle detector is put into operation. • Possible malfunctions in the functional sequence must be found and eliminated. • Potential troubleshooting is possible based on suggestions in the app. • Possible optimizations in hardware and programming, e.g. adjustments to driving speed and adjustments to the motor speed difference when turning, can be carried out.

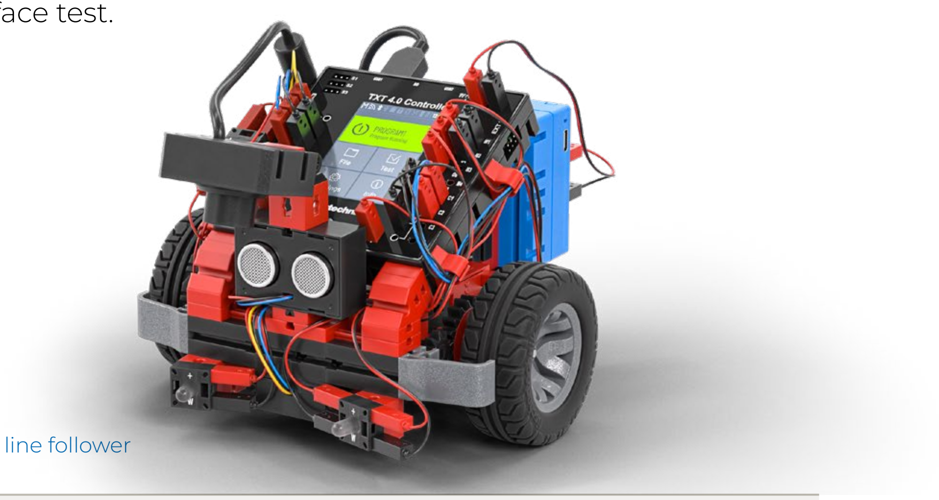
Final/follow-up phase for vehicle training (AGV 1)

Optional: Presentation and allocation of differen-tiations	• Students (or the entire group) eligible for differentiation may be addressed by the teacher. Here, options for avoiding obstacles are discussed. • The integration of the USB camera using different color spaces is introduced.

Construction phase for the digital line follower (AGV 2)

Partner or individual work

- The students use the app to build the digital line follower. The app guides them through the program in short steps.
- The students check the wiring of the sensors using the interface test.



Digital line follower

Programming phase for the digital line follower (AGV 2)

Partner or individual work

- The students sketch a state transition diagram to plan the programming task for obstacle avoidance using the digital line follower.
- The students extend the line-following program from AGV 1 with recognition and avoidance of an obstacle.
- The students plan the integration of the USB camera so that the AGV can react to color surfaces to the right or left of the line.
- The students configure the color recognition in the HSV color space.
- The students define the actions for each color surface and integrate them into the program from AGV 1, so that the AGV now also reacts to recognized color surfaces while line following.
- The app guides them step by step with open questions through the programming task.
- The app offers assistance.
- The program is transferred to the TXT 4.0 controller after each differentiation step.

Experimentation and test phase for the digital line follower (AGV 2)

Partner or group work

- The digital line follower is put into operation.
- The students test the reactions of the AGV to the color surfaces along the course.
- Possible malfunctions in the functional sequence must be found and eliminated.
- Potential troubleshooting is possible based on suggestions in the app.

Final/follow-up phase for the digital line follower (AGV 2)

Optional: Presentation and allocation of further differentiation

- Students (or the entire group) eligible for differentiation may be addressed by the teacher. Here, the concept of the control loop in control engineering is introduced.

Construction phase for the analog line follower (AGV 3)

Partner or group work

- The students use the app to build the analog line follower. The app guides them through the program in short steps.

Analog line follower

Programming phase for the analog line follower (AGV 3)

Partner or group work

- The students plan the integration of the USB camera so that the AGV in this case can follow a line using line recognition with the camera.
- The students configure line recognition with the camera in the RGB color space.
- The students plan and implement a P controller.
- The students extend the programming to a PD controller.
- The app guides them step by step with open questions through the programming task.
- The app offers assistance.
- The program is transferred to the TXT 4.0 controller after each differentiation step.

Experimentation and test phase for the analog line follower (AGV 3)

Partner or group work

- The analog line follower is put into operation.
- Possible malfunctions in the functional sequence must be found and eliminated.
- Possible troubleshooting is possible based on suggestions in the app.
- The students test different settings for the proportionality factor of the P controller.
- The students compare the driving behavior of the AGV when controlled with a P vs. a PD controller.

Final phase for the analog line follower (AGV 3)

Optional: Presentation and allocation of further differentiation

- Students (or the entire group) eligible for differentiation may be addressed by the teacher. Here, the concepts of neural networks and AI are to be presented.

Construction phase of the AI line follower (AGV 4)

Partner or individual work

- The students use the app to build the digital line follower. The app guides them through the program in short steps.
- The students check the wiring of the sensors using the interface test.
- If the digital line follower in AGV 2 has already been built, the construction phase is omitted.

Programming phase of the AI line follower (AGV 4)

Partner or individual work

- The students familiarize themselves with the structure of a simple neural network.
- The students configure a neural network using the app, which calculates motor speeds from the track sensor data and provide training data.
- The students extend the programming from AGV 1 if necessary, so that the motor speeds of the two motors are now determined by the neural network.
- The students train the neural network.
- The students supplement the program with the query of the distance sensor.
- The students extend the neural network with an input neuron that receives the sensor data from the distance sensor.

Experimentation and test phase of the AI line follower (AGV 4)

	Partner or group work	• The students observe the driving behavior of the AGV and test how adjustments to the training data table, different training settings, and different configurations of the neural network affect driving behavior. • The students test the braking behavior of the AGV. • The students optimize the driving and braking behavior of the AGV.

Final phase of the AI line follower (AGV 4)

	Optional: Presentation and allocation of further differentiation	• Particularly efficient students or groups can be given the extension task of developing their own use cases for the driving behavior of the AGV and selecting the necessary sensors for this. • The students can combine the programming knowledge and control techniques acquired during the lessons to develop an AGV with the most consistent driving behavior possible and a wide range of reactions. • Different groups can compete against each other: Which AGV completes the course the fastest? Which AGV is the most stable against disturbances? Independent work without learning cards is intended here.
	Discussion in plenary	• Project debriefing in class. • Clarification of future possible applications in everyday life

METHODOLOGICAL AND INSTRUCTIVE TIPS

Differentiation options

Depending on the duration of the lesson series and the abilities of the students,

- the complexity of the task can be increased by incorporating multiple sensors.
- an extension of the AGV's capabilities through more complex strategies for avoiding obstacles or capturing and reacting to color stimuli from the environment can be implemented.
- the driving behavior of the AGV can be made more consistent through the use of P and PD controllers.
- the motor control for line following or braking can be taken over by a neural network.

Motivational aspects

Working with automated guided vehicles (AGV) ties directly to students' everyday experiences. Already during **vehicle training** they experience how just a few lines of code can reliably set a vehicle in motion – direct feedback that is highly motivating and sparks curiosity.

With the **digital line follower**, the everyday reference is reinforced by parallels to robotic vacuum cleaners, robotic lawnmowers, or driver assistance systems that recognize lines and independently avoid obstacles. The idea that the AGV can now "see" and react like a robot promotes identification with the task and increases interest in practical testing.

The **analog line follower** provides exciting insights through camera use and control engineering into modern vehicle technologies such as lane keeping assist systems. Here it becomes clear how parameter changes or the addition of a derivative component influence driving behavior – a motivating field for experimentation.

With the **AI line follower**, artificial intelligence finally takes center stage: Students experience how training a neural network improves driving behavior. This bridges to current discussions about AI in everyday life and gives students the opportunity to understand and apply future technologies themselves.

BASICS OF CONTROL ENGINEERING

A P controller is the simplest form of **feedback control**: It continuously compares a desired setpoint (target) with the current measured value (actual) and converts the resulting deviation – the error – directly into a correction whose size is **proportional** to the error. The larger the error, the stronger the counteraction; the "sensitivity" of the system is determined by the proportional factor k_P. If k_P is too small, the system reacts sluggishly and returns to course only slowly; if it is too large, it overreacts, begins to oscillate, and becomes unstable. In the classroom context of the analog line follower, this means: The camera measures the lateral deviation from the line, the P controller adjusts the left and right motor speed in opposite directions and brings the vehicle back to the center of the line. Th P controller is thus the basis of any further refinement (e.g. through a D component) and is ideally suited to clearly demonstrate the basic principle of automatic, stabilizing controls with little programming effort.

The central control variable in line following is the lateral deviation of the line center from the image center of the camera. We denote this deviation as error e in **pixels** (positive if the line is to the right of the image center; negative if it is to the left). The controller generates a control variable u from this error, with which the left and right motor speeds are adjusted in opposite directions. With a base speed v_0 and the limitation to the permissible range $[0, v_{max}]$, the result is

$$v_{left} = \text{clip}\,(v_0 + u, 0, v_{max}),\ \ v_{right} = \text{clip}\,(v_0 - u, 0, v_{max})\,.$$

Here, the clip function limits ("saturates") the speed value to the permissible range. If the line is to the right of the center, the AGV should turn right to find the line again. To do this, the right wheel must turn slower and the left wheel faster. If the line is to the left of the center, the AGV must make a left turn, i.e. the left wheel must now turn slower and the right wheel faster.

With the **proportional controller (P controller)**, the control variable is proportional to the current error

$$u = k_P \cdot e,$$

where k_P has the unit "speed per pixel."

Example 1: v_0 = 320, v_{max} = 500, k_P = 1.2, line at +40 px, to the right of the image center, thus e = +40 px. This results in u = 1.20 · 40 = 48, v_{left} = 368, v_{right} = 272 (no clipping necessary).

Example 2: v_0 = 320, v_{max} = 500, k_P = 1.2, line at −35 px, to the left of the image center, thus e = −35 px. This results in u = 1.20 · (−35) = −42, v_{left} = 278, v_{right} = 362 (no clipping necessary).

If a speed exceeds the limits $[0, v_{max}]$, it is limited by the clip function. However, permanent clipping should be avoided by using slightly smaller values for k_P or v_0 .

The **proportional-derivative controller (PD controller)** extends the P controller with a predictive D component, which considers the rate of change of the error. With discretization i formulated, at a sampling time Δt for the control variable $u[i]$

$$u[i] = k_P \cdot e[i] + k_D\,\frac{e[i] - e[i-1]}{\Delta t}\,.$$

Here, k_D has the unit "speed per pixel per second." As the AGV approaches the line center, the D component acts as a brake and reduces overshooting. If the error increases, i.e. the AGV deviates further from the line, the D component reinforces the correction and thus helps return to the line more quickly.

Example 3 (line right, approaching the center):
v_0 = 320, v_{max} = 500, k_P = 1.2, k_D = 0.03 (speed per pixel per second),
Δt = 0.02 s (sampling rate), $e[i]$ = +45 px, $e[i-1]$ = +60 px.
For the P component then $k_P \cdot e$ = 1.20 · 45 = 54.

The rate of change is $(45–60)/0.02 = –750$ px/s. The D component thus provides $0.03 \cdot (–750) = –22.5$. Overall, this results in $u = 54 - 22.5 = 31.5$. Thus, the motor speeds become $v_{left} = 320 + 31.5 = 351.5$, $v_{right} = 320 - 31.5 = 288.5$. Without the D component, a pure P controller would give $u = 54$, i.e. $v_{left} = 374$, $v_{right} = 266$; the PD component visibly damps the correction because the vehicle is already approaching the line center.

Example 4 (line left, error increasing):
$v_0 = 320$, $v_{max} = 500$, $k_P = 1.2$, $k_D = 0.03$ (speed per pixel per second),
$\Delta t = 0.02$ s (sampling rate), $e[i] = –45$ px, $e[i-1] = –20$ px.
For the P component then $k_P \cdot e = 1.20 \cdot (–45) = –54$.

The rate of change is $(–45–(–20))/0.02 = –1250$ px/s. The D component thus provides $0.03 \cdot (–1250) = –37.5$. Overall, this results in $u = -54 - 37.5 = -91.5$. Thus, the motor speeds become $v_{left} = 320 - 91.5 = 228.5$, $v_{right} = 320 - (-91.5) = 411.5$; the PD component significantly reinforces the correction because the vehicle is moving further away from the line center.

If the program is to work for different image widths W, it would make sense to use a normalized error, i.e. $e_n = x/W \in [–1, 1]$.

PROGRAMMING SKILLS

- Program start
- Infinite loop **repeat forever**
- Camera configuration
- Loop **repeat while**
- Command **wait until**
- Condition **if – do**
- Use of variables and their change
- Working with variables
- Working with subprograms
- Programming of neural networks
- Input of AI training data

Optional download:
- Circuit diagram
- Building instructions

ADDITIONAL MATERIALS

- Drawing media (paper, whiteboard, or projection screen).

FUNCTIONS OF THE MODEL AND THEIR TECHNICAL SOLUTIONS

Function of the sensors/actuators	Technical solution
Rotation of encoder motors	Speed adjustment for vehicle control
Mini push button	Obstacle detection **(AGV 1)**
Measurement of brightness differences	Line recognition **(AGV 1–2, AGV 4)**
Measurement of distances	Collision avoidance **(AGV 2, AGV 4)**
AGV 2: Color recognition using USB camera	Reaction to color surfaces
AGV 3: Line recognition using USB camera	Line following using a P controller and a PD controller
AGV 4: Line recognition using track sensor	• Development of a neural network • Input of training data • Implementation of a regression problem using AI
Further differentiation options	Optimization of speed control, optimization of obstacle avoidance strategies, optimization of the strategy for rediscovering the line after complete line loss, design of a custom vehicle with self-selected sensor configuration

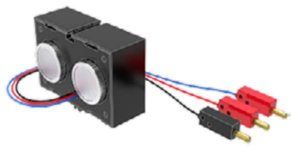
MATERIAL LIST

Sensors	Function
1 On/Off push button on the TXT 4.0 controller	Switching on the AGV
2 mini push buttons	Obstacle detection **(AGV 1)**
1 track sensor (with 2 IR sensors)	Line recognition **(AGV 1–2, AGV 4)**
1 ultrasonic sensor	Distance measurement **(AGV 2, AGV 4)**
1 USB camera	Color recognition **(AGV 2)** Line recognition **(AGV 3)**

Actuators	Function
2 encoder motors	Vehicle drive
2 LEDs (2 × white)	Headlights **(AGV 2–4)**

Where to use what?

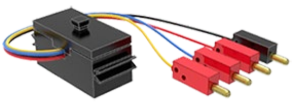
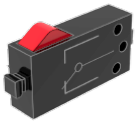
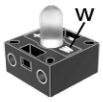
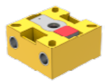
Encoder motor
Item No. 153422



Ultrasonic sensor with ft connectors
Item No. 219835

USB camera
Item No. 152522

		Encoder motor	Ultrasonic sensor with ft connectors	USB camera
1	Analog sensors MODEL 1		1 ×	
2	Syringe pump MODEL 2	1 ×		
3	Gate model MODEL 3	1 ×	1 ×	1 ×
4	Parking garage 1: Barrier MODEL 4	1 ×		
5	Parking garage 2: Entry and exit 2 × MODEL 4	2 ×		
6	Parking garage 3: License plate recognition MODEL 5	1 ×		1 ×
7	Traffic light control 1: Traffic light MODEL 6			
8	Traffic light control 2: Intersection 2 × MODEL 6			
9	Traffic light control 3: AI phase control MODEL 7			1 ×
10	AGV 1: Vehicle training MODEL 8	2 ×		
11	AGV 2: Digital line follower MODEL 9	2 ×	1 ×	1 ×
12	AGV 3: Analog line follower MODEL 10	2 ×		1 ×
13	AGV 4: AI line follower MODEL 9	2 ×	1 ×	1 ×

 IR track sensor with ft connectors Item No. 219831	 Mini push button Item No. 037783	 LED Item No. 162134	 Light barrier LED Item No. 162135	 Phototransistor Item No. 036134	 NTC resistor Item No. 219830	
					1 ×	1
	2 ×	2 ×				2
	2 ×	2 ×	1 ×	1 ×		3
	2 ×	2 ×	1 ×	1 ×		4
	4 ×	4 ×	2 ×	2 ×		5
	2 ×	2 ×	1 ×	1 ×		6
	1 ×	4 ×	1 ×			7
	2 ×	8 ×	2 ×			8
	1 ×	4 ×	1 ×			9
1 ×	2 ×					10
1 ×		2 ×				11
		2 ×				12
1 ×		2 ×				13