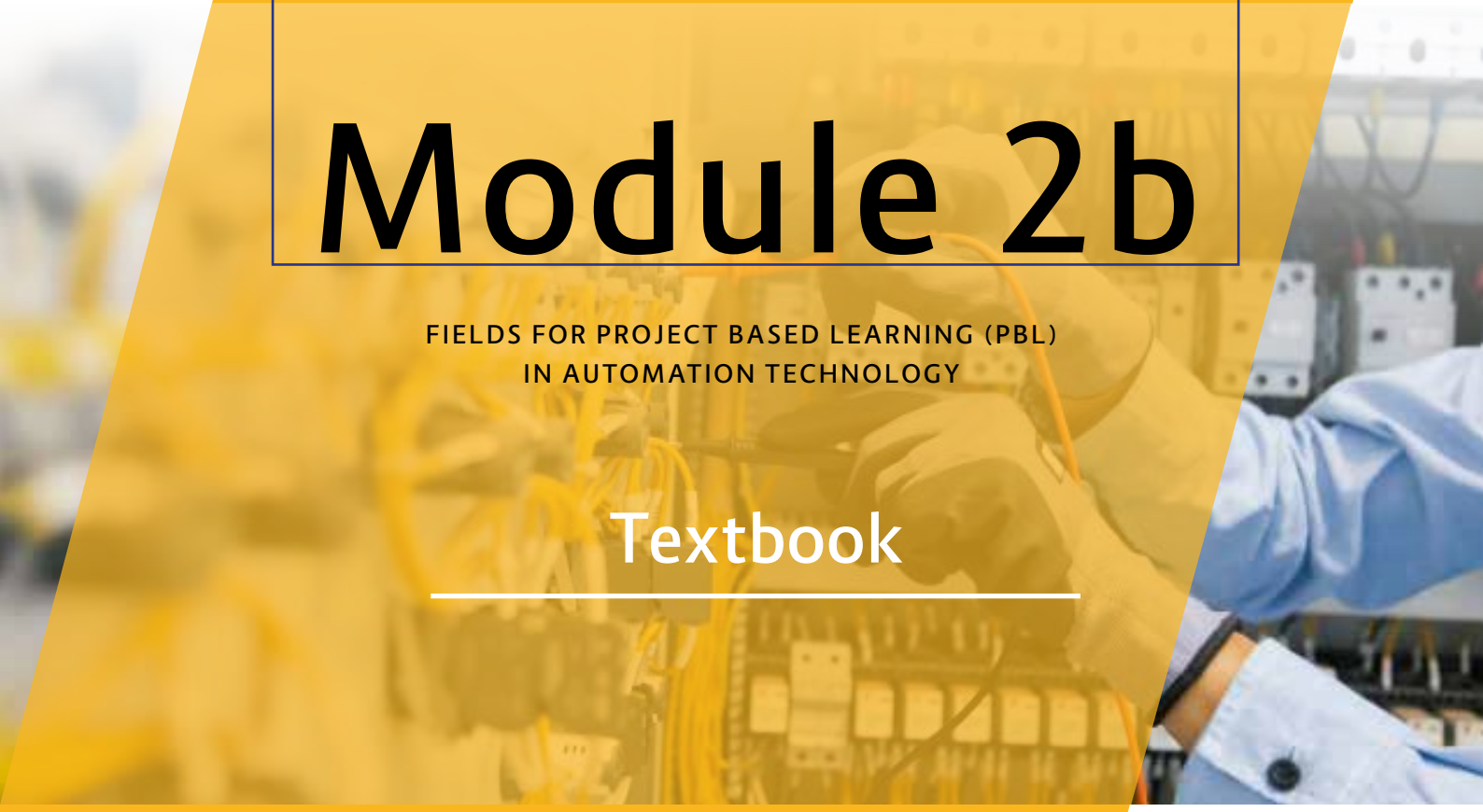
ÜBZO - INTER-COMPANY TRAINING CENTER IN EASTERN BAVARIA 2024

Module 2b

FIELDS FOR PROJECT BASED LEARNING (PBL)
IN AUTOMATION TECHNOLOGY

Textbook



Education is future

MODULAR TRAINING & EDUCATION

IN MECHANICAL
& ELECTRICAL ENGINEERING

TRAINME 2
Modular training and further education of South African TVET lecturers in mechanical and electrical engineering

Editing

Author:	Dipl.-Ing. (Univ.) Helmut Windschiegl
Illustrations, Layout:	Dipl.-Ing. (Univ.) Helmut Windschiegl
Cover Photos:	Carina Adam, Helmut Windschiegl
Layout Assistance:	Sebastian Gmeiner / Melanie Maeller
Editor:	Carina Adam
Version:	September 2024

This textbook has been developed within the framework of the TRAINME & TRAINME 2 programme (Modular Training of South African TVET Lecturers in Mechanical and Electrical Engineering). The TRAINME programmes have been designed by the Inter-Company Training Centre in Eastern Bavaria (ÜBZO) and the University of Stuttgart (Dept. BPT) on behalf of the German Federal Ministry of Education and Research (BMBF) and the DLR Project Management Agency (DLR-PT) in cooperation with the South African Department of Higher Education (DHET).

Powered by:

ÜBZO Überbetriebliches Bildungszentrum in Ostbayern gemeinnützige GmbH

University of Stuttgart Germany

Supported by:

higher education & training Department: Higher Education and Training REPUBLIC OF SOUTH AFRICA

SPONSORED BY THE Federal Ministry of Education and Research

Preamble

The field of industrial automation technology is currently developing and changing at a breathtaking pace. It is becoming increasingly important for users and developers to understand automated systems as a whole and to be able to analyse them on the basis of fundamental knowledge.

Take your students into the new world of automation, teaching technologies of the present and the future, building on a solid understanding of basic enabling technologies of sensors, actuators and networking.

This studybook is intended to help bridge the gap between fundamental technical contexts to Industry 4.0 systems. Internet of Things, edge computing, cloud-based and networked systems, and the use of artificial intelligence are just a few buzzwords that will take us directly into the future in this context.

The TRAINME and TRAINME 2 projects deliberately focuses on content and technologies that are easily accessible and can be implemented in schools without major financial outlay. For both teachers and students, the introduction to the Arduino development environment opens up a global maker scene with no shortage of projects and project ideas to replicate and improve.

This very versatile microcontroller board is available in many versions and has now also found its way into the world of small industrial controllers with the development and marketing of the Controllino. In addition to the fact that the boards, the sensors, the actuators and all other materials are relatively inexpensive to purchase, it is particularly positive that the development environment (IDE) and numerous software libraries, programming examples and tutorials are freely available on the Internet and can be downloaded free of charge.

Another advantage is that there is a myriad of Internet sites where other users offer advice and valuable tips on how to implement complex projects of their own. It is therefore relatively easy to gain in-depth knowledge and skills in the use of different sensors and the control of actuators based on them. In independent small projects it is possible to understand and reproduce more and more complex applications of control and regulation systems. From a sufficient depth of knowledge, you will even be able to use wireless connection technologies such as Wifi and Bluetooth for component communication.

The fact that a simplified variant of the C++ programming language is used for programming can be seen as a particularly valuable side effect. This generally applicable and easy-to-learn programming environment forms a solid basis for other programming languages such as Python, which is also very widely used for programming complex data structures and in the use of artificial intelligence.

Get involved in a particularly practical and motivating way of teaching important and future-oriented learning content.

Content

Unit 1: Project based lessons

Unit 2: Basic knowledge of how IoT works

Unit 3: Basic knowledge of sensor technology

Chapter 1

Project based lessons

In order to better understand the approach to technical projects in the classroom, this Study Book starts with basics on the theory of project-based learning.

1.1 NEW REQUIREMENTS IN THE PROFESSIONAL WORLD

1.1.1 How does the modern world of work change the way we learn?

The **"Old-School-Model"** of learning facts and recalling them out of context is no longer sufficient to prepare students to survive in the 21st century.

Solving highly complex problems requires that learners have both

Fundamental Skills

- reading,
- writing and
- mathematics

and

21st Century Skills

- teamwork,
- problem solving,
- information capture,
- time management,
- information synthesising,
- utilising high-tech tools.

With this combination of skills, students become managers of their own learning process, guided and mentored by a skilled teacher.

Other skills that are also included:

- personal and social responsibility
- planning, critical thinking, reasoning, and creativity
- strong communication skills (needed for interpersonal discussion and presentation)
- cross-cultural understanding
- visualising and decision making
- know how and when to use which technology
- choosing the most appropriate tool for the task

1.1.2 Need for change - What does this mean for vocational schools?

The development in the industry and working environment favours the dismantling of frontal teaching. In the professional world, more and more group work is being introduced.

The world of work is moving towards group work in the form of:

- quality circles
- project and working groups,
- production teams and
- semi-autonomous working groups.

Students at vocational schools must therefore be made fit for work, in view of developments in industry.

1.1.3 How does the learning process work?

In order to understand the importance of a necessary change towards activity orientation, it is essential to see learning from a global perspective.

Learning from a holistic point of view ...

- is a cognitive process in which knowledge structures are built up.
- leads to knowledge and competences for future activities.
- Learning can basically be understood as intentional behaviour, i.e. it pursues a specific goal.

This intentional behaviour is characterised by a goal or purpose that is based on one or more motives. Accordingly, a person is considered motivated in his or her behaviour if he or she wants to achieve a certain goal through his or her activity or is generally pursuing a certain purpose with it.

1.1.4 Possible solution: Activity-based learning?

As early as the turn of the millennium, the Conference of Ministers of Education and Cultural Affairs in Germany recognised the need for change in vocational education and training:

> *"The objective of vocational education and training requires that teaching be based on a pedagogy tailored to the tasks of the vocational school, which emphasises action orientation and enables young people to independently plan, carry out and assess work tasks within the framework of their occupational activity (...)"*

Source: KMK (1996/2000). Sekretariat der Ständigen Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland (Hrsg.), Handreichungen für die Erarbeitung von Rahmenlehrplänen der Kultusministerkonferenz (KMK) für den berufsbezogenen Unterricht in der Berufsschule und ihre Abstimmung mit Ausbildungsordnungen des Bundes für anerkannte Ausbildungsberufe (Version of September 15, 2000)

Based on learning theory and didactic findings, **"orientation points"** are named, such as:

- professional situations,
- actions should be carried out by oneself,
- plan to assess, if necessary supplement with correcting and evaluating,
- holistic grasp of professional reality,
- integration into the learner's experiences and
- establish references to social processes.

1.1.5 The South African view

The leaders of education in South Africa have also been thinking about the need for a change in the way learning takes place for quite some time:

> *"Synthesising descriptions in the White Paper on Education and Training (1995) and the White Paper for Post-school Education and Training (2014), the key principles underpinning open learning are:*
>
> *i. students are provided with opportunities and capacity for lifelong learning;*
>
> *ii. learning processes centre on the students and contexts of learning, build on their experience and encourage active engagement leading to independent and critical thinking;*
>
> *iii. learning provision is flexible, allowing learners to increasingly determine where, when, what and how they learn, as well as the pace at which they will learn;*

> *iv. students have access to learning opportunities and unnecessary barriers to access have been removed;*
> *v. prior learning and experience is recognised wherever possible;*
> *vi. arrangements for credit transfer and articulation between qualifications facilitate further learning;*
> *vii. providers create the conditions for a fair chance of learner success through learner support, contextually appropriate resources and sound pedagogical practices; and*
> *viii. provision of open learning is working within legacy systems. This means that open learning will be embedded in, as well as enhance and expand, the current education and training system, and not be an add-on to the current system."*

Reference: Department of Higher Education and Training's Position on Online Programme and Course Offerings, 22 March 2017, Page 18f

Most of these demands can easily be met by at least partially introducing **activity-based or problem-based teaching.**

1.2 CLASSIFICATION AND ASPECTS OF ACTIVITY-BASED TEACHING

1.2.1 Concept of constructivist didactics

The concept of activity-oriented teaching essentially belongs to constructivist didactics (see also Study Book Module 1 – pedagogy and didactics).

This is based on the fact ...
1. that people can only perceive external reality with the help of their sense organs, and
2. that every person constructs his or her idea of reality from the perceptions in the course of his or her life.

→ **This leads to the conclusion that the respective reality must be individual.**

Particularly relevant for activity-oriented didactics is the conclusion that it cannot be a meaningful goal to anchor a certain idea of "reality" in another person, as meanwhile outdated forms of instruction attempt to do.

Instead, the learner needs to be supported in the construction of their imagination by providing opportunities to test their own ideas against reality.

1.2.2 What is activity-based teaching?

Activity-based teaching is holistically demanding, student-active teaching in which the activity products agreed between the teacher and the students determine the organisation of the teaching process. Action products are the results of action-oriented teaching.

Activity-based teaching follows the principle of the **inductive approach** and
1. usually takes a concrete situation as a starting point for the exercise,
2. in order to derive a general regularity from it or to explain a general principle.

The counter concept is the **"Old-School-Model"** of instructional learning based on the deductive approach.
1. It first explains the principle, the regularity or the connection,
2. to then deal with individual cases in the form of exercises or examples.

1.2.3 The core of a constructivist teaching model

Constructivism is a theory of learning that states that learners construct their knowledge based on their experiences and prior knowledge. Constructivism emphasises the importance of active learning and the role of the teacher as a guide in the learning process.

The accumulation of experiences requires self-activity.

Constructivism is a theory of learning that states that learners construct their knowledge based on their experiences and prior knowledge. Constructivism emphasises the importance of active learning and the role of the teacher as a guide in the learning process.

The didactic goal that the **construction model of learning** is meant to realise, is how to create a learning environment in which students
- take learning initiative,
- gain experience,
- learn to articulate themselves and
- can reflect the entire learning process on their own work and on themselves.

1.2.4 New role of the student

In activity-based teaching, the role of the student is to

- ask questions,
- build knowledge and
- find a real solution to the problem/question posed.

Students need to collaborate, extend their active listening and demand intelligent, focused communication so that they can think rationally about solving problems. Activity-based teaching forces students to take responsibility for their success.

1.2.5 New role of the lecturer

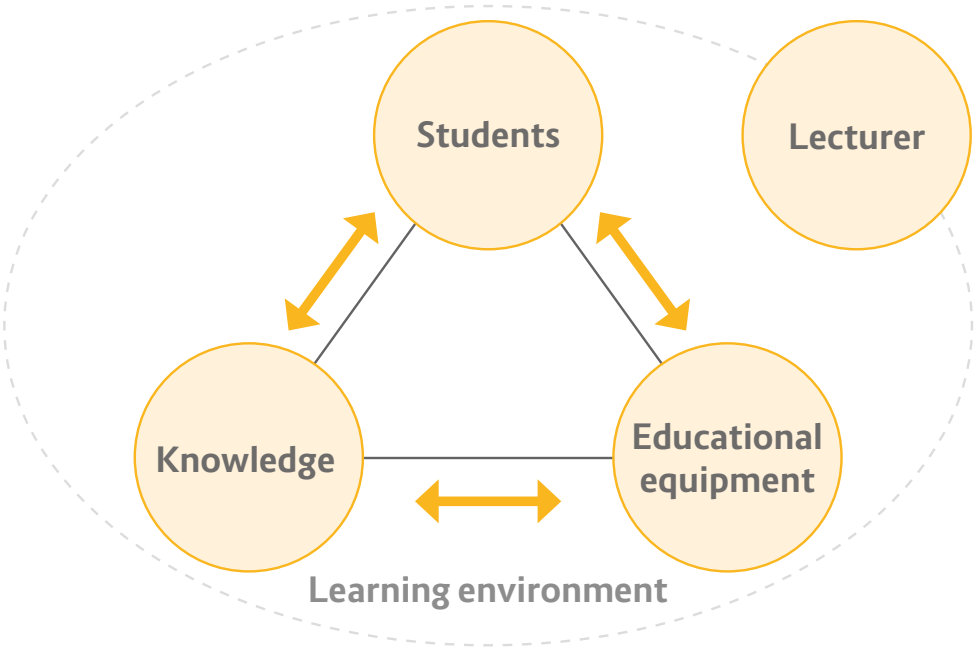
The role of the lecturer in activity-based learning and teaching is that of a **facilitator**. You do not surrender control over the lesson or the students learning but develop an atmosphere of shared responsibility.

The lecturer must structure the proposed question/topic in such a way that student learning is directed towards content-based material. Lecturers control student success with temporary, transitional goals to ensure that student projects remain focused, and students have a deep understanding of the concepts being studied. Students are held accountable to these goals through continuous feedback and assessments.

Ongoing assessment and feedback are essential to ensure that the student stays within the scope of the questions and standards that the project is trying to extract.

Task of the lecturer	• provides structural commitment, • offers content-related support and • creates mental freedom
Requirements for the lecturer	• transition from lecturer to educational consultant • balance between structural commitment and openness in terms of content • organisation of cooperative and participatory learning • lecturer role becomes more “scientific” • transparent performance consideration
Lecturer activities in the project	• learning facilitator • educational consultant • learning observer • learning organiser • learning leveller

1.2.6 "Learning triangle" of activity-based teaching



The lecturer has to initiate the "learning triangle" and control the impact.

1.2.7 Professional expertise

When exploring the subject depth of the lessons, it is essential to pay attention to the following framework conditions:

- Work with the real horizon of your students!
- Projects expose gaps more relentlessly than regular lessons!
- Students are not good at fitting into projects!
- Projects require initiative!
- Projects show the difference between knowledge and competence!

1.2.8 General approaches

- Use the changed role of the lecturer!
- Use checklists!
- Monitor performance transparently!
- Reviewer rounds: Students take over the reflection of the contents!
- Use methods for gaining knowledge more extensively!
- Put new ideas in the foreground and push back standard solutions!

- As a lecturer, act more as a cue-giver!
- Emphasise the students‘ responsibility for their own work!
- Ask the “research question“ correctly!

1.2.9 Research question / aim instead of topic

- Questions structure the project work!
  (“From complex questions to solvable individual tasks“.)
- When approaching the task, always prefer the top-down design!
  (From “Big“ to “Small“.)
- The entire project work serves only to answer the research question comprehensively!
- The quality of the project work is measured by how comprehensively
  the research question is clarified!

> *“Teaching can only succeed if it leads the student to see a problem as his or her own!“*

John Dewey; American philosopher and educator; 1859-1952

1.2.10 Importance of evaluation

Even in project-based lessons, it is important for the students to give them feedback on their performance in the project work. The biggest problem is that in team-building lessons, individual assessments contradict the basic idea of cooperation.

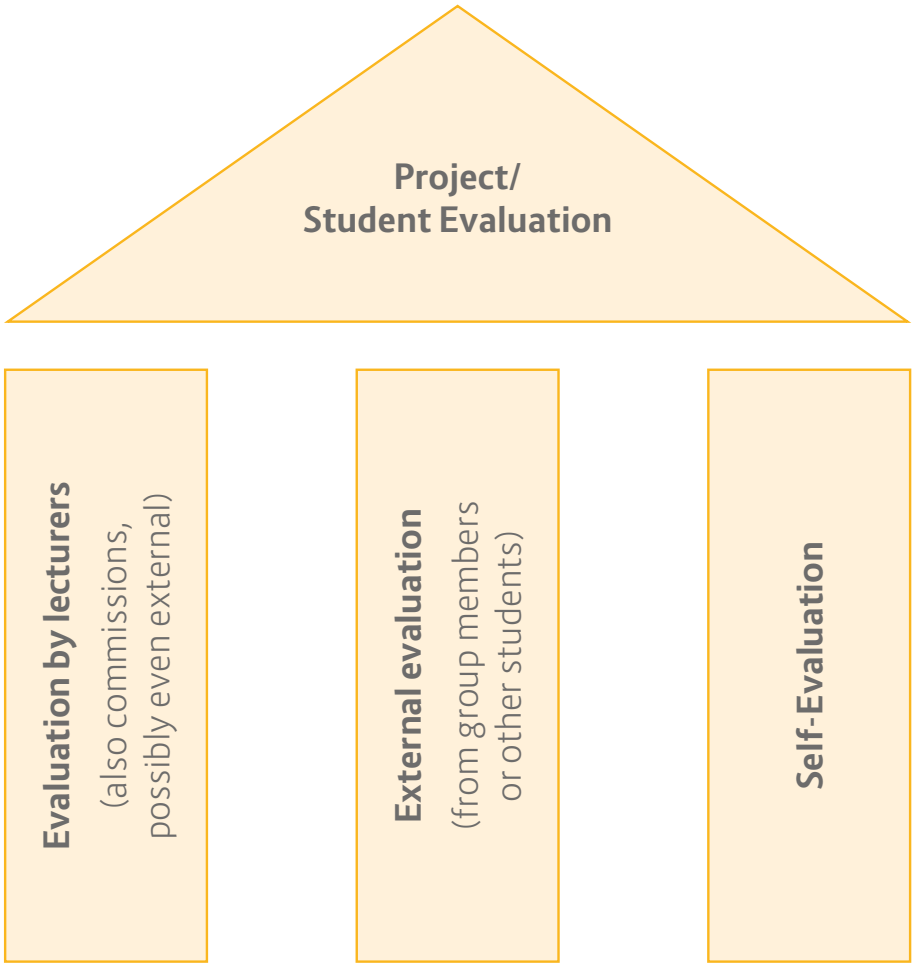
How can we succeed in assessing the performance of the students in a transparent, comprehensible, and thus as fair as possible way?

Evaluation briefly:

- “Individual assessments“ contradict the idea of projects in teams.
- Evaluation requires comparisons
- Transparency: Criteria are fixed in advance
- Evaluation is all the better the broader and more diverse the senders of feedback are.
- Evaluation should be more “formative“ instead of “summative“.
- Problem: From group to individual assessment

Who should evaluate?

The evaluation should be based on at least three pillars.



In the next chapter, the specifics of **Project-Based Learning** will be discussed in more detail.

1.3 INTRODUCTION PROJECT-BASED LEARNING (PBL)

1.3.1 What is "Project-Based Learning"?

Project-based learning is a very practice-oriented learning method in which the students learn independently and action-oriented in realistic situations. The focus is on planned action and a structured project process. Project-based learning is mainly used in classical face-to-face teaching.

Project-based learning (PBL) is a teaching method in which students acquire knowledge and skills by investigating and responding to authentic, exciting, and complex questions, problems, or challenges over an extended period of time.

Dr Thom Markham describes **Project Based Learning (PBL)** like this:

> *"PBL integrates knowledge and action. Students learn knowledge and elements of the core curriculum, but also apply their knowledge to solve authentic problems and achieve outcomes that matter. PBL refocuses education on the student rather than the curriculum - a shift mandated by the global world that rewards intangible assets such as drive, passion, creativity, empathy and resilience cannot be taught from a textbook but must be activated through experience."*

https://www.pblglobal.com/seven-steps-to-high-quality-pbl/ (last accessed on 31.08.2023)

1.3.2 What effects does PBL have on students?

The elements promoted in PBL include basic knowledge and understanding of content and skills.

Students are expected to engage in
- an extended process of asking questions,
- searching for information and
- applying the information

to find their chosen solution.

(Picture credit: C. Adam, TRAINME 2019)

Fundamental to PBL is that students make choices, both about the solutions to the problem and about how they work and what they create. Reflection on learning and revision of process and methodology are essential. Students are expected to question their own effectiveness and that of others, the quality of their work, how they overcame obstacles and what could be improved next time.

1.3.3 What effects does PBL have on students?

PBL aims to develop critical and creative students who can take ownership of their learning. The two processes, both of which promote metacognitive thinking (the ability to reflect on our thinking) and sophisticated, multidimensional problem solving, aim to develop young people who are responsible, aware, and engaged in their world.

View of an unknown teacher:

> *"One of the most important aspects of project-based learning are real-world problems and their solutions. Many of our students are not confronted with real life problems, they only deal with them when they leave college.*
> *By exposing them to project-based learning, they are empowered to deal with these problems. It also makes them realise that their college education is not an academic exercise, but a preparation for the real world."*

1.3.4 What questions do lecturers need to ask themselves?

- What are the benefits of project-based work?
- Which topics do projects make sense for?
- How can projects be anchored in or at the classroom in such a way that the quality of school education can benefit from them?
- How can professional quality be guaranteed?
- How can structural commitment be combined with content-related support in a way that makes the work successful?
- What are the ways in which students can gain knowledge on their own?
- How can reservations and fears about project-oriented teaching be overcome?

And very important:

- How can projects be evaluated?

1.4 THE RIGHT WAY TO SUCCESSFUL PROJECT LESSONS

1.4.1 Steps to successful project teaching

If you want to carry out a project work in school, you should have a clear stage plan in mind and communicate it to the students. The following seven steps will help you to structure the project work and give you a sense of direction:

Project initiative
Every project begins with an initial impulse. This can be an idea, a wish, a problem or a task. It can be brought in by the students as well as the lecturers.

Project draft
From the various proposals, one is selected and subsequently refined. At the end of this step, there should be a first project draft developed by the students.

Project schedule
The draft becomes a work plan. The project goal and a time frame are set and the tasks are distributed.

Execution
The students implement the plan. They take care of their tasks independently and on their own responsibility.

Check
The team sits down again and again and checks whether and how the agreed goals have been achieved.

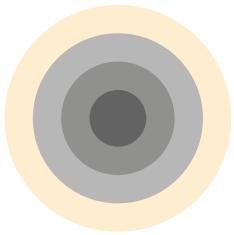
Conclusion and presentation
The project is brought to an end and the results are presented by the project participants.

Reflection
The students review the project, agree on what went well, what went wrong and work through the conflicts that happened during the project.

1.4.2 Special aspect of technical projects

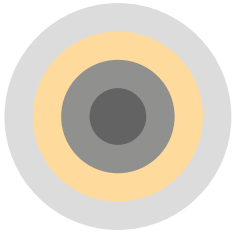
In the case of technical projects, it is more useful for the lecturer to prescribe the steps, as the students are usually not in a position to judge the level of difficulty of the task. However, the idea for the project may well come from the students.

The different **"levels of knowledge"** should be considered as an orientation for approaching a scientific or technical project.



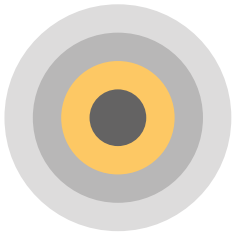
Shell 1: Orientation and overview knowledge

e.g. rough overview, structuring, functional unit level, consideration of the overall system



Shell 2: Context knowledge

e.g. interaction of subsystems, part-systems, and their functions



Shell 3: Detail and functional knowledge

e.g. tasks of the individual components of the various functional units

Shell 4: Subject-specific and in-depth knowledge

e.g. detailed considerations, complex interrelationships, component level, physics, mode of action of components, calculations

The content and relevance of the individual levels depend very much on the chosen topic and can therefore vary.

1.4.3 Project Based Learning in Electrical and Automation Engineering

The use of projects is particularly interesting for learning content from electrical engineering, automation technology, information technology and construction technology. A project-based learning approach encourages students to think about how technical elements work and leads to the development of problem-solving skills and critical thinking.

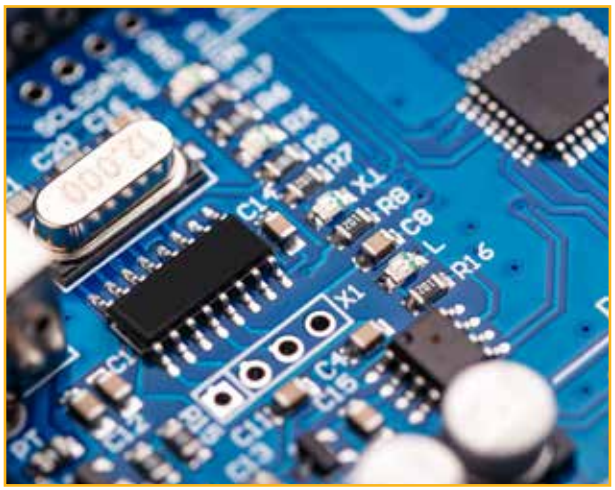
> Especially in the areas of electrical engineering, electronics and programming, students develop a more comprehensive understanding of electronic circuits, their function, and the associated programming more quickly with suitable learning material for the classroom. The focus is on gaining a deeper understanding of the interaction of signals and components in a learning environment with industry-related material.

This approach leads to a higher readiness among the students, who are also better prepared for today's Industry 4.0 needs.

1.4.4 Microelectronics

Microelectronics plays an increasingly important role in the increasingly modern world. Microelectronics are contained in (almost) all electronic devices:

For example in smartphones, monitors, calculators, WLAN radios, Bluetooth speakers, cars, electronic kitchens, body scales or displays showing train or bus delays.



Grafvision - stock.adobe.com

There are microprocessors in computers, machines and smartphones that can perform all kinds of calculations. For concrete applications, there are also flexible microcontrollers and special application-specific integrated circuits.

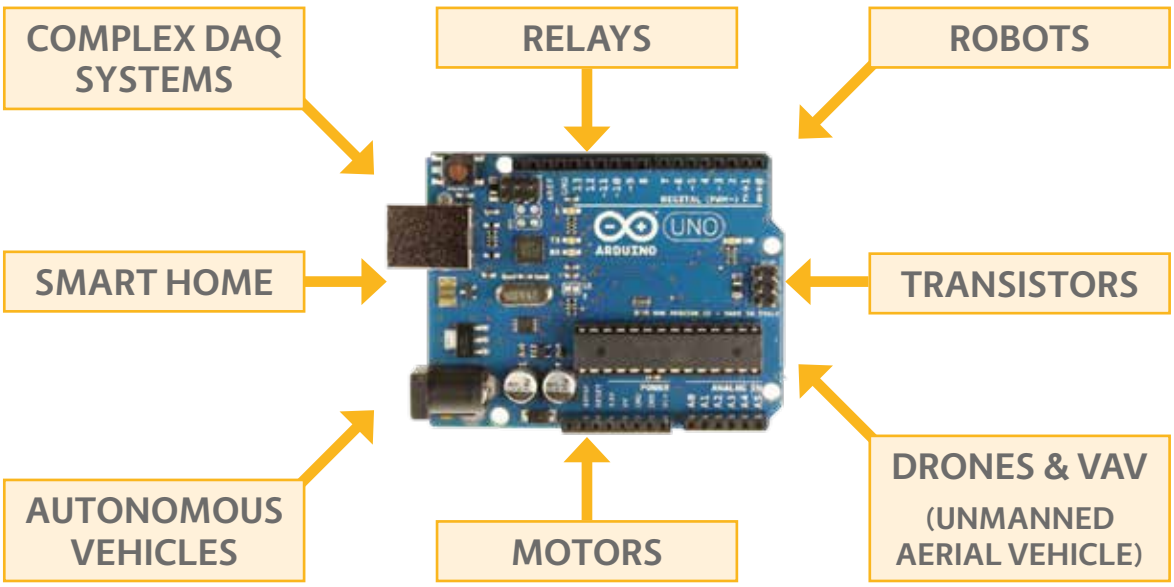
All the building blocks of processors and controllers are combined on microchips. The Arduino microcontroller will serve here as a globally widespread, inexpensive and therefore ideally suited example for learning content in the field of automation and microelectronics.

1.5 EXAMPLE OF A PROJECT-BASED LEARNING SEQUENCE

1.5.1 Programming with the Arduino

The Arduino is a microcontroller and open-source project by Massimo Banzai and David Cuartielles from 2005. Hardware and software are freely available on the internet under a Creative Commons licence. A CC (Common Creative) licence allows creators to make their creative works available to the public while reserving certain rights of use.

A microcontroller consists of a processor and various peripheral elements. In many household appliances or machines, microcontrollers are used for numerous tasks. They can be found in washing machines, televisions, remote controls, printers, but also in vehicles to control window regulators, airbags or air conditioning systems and other sensory components.



The following teaching sequence is based on a passenger car, whose functions are enhanced by sensors and programmes, for example automatic low beam, a parking sensor and automatic distance control.

1.5.2 Competences taught

Technical competence

The students ...

- ... get to know basic electrotechnical connections.
- ... get to know the functions of simple electronic circuits.
- ... get to know basic programming techniques.
- ... get to know how a microcontroller works.

Media competence

The students ...

- ... use digital tools and electronic components to solve everyday tasks.
- ... recognise the structure of algorithms and modify them to solve tasks.
- ... independently search for solutions and required information on the Internet.

Social competence

The students ...

- ... work in a team to solve tasks.
- ... learn in a self-organised way with prepared material.
- ... support each other across teams in solving tasks, if necessary.

1.5.3 Didactic-methodical classification

Previous knowledge

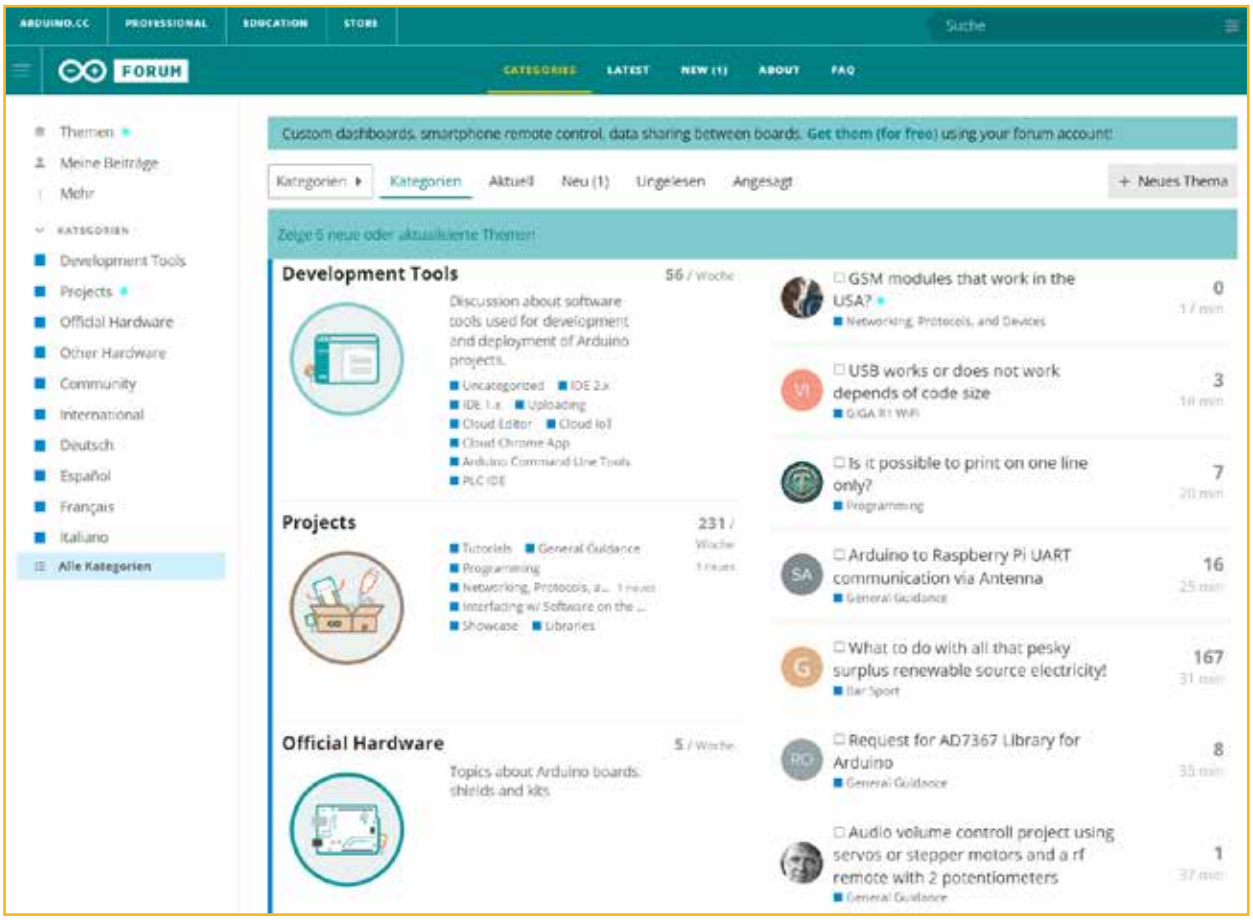
Students do not need any previous knowledge of programming. However, they should have basic computer skills.

Didactic analysis

Both the topic of programming and the topic of electronics are generally relevant for all students because digitalisation is bringing more and more automation and **Artificial Intelligence (AI)** into people's everyday lives. In view of their future, today's students should know basic contexts of programming in order to grasp the possibilities but also the limitations of software.

Nowadays, programming is rarely done on a blank piece of paper. Code is downloaded from the Internet, analysed, reused and modified. Globally organised communities and forums on the Internet serve as sources.

It is important to have a complete understanding of reused programme code in order to use it in your own projects.

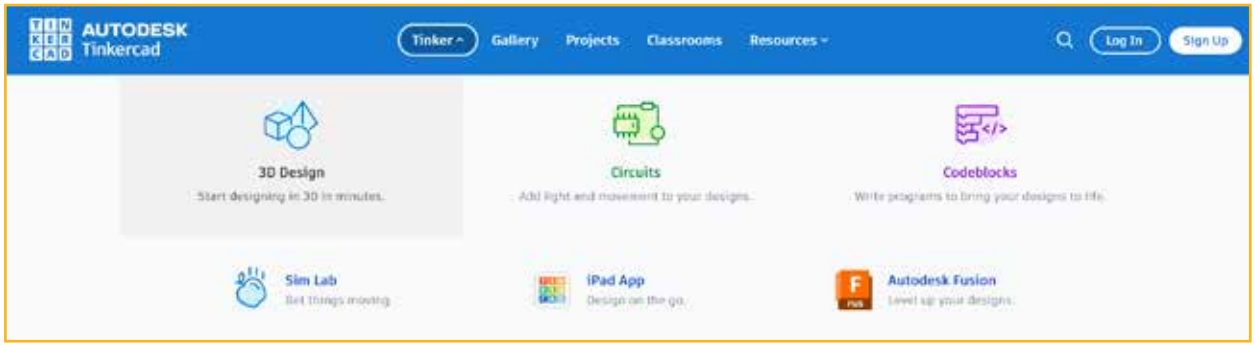


It is also essential that lecturers encourage their students to do the following:

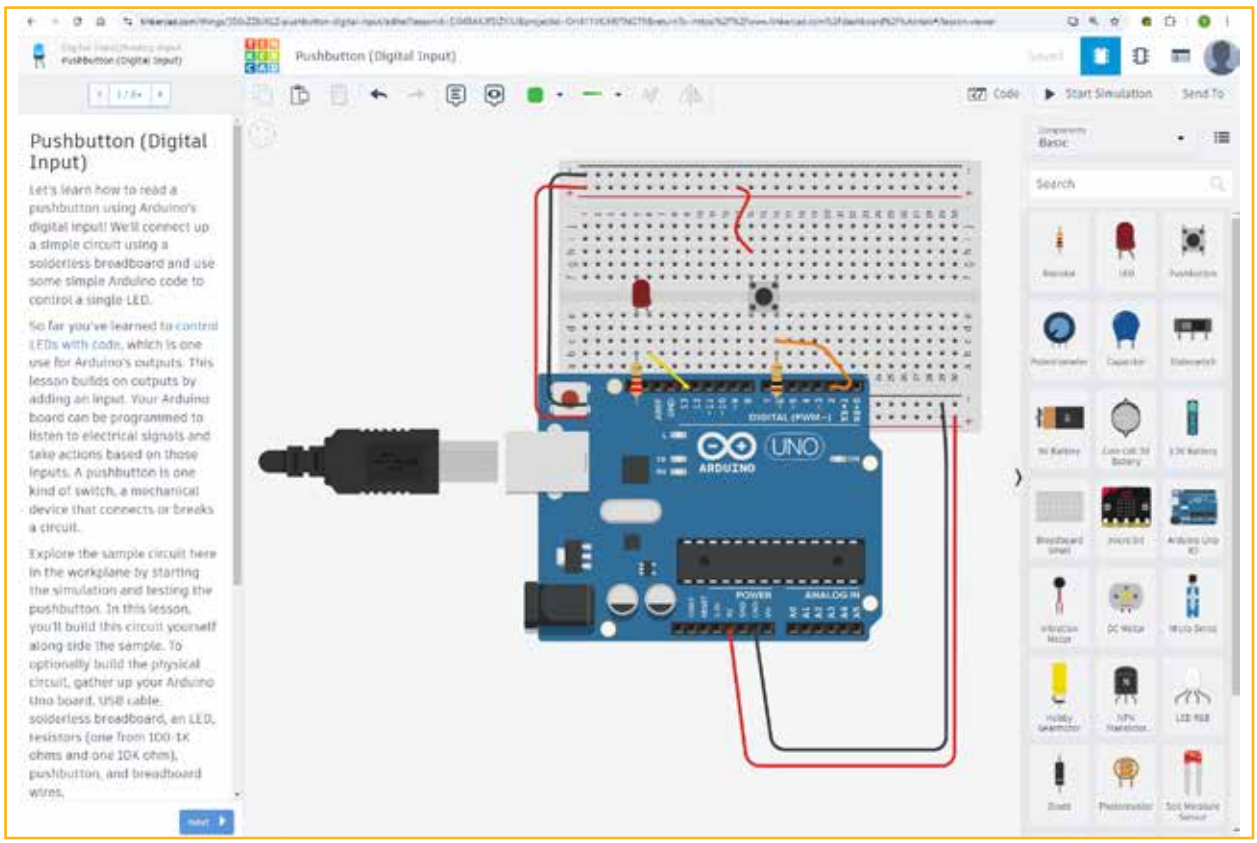
A detailed commentary on the programme code can help to ensure that learners have the necessary understanding.

Methodical analysis

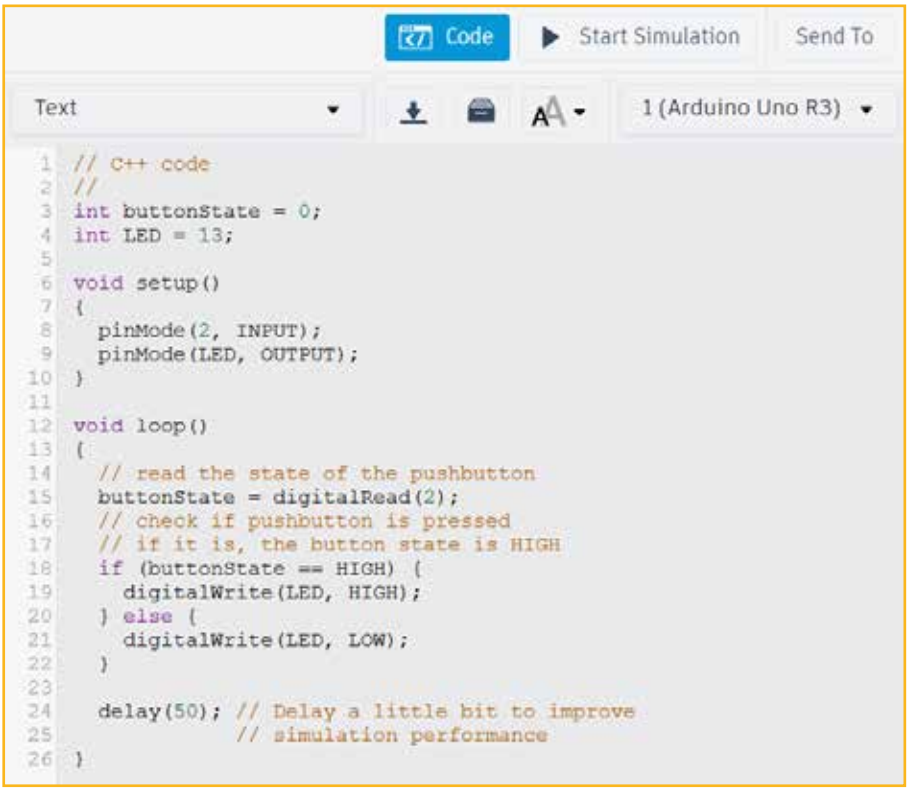
In principle, it is possible to work on the learning units via the learning platform Tinkercad (www.tinkercad.com).

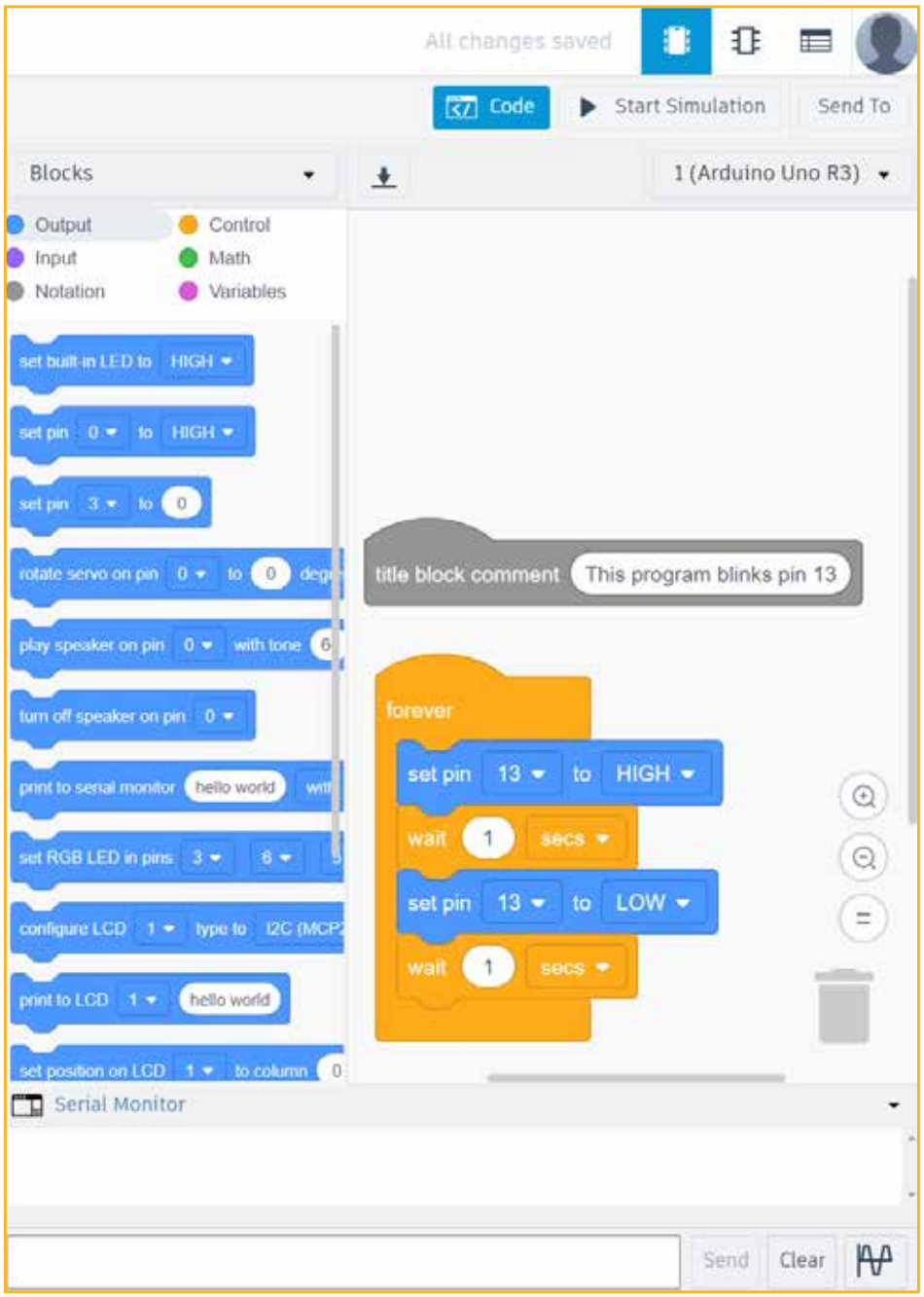


In this virtual environment, it is possible to assemble virtual components into circuits and run the programme code on these **“digital twins“**. A **“digital twin”** is a digital replica of a physical object or system that collects and simulates data in real time to enable insight, analysis and optimisation.



The programme code can be written as normal code (similar to C++) or edited in block form.





Furthermore, it is possible to work on the tasks with real components and microcontrollers because both programming code and hardware are mapped identically in the simulation. Corresponding hardware sets as well as additional hardware are available at low cost.

For good students with previous knowledge, each lesson can include a “professional task“.

1.5.4 Possible lesson plan

	Content	Interaction method	Time
Teaching Unit 1: Introduction (First steps)	Introduction to the topic of microcontroller programming; first small project "flashing light"	lecturer presentation, individual work/ partner work on the computer with Tinkercad or on the model	4 to 8 hours
Teaching Unit 2: Light	Project with buttons to switch the light on and off (Pull-down resistor)	lecturer presentation, individual work/ partner work on the computer with Tinkercad or on the model	4 to 8 hours
Teaching Unit 3: Low beam light	Photo resistor as sensor: Automatic switching between high and low beam in a car	lecturer presentation, individual work/ partner work on the computer with Tinkercad or on the model	4 to 8 hours
Teaching Unit 4: Parking sensor	Project for programming a parking aid with ultrasonic distance measurement	lecturer presentation, individual work/ partner work on the computer with Tinkercad or on the model	4 to 8 hours
Teaching Unit 5: LCD display	Project: LCD display to show the measured distance	lecturer presentation, individual work/ partner work on the computer with Tinkercad or on the model	4 to 8 hours
Teaching Unit 6: Adaptive cruise control	Project: Adaptive Cruise Control (ACC) (Programming a "for-loop")	lecturer presentation, individual work/ partner work on the computer with Tinkercad or on the model	4 to 8 hours
Teaching Unit 7: Switch Case	Project: Automatic instrument backlighting depending on ambient brightness ("switch case")	lecturer presentation, individual work/ partner work on the computer with Tinkercad or on the model	4 to 8 hours

In the following section, the principle of the teaching unit will be briefly explained with the help of examples. Unit 1, Unit 2 and Unit 4 will be examined in detail as examples. The central element for controlling the lesson is a work sheet for the students.

Teaching Unit 1: Introduction (First steps)

After an introduction to the theoretical basics of programming a microcontroller, the first applications will already be implemented.

The first task is the realisation of a flashing light with the help of an Arduino and a light-emitting diode.

The following worksheets (written in blue colour) are available for download here:

https://drive.google.com/drive/folders/1lCH-JEmothzEdg8xiDfOAUL5eF7Dp6aW3

The worksheet is also available as a Word document called "01_Worksheet.docx" and can be adapted to suit your needs.

Example Worksheet 1:
"Flashing of an LED with digital output"

Task: Blink an LED with digital output

Learning Objectives:

- Get to know the simulation tool Tinkercad.
- Use of a microcontroller output.
- Basics about the function of an LED and the necessity of contacting via a series resistor.
- Writing a simple code with blocks.
- What the corresponding code looks like
- Understanding the **delay()** command.
- Repeat on the physical Arduino board.

Required equipment:

Tinkercad account
Arduino board / breadboard / cables / LEDs / resistor 220 kΩ

Short introduction:

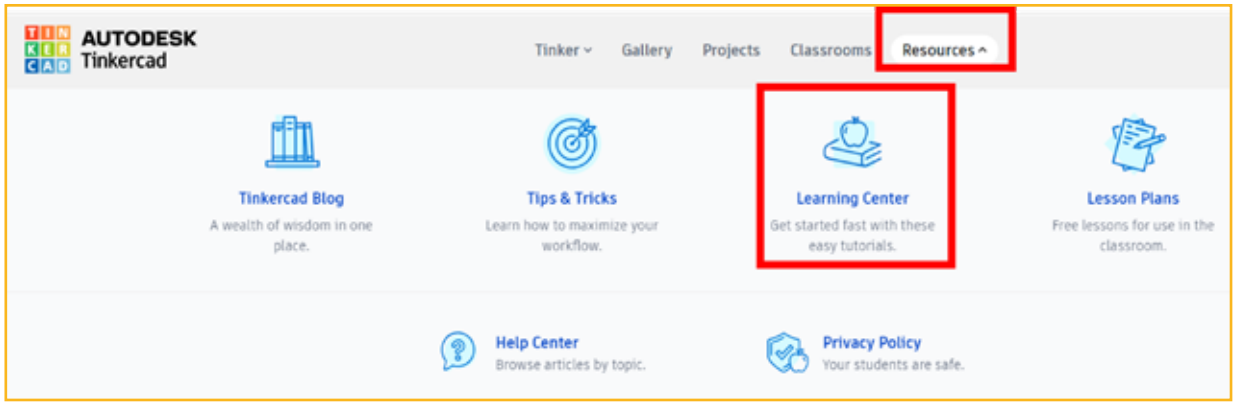
Get to know the functionality of the simulation tool **Tinkercad** with the help of the tutorial **"Blink an LED with digital output"**.

Use the explanations to learn the basics of correctly connecting LEDs and take your first steps in microcontroller programming. Compare the different approaches to **"block programming"** and **"code programming"**. Learn the first simple commands for programming microcontrollers.

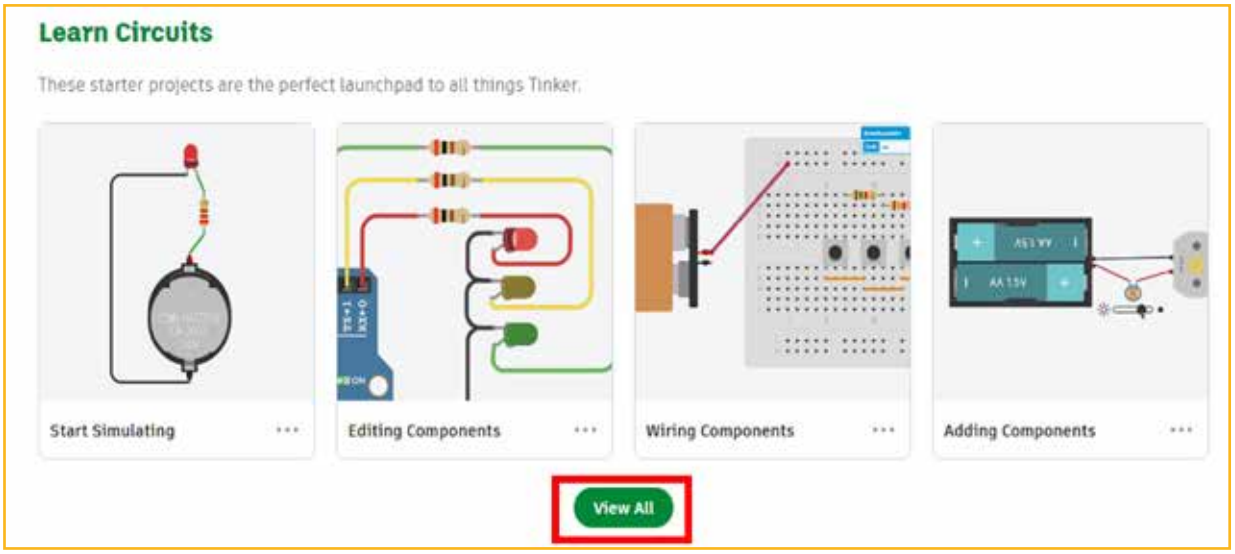
What is to be done:

Log in to Tinkercad and search for the tutorial "Blink an LED with digital output".

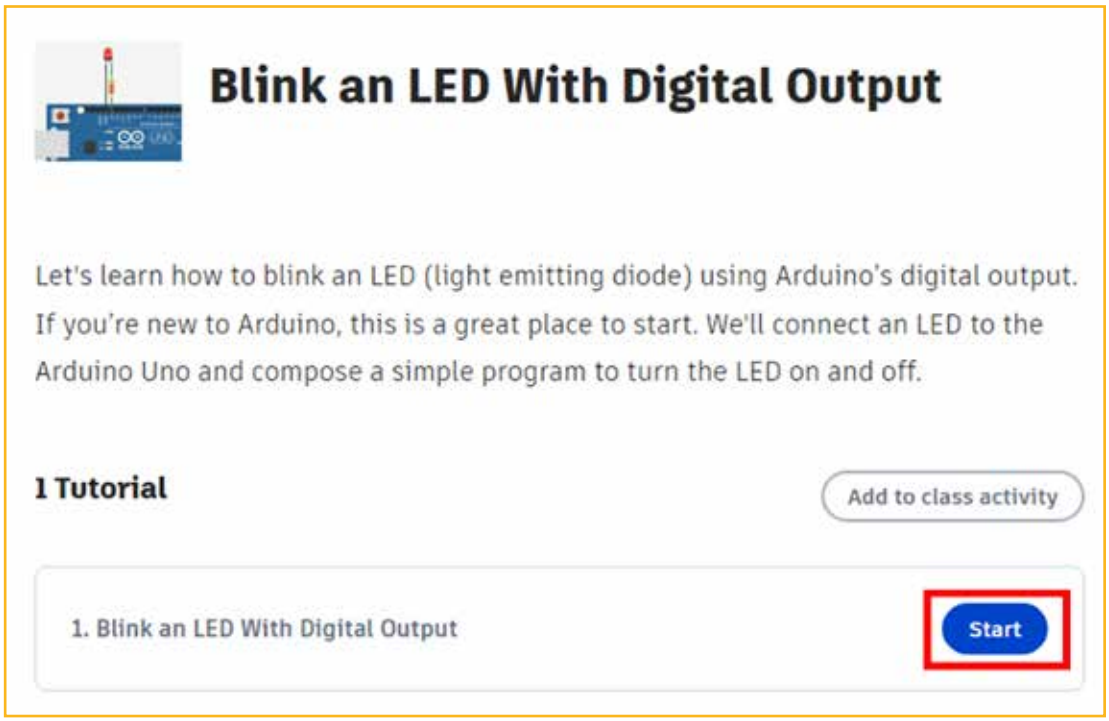
Click on the **"Resources"** tab and open the **"Learning Center"** section.



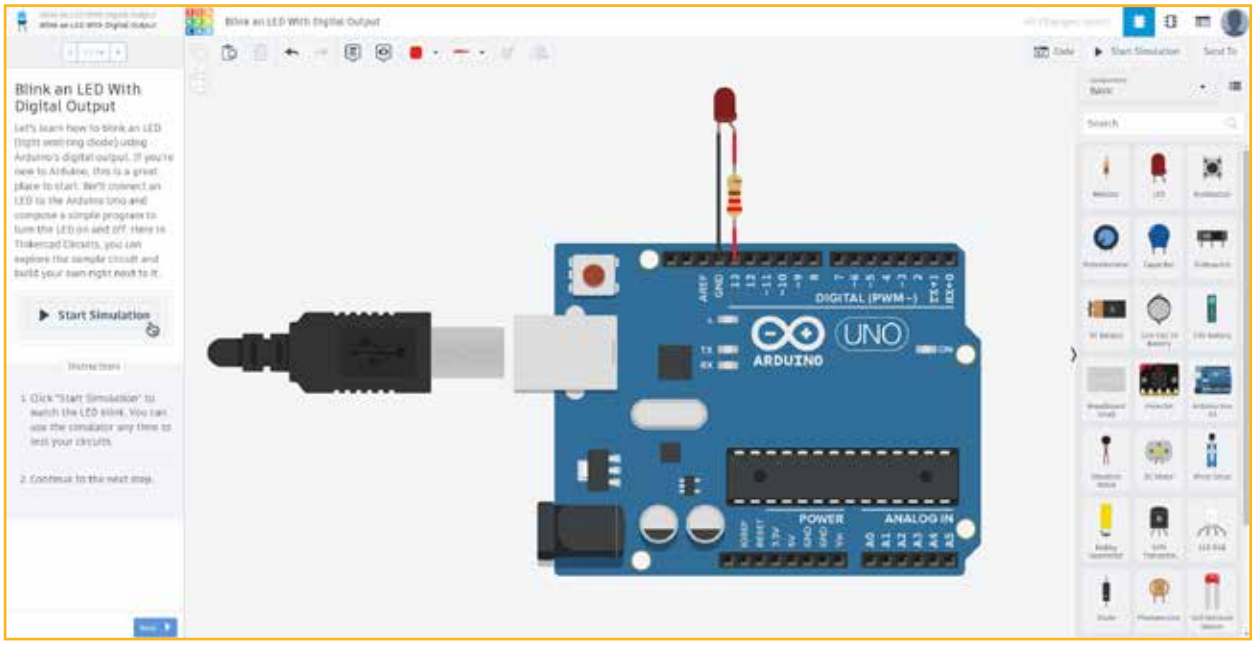
Scroll down to the category **"Learn Circuits"** and open it with the button **"View all"**.



Search for the tutorial **"Blink an LED with digital output"**, open it and **"Start"** it.



Follow the instructions and try to understand how the simulation and the microcontroller work.



Tasks

Task 1: Follow the steps and explanations described in the tutorial.
Try to understand how the simulator works.

Task 2: Compare programming with “Blocks” with programming with “Code”.
Which type is easier for you?
What problems might arise when using the blocks?

Task 3: Change the programme so that the LED is on for 0.5 seconds and off for 2 seconds.

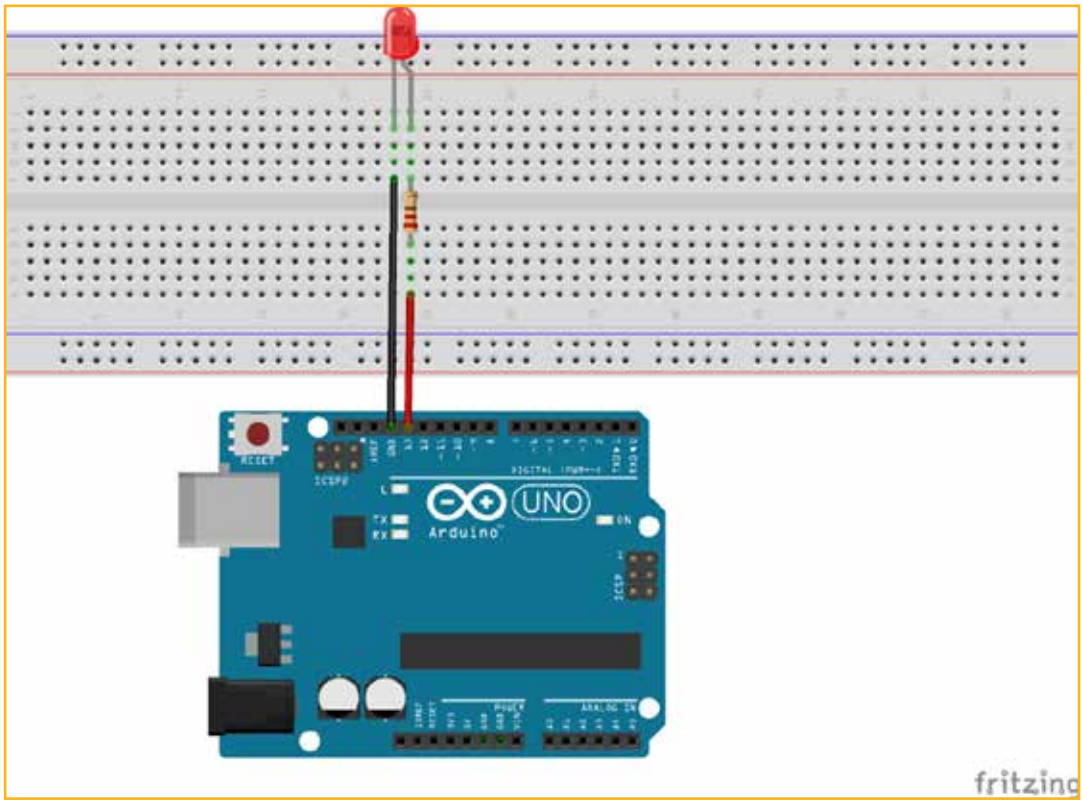
Task 4: Transfer the simulated circuit and programme to a real physical circuit with an Arduino Uno.

Tasks for experts

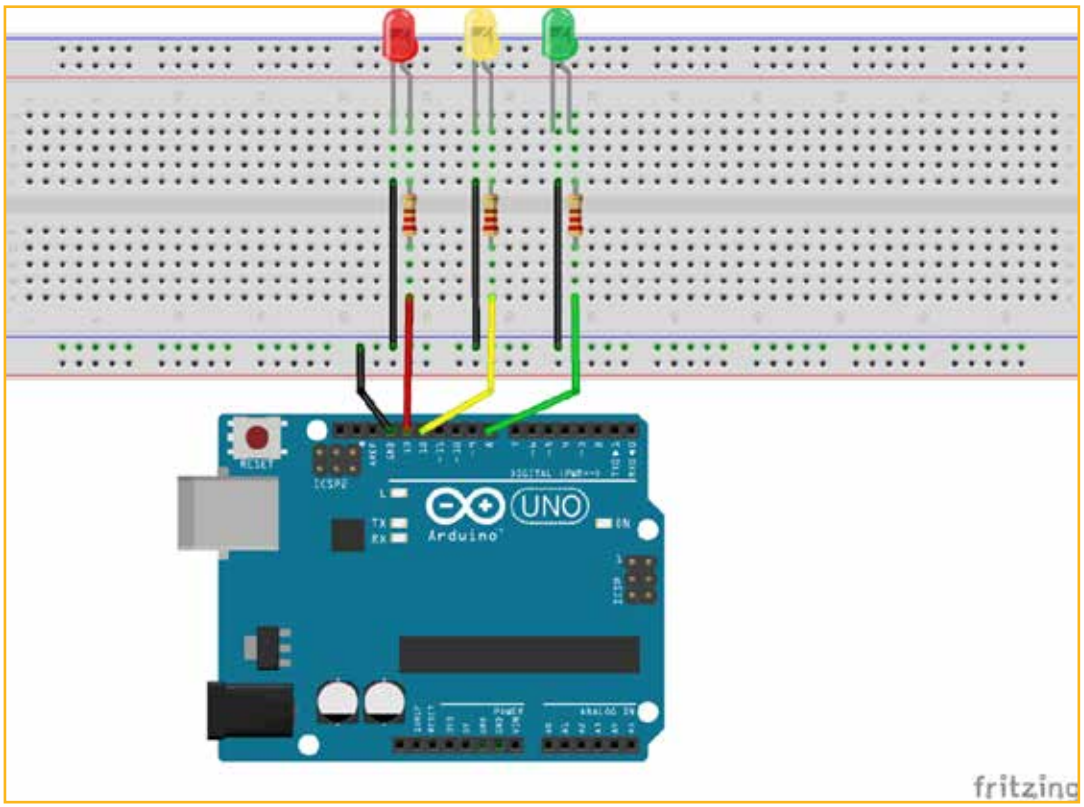
Task 5: Assign the pin assignment to the variables right at the beginning of the variable declaration (creating the name).

Task 6: Add 2 more LEDs to your circuit and control them alternately.
(Running light)

Example for a connection diagram Task 1 to Task 5:



Example for a connection diagram Task 6:



Note

A comprehensive introduction to the basics of automation technology and the basics of programming can be found in the **Textbook Module 3e "Content knowledge for Electrical Engineers - "Automation with Arduino".**

Very detailed explanations on "Getting Started with Arduino" can also be found on the Internet:

https://www.arduino.cc/en/Guide

Example Worksheet 2:
“ Using a pushbutton for switching“

All worksheets (written in blue colour) are available for download here:

https://drive.google.com/drive/folders/1lCH-JEmothzEdg8xiDfOAUL5eF7Dp6aW3

Task: Activate an LED at the touch of a pushbutton (digital input)

Learning Objectives:

- Use of a pull-up or pull-down resistor to provide a clearly defined signal (LOW or HIGH) at an input pin on a microcontroller board.
- Configuring a digital input and a digital output of the Arduino board.
- Understanding of the electrotechnical context when using pushbuttons as switches on digital inputs
- Recognition of the problem “bouncing“ of a mechanical switch
- Simulation of a current impulse circuit.

Required equipment:
Tinkercad account
Arduino-Board / breadboard / cables / pushbuttons / LEDs / resistors 10 kΩ,
(optionally resistors 470 Ω)

Short introduction:
For a basic understanding of the technical context, it is helpful to first work through the tutorial **“Pushbutton (Digital Input)“** in the simulation tool Tinkercad. Many technical explanations are also given here. Follow the instructions.

Log in to Tinkercad and search for the tutorial **“Pushbutton (Digital Input)“**.

Click on the **“Resources“** tab and open the **“Learning Center“** section.

However, it is also possible to build the tasks immediately on the hardware and write the programme via IDE-Editor.

Description how a pull-down or a pull-up resistor works

What questions need to be clarified?

- Input of a microcontroller.
- Technical necessity of using resistors.
- Description how a pull-down resistor works.
- Description how a pull-up resistor works.

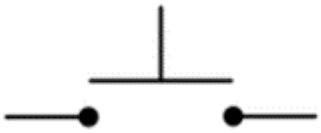
Short explanation of pull-up and pull-down circuits

1. Basics pushbutton

In everyday use pushbuttons can be found on many devices. By pressing a button, light sources can be switched on or other activities such as opening a gate or running a motor can be started.

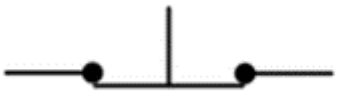
Pushbutton as normally open contact

A push button can assume two states. If it is pressed, the circuit is closed. When the button is released, the circuit is open again and no current flows. In technical language, these switches are called **"Normally Open" (NO)**. They are open in the normal state and close the circuit when actuated.



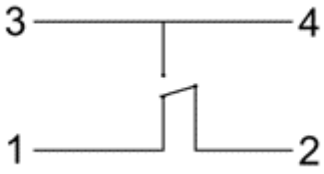
Pushbutton as opener

However, there are also pushbuttons with reversed wiring. These are called "break contacts" or "opener". They are closed in the inactive state (**"Normally Closed"** or **"NC"** for short) and interrupt the circuit when actuated. "Openers" have a great advantage in safety areas, as a wire breakage is reliably detected.



Special shapes

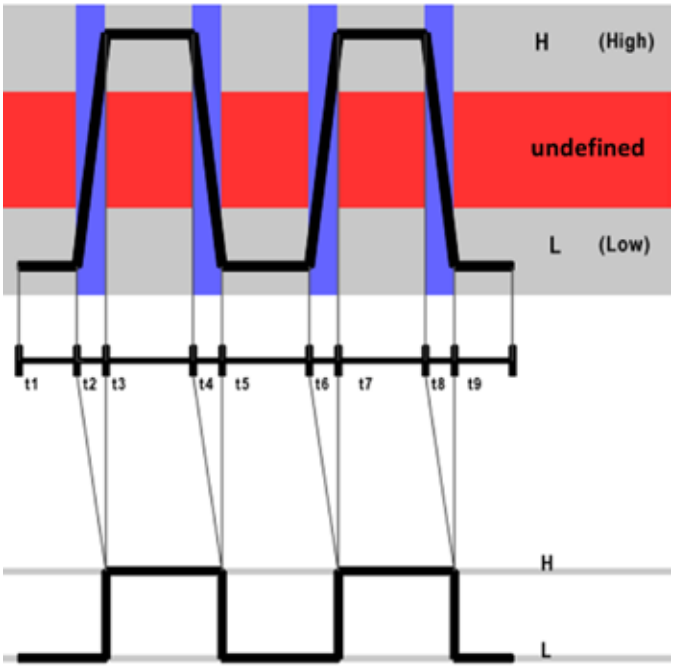
For safety-critical applications, there are also **NC/NO switches**. Since both switch states are always available, there is redundancy via the switching state. This helps to continuously check the proper function of the switch.



2. **Pushbutton on a microcontroller**
   There are two ways to connect a pushbutton (normally open contact) to a microcontroller so that a unique voltage level is applied to the input pin:
   1. If a high voltage signal (HIGH) is to be present at the input pin when the button is pressed, this is called a **"pull-down circuit"**.
   2. If, on the other hand, a low voltage signal (LOW) is to be read in when the button is pressed, a **"pull-up circuit"** is used.

A pull-up or pull-down resistor is used to provide a clearly defined signal (LOW or HIGH) at the input pin. Without a pull-up or pull-down resistor, disturbances on an open input pin can lead to falsifications so that false signals are detected by the microcontroller.

Logic level for a digital Input



An input signal at a pin can only be reliably recognised as HIGH or LOW from a certain value. If the level is between the defined values, one speaks of an undefined state. Therefore, no decision can be made during a certain period of time.

This undefined area on a pin can also be caused by external influences such as induction on the leads from electromagnetic fields. This can also lead to misinterpretation of the input signal.

3. Application with a pull-down resistor

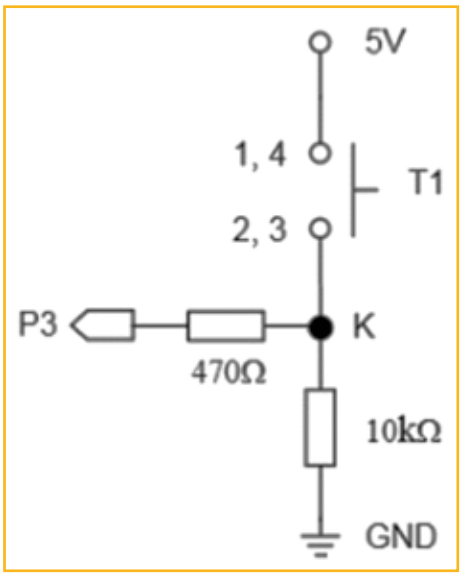
If you simply connect the pushbutton to the microcontroller, the given operating voltage (normally 5V) is applied to the input pin as soon as the pushbutton is pressed. Electrons flow from the source to the pin.

When the button is released, no more electrons can flow in, but the electrons that are on the pin remain there for a very long time because they can only flow out slowly via small leakage currents. This effect results from the fact that the input of a microcontroller is normally very high impedance. For the microcontroller, the button is therefore still seen as pressed for some time after it is released. Only when the potential has dropped due to a sufficiently large number of electrons flowing out, a LOW level is detected.

This problem can be solved by connecting the pin to GND via a sufficiently large resistor (usually 10kΩ on the Arduino boards). This allows the electrons to flow off the input pin faster and the microcontroller recognises very quickly that the button has been released. Since the resistor always “pulls down“ the voltage on the input pin to 0V, it is called a pull-down resistor.

Attention

If a too small resistor is used for the pull-down resistor, a short circuit may occur on the microcontroller board when the button is pressed!



Change in automation structures

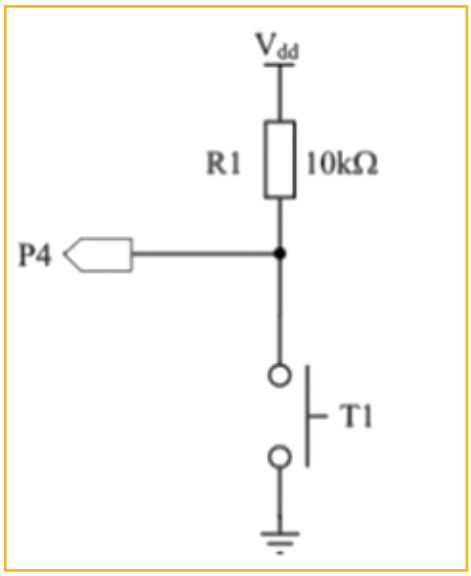
The 470Ω resistor in the circuit is intended to protect the input from excessive currents when the button is pressed. It is not absolutely necessary and can also be omitted.

4. Application with a pull-up resistor

When using a pull-up circuit, it is exactly the opposite. The pull-up resistor pulls the level up. If a pushbutton (normally open contact) is connected to the microcontroller via a pull-up, a “HIGH signal“ is present at the digital pin when the pushbutton is not pressed. When the button is pressed, it pulls down the voltage via the resistor and ground is applied to the input pin. The microcontroller then receives a LOW. At the same time, the resistor prevents a short circuit.

The logic of a pull-up circuit is therefore reversed. A pressed switch results in the level LOW at the input. This must be taken into account when evaluating the input.

For pull-up circuits, resistors in the range of a few kΩ are used.
For most microcontrollers, 10kΩ is a safe default value.



Circuit with pull-up resistor

5. Debouncing of a switch

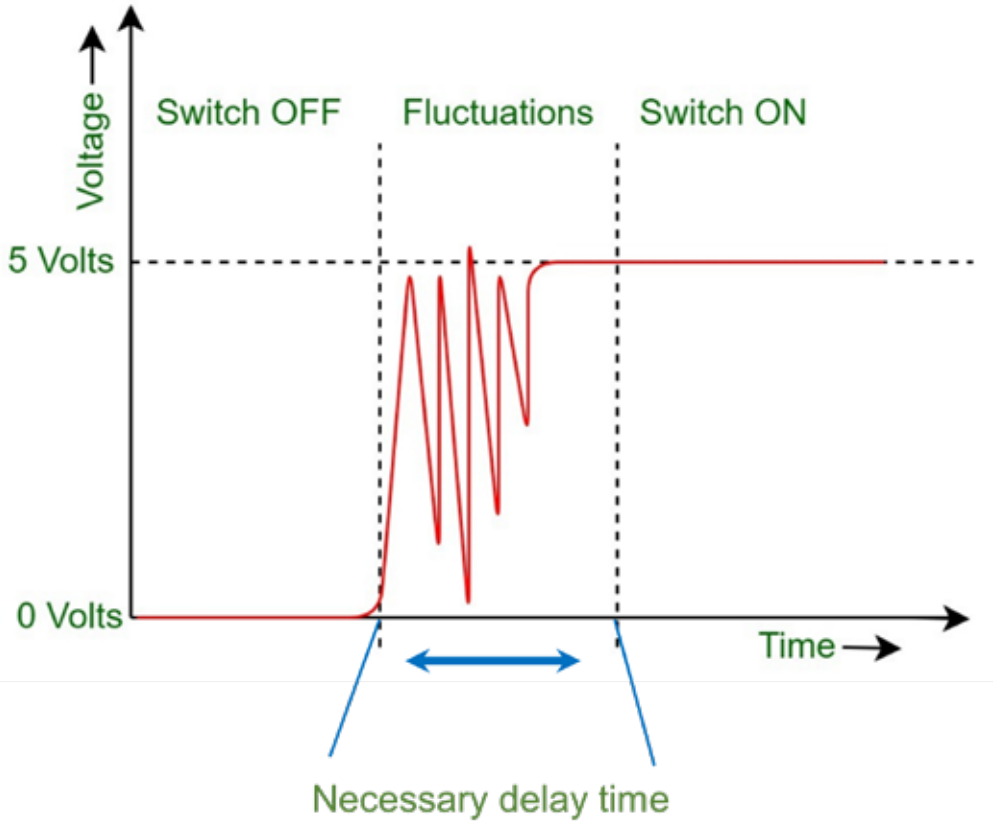
The signal of a normal button does not immediately change to the desired state (LOW to HIGH or HIGH to LOW) when pressed or released. This behaviour is called **“bouncing“**.

Bouncing describes the rapid oscillation between the states “High“ and “Low“ or between “1“ and “0“. The reason for this is the mechanical oscillation of the switch contacts.

Since the microcontroller can read the state of a digital input very quickly, depending on the programme, it is possible that it interprets the pressing of the button as several button presses. The input signal will oscillate between two states for some time after the button is pressed, making it difficult to tell the real state.

To obtain a safe signal at the digital input of the microcontroller, it is necessary to **"de-bounce"** the input signal. This can be done on the hardware side, but also by software.

The simplest method is to insert a delay() command after reading in the state. This delays the immediate re-reading of the switch state in the critical switching phase. Usually 10 milliseconds is sufficient (delay(10)).



The 470Ω resistor in the circuit is intended to protect the input from excessive currents when the button is pressed. It is not absolutely necessary and can also be omitted.

Also have a look at:

https://roboticsbackend.com/arduino-push-button-tutorial/

https://www.arduino.cc/en/Tutorial/BuiltInExamples/Debounce

Tasks

Task 1: An LED should light up for 5 seconds when a pushbutton has been pressed.

Task 2: An LED should light up for 5 seconds when pushbutton_1 is pressed.
The LED is to flash 5 times when the pushbutton_2 is pressed.

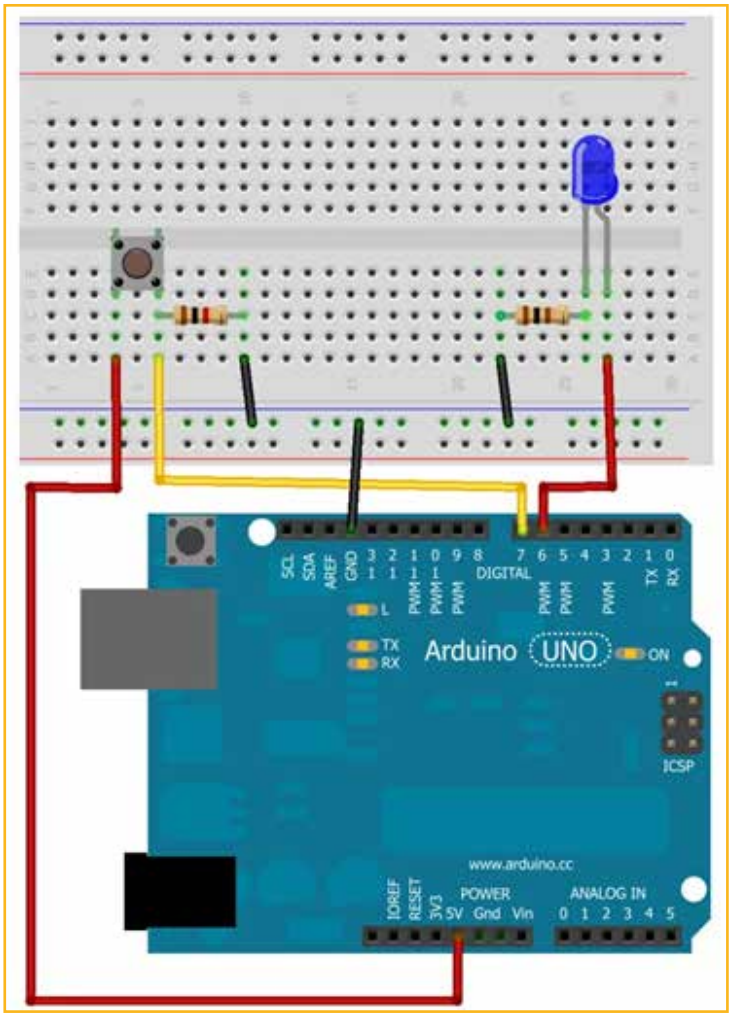
Task 3: Explain the difference between the circuit with a pull-down resistor and a pull-up resistor.

Task 4: Change your programme in a way that the LED lights up by default and goes dark when the button is pressed.

Tasks for experts

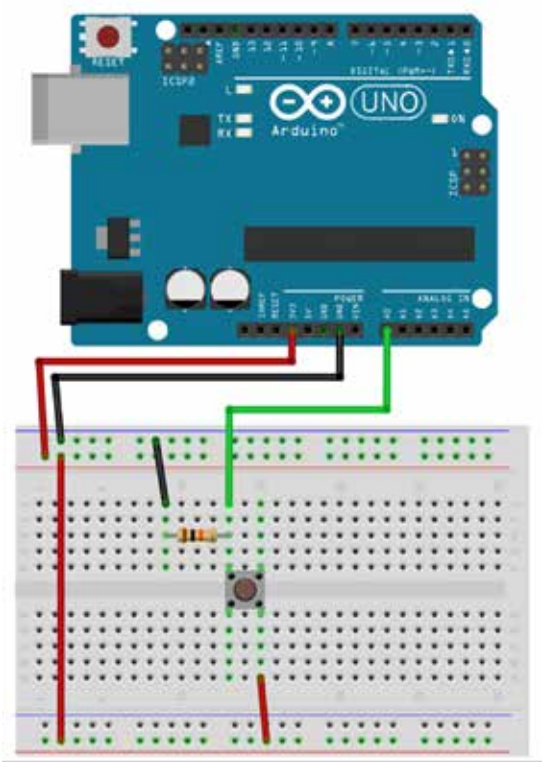
Task 5: Reconstruction of a changeover switch based on a current impulse circuit:
An LED is to be switched on or off with a pushbutton_1 as well as with a pushbutton_2.

Example for a connection diagram Task 1 and 2:

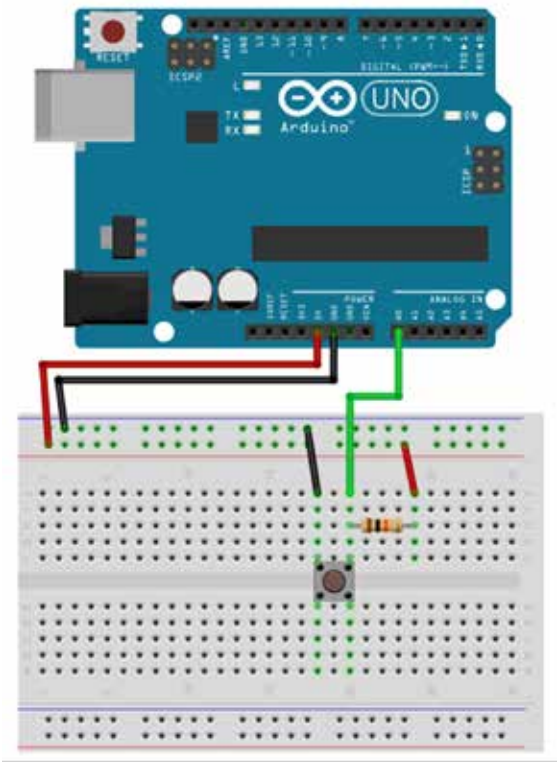


Connection diagrams Task 3

Explain the difference between the circuit with a pull-down resistor and a pull-up resistor!

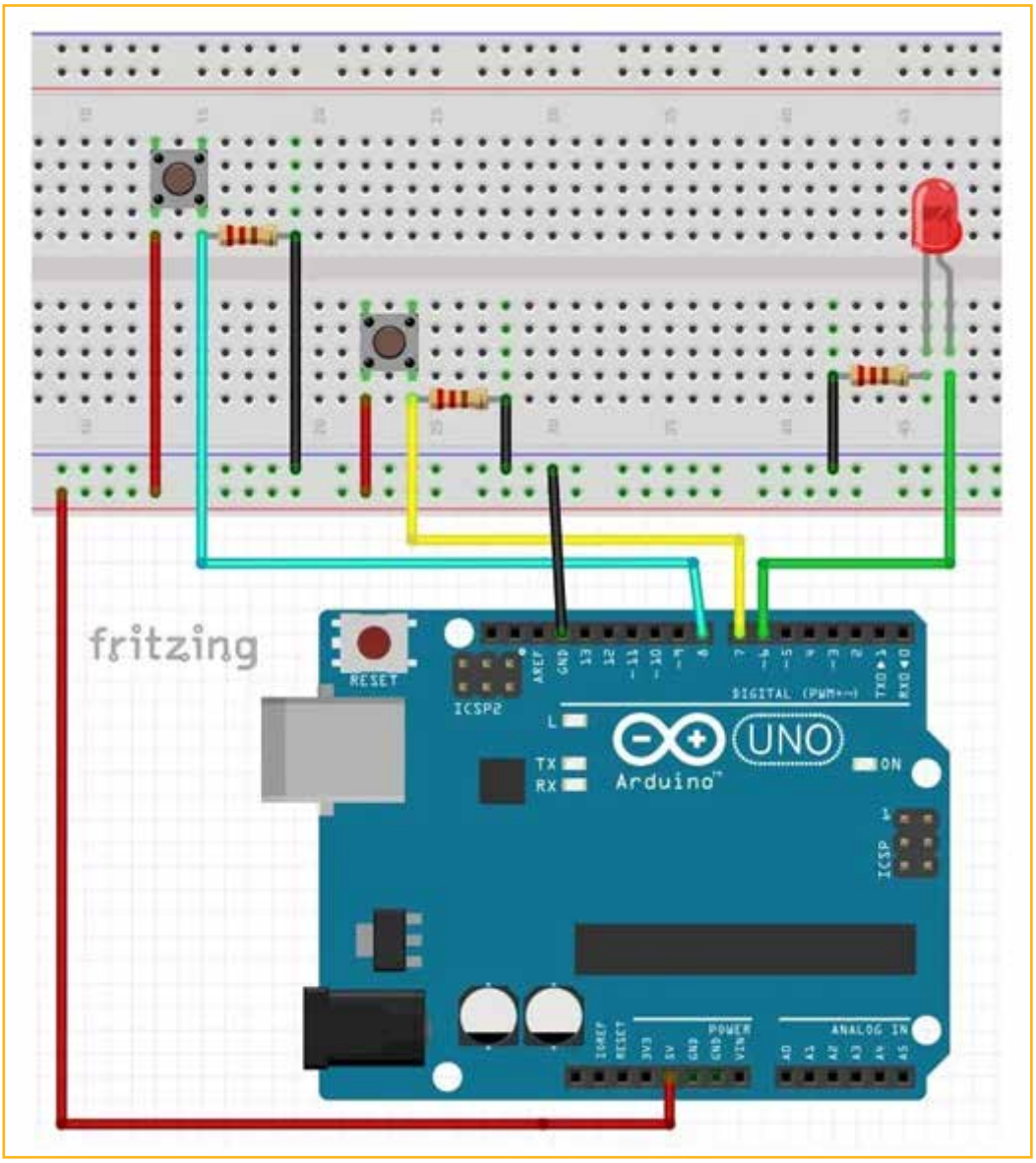


Circuit with pull-down resistor



Circuit with pull-up resistor

Example for a connection diagram Task 5:



For all other teaching units, you can also rely on suggestions from Tinkercad and the Internet. However, it is important to supplement these very rudimentary tasks with your own tasks and ideas. Use worksheets 1 and 2 as a guide.

List of usable Tinkercad contents

	Content	Tasks	Tinkercad Links
Teaching Unit 1: Introduction (First steps)	Project: Introduction to the topic of microcontroller (first programming: "flashing light")	Let's learn how to blink an LED using Arduino's digital output. (Connecting an LED to a microcontroller and writing a first programme.) **Have a look at the explanations in Worksheet 1!**	
Teaching Unit 2: Light	Project: Pushbutton (Digital Input)	In the next step, the light is to be switched on and off by a push button. **Have a look at the explanations in Worksheet 2!**	
Teaching Unit 3: Low beam light	Photo resistor as sensor: Automatic switching between high and low beam in a car	In modern cars, the high beam is automatically dimmed for oncoming cars so that drivers are not dazzled. The next task is to realise such a function with the help of a photoresistor. (In this context, the basics of the electrical voltage divider must be taught.)	
Teaching Unit 4: Parking sensor	Project for programming a parking aid with ultrasonic distance measurement	Almost all parking assistants on the market are based on the same technology for distance detection by ultrasonic sensors. Our project is now also to be supplemented by a simple parking aid, so that a warning message is displayed by LED and/or buzzer when the distance falls below a certain level.	
Teaching Unit 5: LCD display	Project: LCD display to show the measured distance	The parking sensor circuit from the "Parking Sensor" project is now to be supplemented with a display of the measured distance and the sketch adapted accordingly. A two-line display with 16 characters per line and an I2C bus connection is used for this.	

Teaching Unit 6: Adaptive cruise control	Project: Adaptive Cruise Control (ACC) (Programming a “for-loop”)	Adaptive cruise control (ACC) is used to automatically regulate the vehicle speed. As soon as an obstacle is detected in front of the vehicle, the speed of the vehicle is automatically adjusted and a safe distance is maintained.	
Teaching Unit 7: Switch Case	Project: Automatic instrument backlighting depending on ambient brightness	The brightness of the cockpit lighting in a car is to be controlled in four stages depending on the ambient light intensity. When it is absolutely dark, four LEDs should light up, when it is bright, none.	

Especially in projects 4 and 6, the students have to deal with physical basics and mathematical correlations.

Information on project 4:
In order to implement and program a suitable ultrasonic sensor for distance measurement, it is necessary to know some basic physics about the propagation of sound waves.

1. **Sonic speed and echo**
   Sound waves, tones and noises propagate at the so-called speed of sound. The speed of sound depends on the medium and the temperature in which the sound propagates. For example, the speed of sound at 20 °C is approx. 343 m/s in air and 1.484 m/s in water.

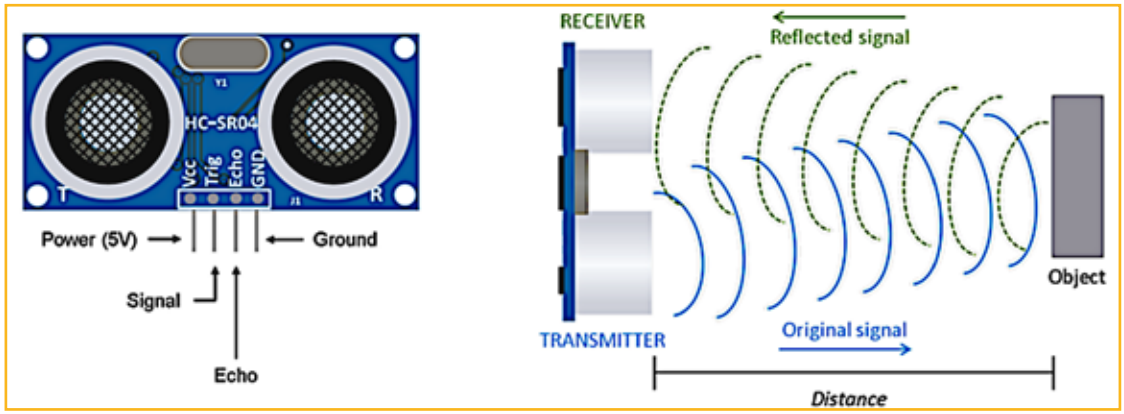
   The following applies:

   $$v = s \cdot t$$ (v – speed; s – distance; t – time)

   According to this, sound in air travels a distance of 343 m within one second, because:

   $$s = v \cdot t = 343\ m/s \cdot 1s = 343\ m$$

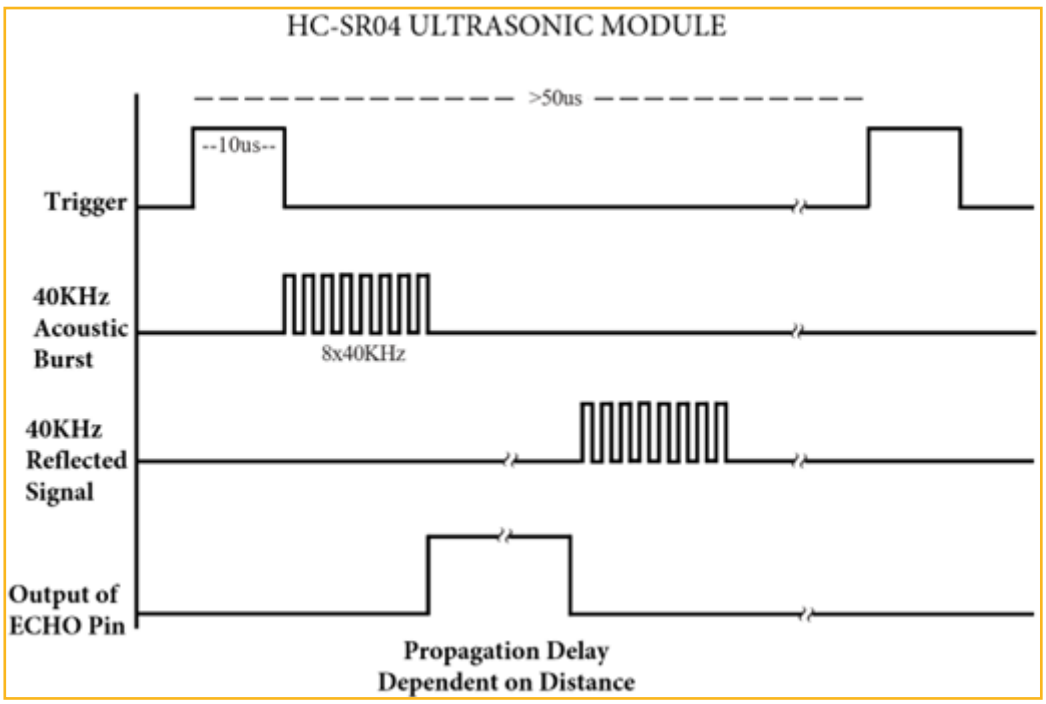
If the sound wave emitted by a source hits another object, such as a wall, as it propagates in air, part of the sound wave is absorbed and part is reflected - i.e. thrown back. If the reflected part of the sound wave reaches the source again, this can be perceived as a so-called echo.



2. How an ultrasonic sensor works

The ultrasonic sensor, consisting of a transmitter and a receiver unit, can measure the travel time of the sound wave and thus calculate the distance between the sensor and the object:

(1) The sensor emits a short pulse.
(2) The time measurement is started.
(3) The sound wave is reflected.
(4) The echo is detected by the sensor.
(5) The time measurement is stopped.
(6) The distance can be calculated from the measured value (cf. calculation formula).



3. Possible tasks

- Construction of the hardware (or simulation in Tinkercad)
- Integration of the ultrasonic module into the programme code
- Control and evaluation of the ultrasonic module

Also have a look at:

https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

Information on project 6:
Understanding and implementing adaptive cruise control (ACC)

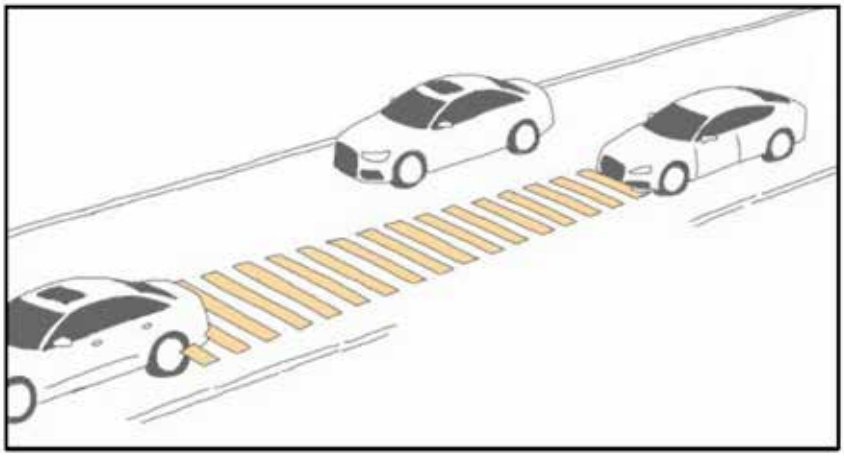
1. What is a relative speed?

Two speeds must be recorded for the task:
The speed of the own vehicle and the relative speed to the vehicle in front.

The speed (v) of the own vehicle can be calculated via the distance travelled (s) per time (t).
A relative speed is related to a specific location.

Example

The speed of the own vehicle is **100 km/h**, a vehicle in front is travelling at **90 km/h**.
The speed of the vehicle in front is **-10 km/h** relative to the own vehicle.
When the speed of the vehicle in front is **-10 km/h** relative to one's own, i.e. the vehicle in front is approaching one's own at **10 km/h**.



Graphic: R. P. Dröge, www.lehrer-online.de

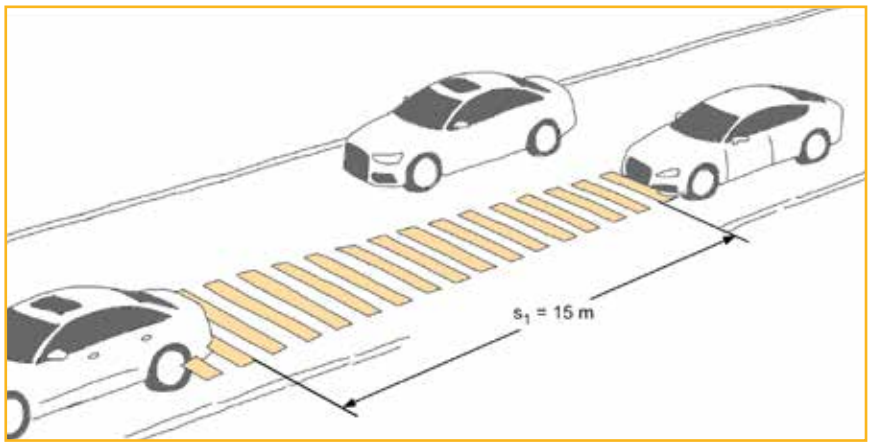
2. Detecting the relative speed

Unfortunately, cars do not yet communicate with each other, so the speed of a vehicle in front must be determined by measurements. This can be done most easily with time-delayed distance measurements.
By taking two distance measurements it is possible to determine a distance.

Example: First measurement 50 m, second measurement 40 m.

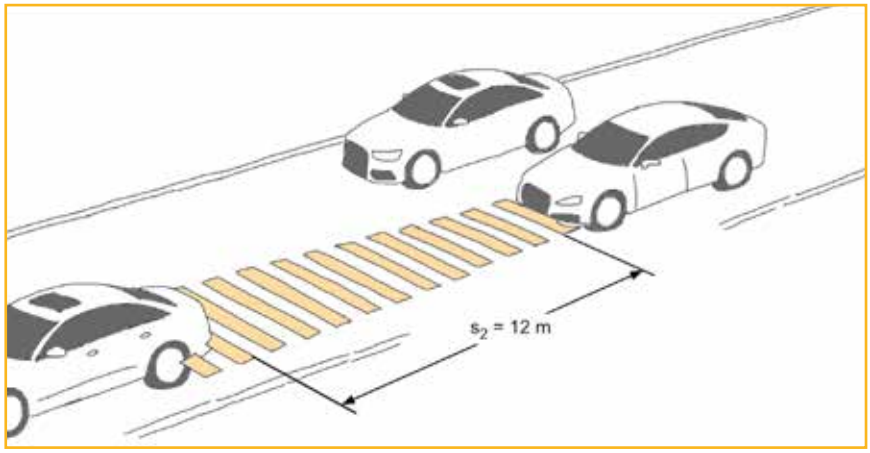
From this it follows that...
...the vehicle in front has moved 10 m towards your own. The speed can be determined if the time between the measurements is known.

Measurement 1:



Graphic: R. P. Dröge, www.lehrer-online.de

Measurement 2 after 1 s:



Graphic: R. P. Dröge, www.lehrer-online.de

In this example, the rear car is approaching the front car with a relative speed of **-3 m/s**.

3. **Possible tasks**

- Create the programme section with which the relative speed can be calculated. This is to be output in m/s.
- Distance measurements with a **"For Loop"**
  As soon as an obstacle appears in the front distance meter, two measurements are to be taken in order to determine the relative distance travelled. This can be realised with a **"For Loop"**, as then the code for the measurement only has to be stored once in the programme.
- If a difference is detected, the own speed is either increased (distance has increased) or decreased (distance has decreased).

You can use these tips very well in an interdisciplinary lesson.

Next steps

Further examples are given in the following chapters.

Special attention will be paid to the very important topics of

- IoT (Internet of Things)
- Robotics / Automation
- Sensor technology

Chapter 2

Basic knowledge of how IoT works

2.1 THE INTERNET OF THINGS (IOT) - DEFINITION, TECHNOLOGY AND APPLICATIONS

The **Internet of Things (IoT)** is a system of machines, plants and devices connected via and with the Internet. The associated "things" must be clearly identifiable. There is currently no generally accepted definition of the term. The term "Internet of Things / IoT" is often also used as a "collective term" for various technologies.

The **Internet of Things** refers specifically to the increasing networking of physical objects that are connected to each other and to the Internet and can exchange data. These "things" range from simple household appliances to highly complex industrial machines. Connected devices use built-in sensors to collect and exchange data with other devices and systems over the Internet, triggering responses to specific values.

The Internet of Things (IoT) is the prerequisite for Industry 4.0!

2.1.1 Origin of the designation

The term **"Internet of Things"** is attributed to the British technology pioneer Kevin Ashton, who described the IoT in 1999 as a system of sensors connected to the Internet,

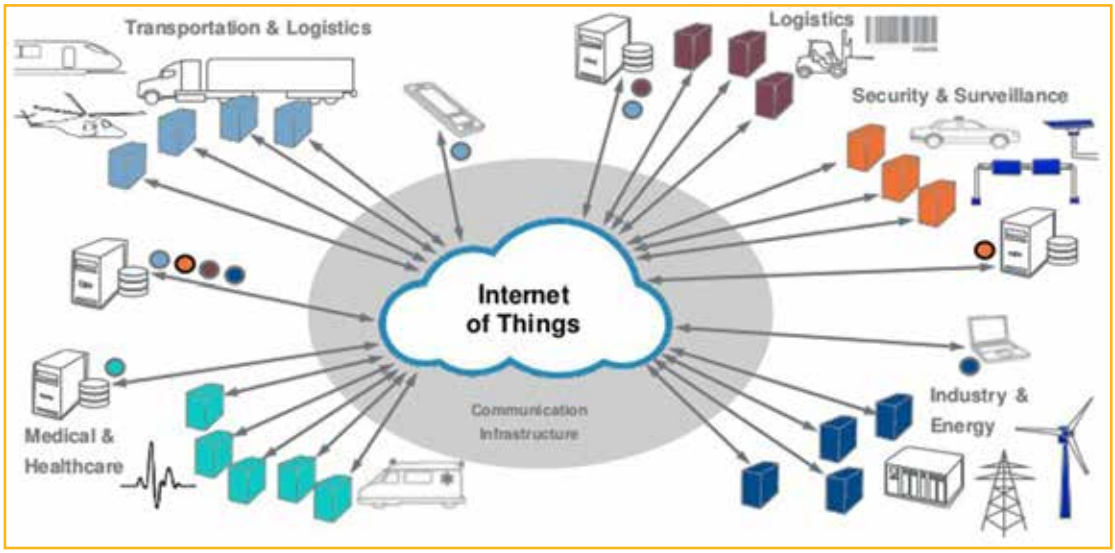
> *"... that behave like the internet by creating open, spontaneous connections, sharing data freely and supporting unforeseen applications."*

The idea of the Internet of Things dates to the early 1980s. The first connected device was a beverage dispenser at Carnegie Mellon University. This early IoT device was networked to check its inventory and determine how many drinks were available.

2.1.2 Significance for industry

In the industrial sector, IoT technologies are applied especially in the control of sensors and devices that are equipped with cloud technologies. In other words, intelligent networking of machines and processes takes place. The term **IIoT - Industrial Internet of Things** - is mainly used here.

The **IIoT** refers to digitally networked, intelligent machines or systems in an industrial context, especially in manufacturing. The goal of the Fourth Industrial Revolution is efficient, self-organised production in which machines, plants, products, and people communicate and cooperate with each other. This networking is intended to optimise the entire value chain in the company, from the idea of a product, through development, production, marketing and sales, to use, maintenance, and recycling.



With the advent of cloud technology in the early 2000s and the development of the **OPC/UA protocol**, it became possible to store data and transfer it securely between different devices. Thus the IIoT was born.

2.1.3 How important are sensors for the Internet of Things?

Sensors are both the starting point and at the same time the biggest data providers of the Internet of Things. More precisely, devices that are equipped with many sensors. They enable **„things“** to collect data and perform actions. With access to the Internet, sensors turn functional objects into **smart devices.** Large amounts of data, i.e. **Big Data,** become a source of valuable information.

2.1.4 What is hidden behind the terms **edge computing** and **cloud computing**?

How and where data streams are processed affects the distinction between cloud and edge computing. To save storage capacity and reduce latency, it may be advisable to pre-store the huge data streams generated by the IoT at the edge of the network and sort them by relevance before storing them, for example in the **cloud**. Working with only relevant data then benefits from efficient use of the available bandwidth and computing power.

Edge computing is when data processing is moved to the edge of the network, system or machine. Here, the information is pre-sorted into useful and useless before it is sent to the cloud or server. In this way, edge computing helps to reduce the load on conventional computing systems.

2.1.5 Benefits for the industry

Predictive Maintenance
Thanks to analytics and machine learning, failures can be detected in time and the machine can be serviced before a failure even occurs. In addition, it is possible to increase the system performance of the processes through clever identification and targeted elimination of bottlenecks.

Flexible production
The more complex a product, the more companies or departments can be involved, contributing to the product step by step. With intelligent networking, the steps can be better set up and machine utilisation can be better planned.

Optimised logistics and smart supply chains
With access to all necessary data, algorithms can calculate ideal delivery routes and quantities of required materials. Intelligently networked logistics save time and resources. Machines can also independently report the need for materials. A smart flow of goods can avoid supply gaps.

Efficiency increases through data analysis
Data on the process of incoming materials, etc., production and the condition of a product are collected and evaluated. This provides information on how to optimise production, increase process efficiency and improve product quality.

2.2 IOT PLATFORMS SUITABLE FOR EDUCATION

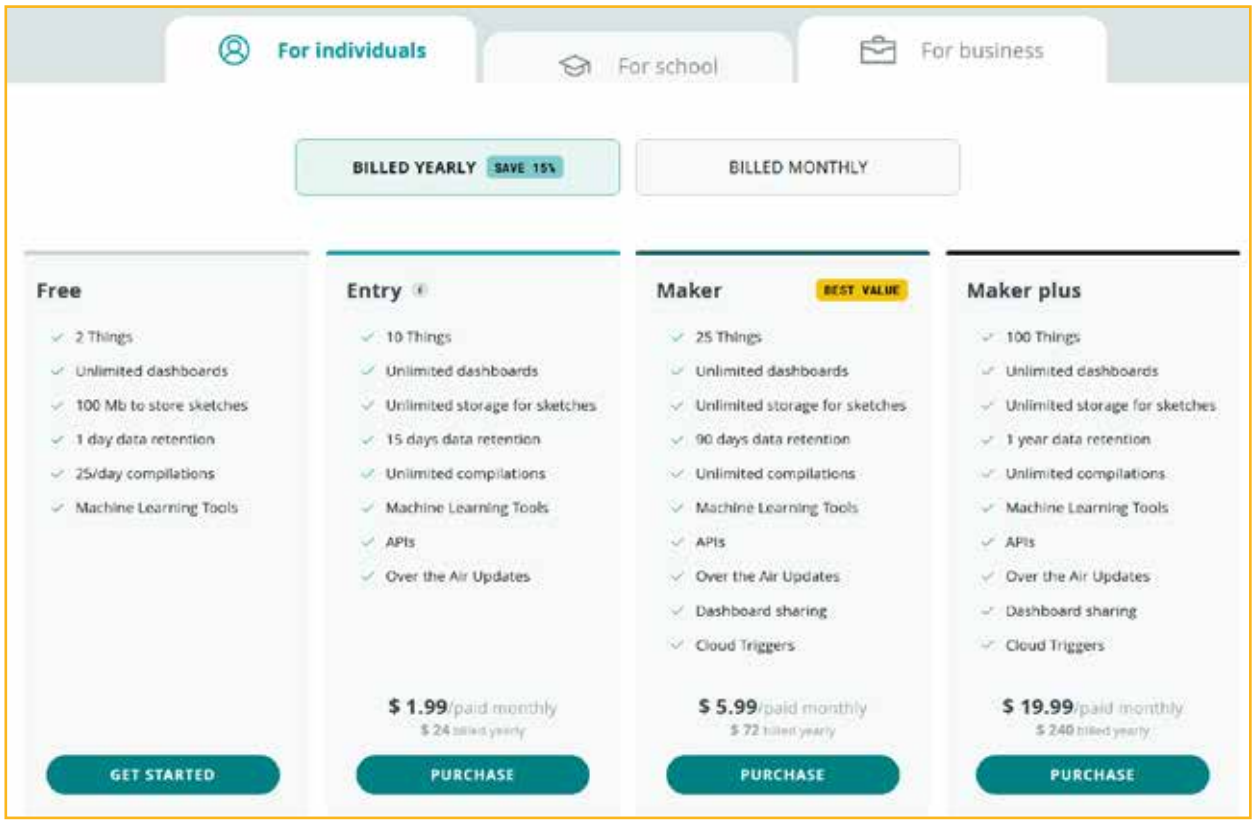
With the help of the Arduino IoT Cloud, many Arduino boards can be connected to the Internet very quickly and securely. Controlling projects and reading out data remotely is easily mastered in this way, even for beginners. The Arduino IoT Cloud project was launched at the beginning of 2019.

Of course, there are other cloud computing offerings with Arduino components, but the Arduino IoT Cloud seems the most suitable because of its high penetration rate and proximity to schools.

https://cloud.arduino.cc

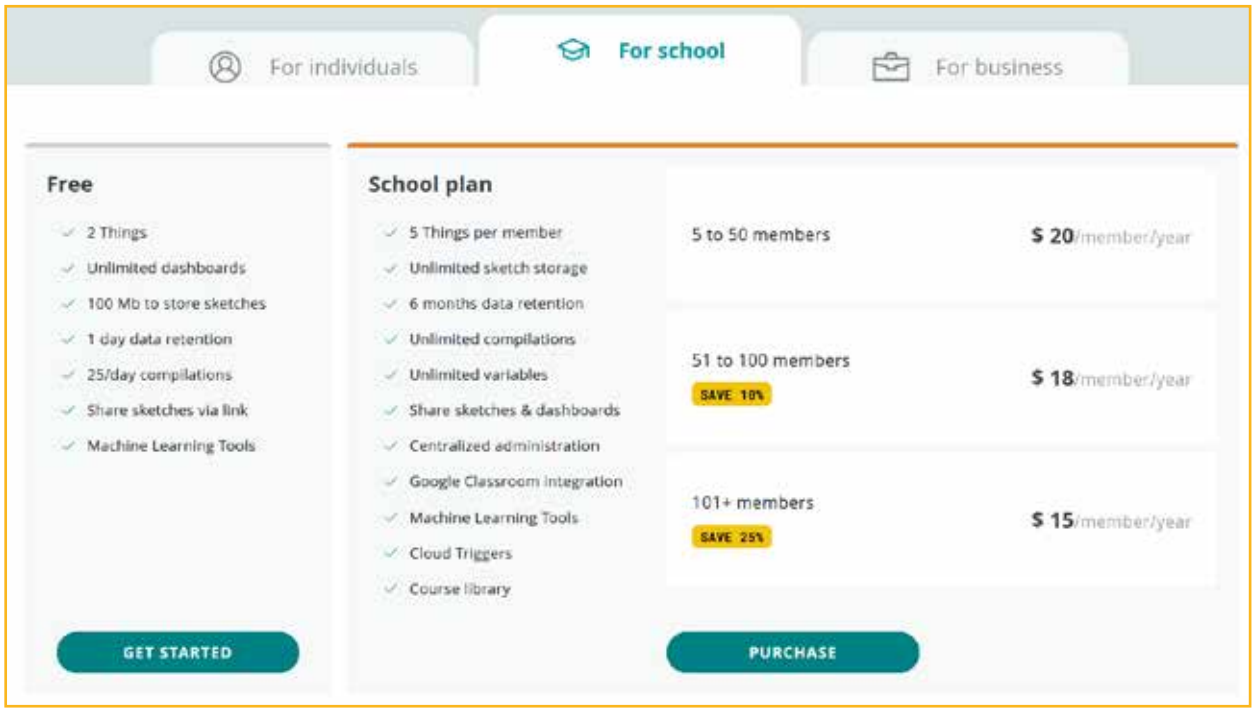
https://docs.arduino.cc/arduino-cloud/

Different account levels **(Plans)** are available for use:
„For individuals“, for example, is good for the students:



(Status: 27.07.2023)

Or „**For school**“ with special features for education:



(Status: 01.08.2023)

Before you decide on a form, first use the „Free Plans“ to get to know the platform.
You can update later as needed.

Interesting alternative to the Arduino Cloud
An interesting alternative to the Arduino Cloud is the Blynk platform. Blynk is very close to industrial solutions but uses a slightly different structure.

- *https://blynk.io/*
- *https://blynk.io/blog/step-by-step-video-guide-to-making-your-first-project-on-the-new-blynk*
- *https://www.instructables.com/Get-Started-With-Blynk/*
- *https://how2electronics.com/home-automation-using-esp32-blynk-2-0-with-feedback/*
- *https://blynk.hackster.io/projects?difficulty=beginner&part_id=10887&sort=trending&type=protip*

2.3 STRUCTURE AND SECURITY OF THE ARDUINO CLOUD

2.3.1 What's behind the "Arduino Cloud IDE"?

The **Arduino Cloud IDE** (formerly known as Arduino Create) is a cloud-based development environment for Arduino that can be used with any modern internet browser. You can easily log into a fully functional Arduino IDE from anywhere in the world, which stores and compiles your code in the cloud. It all comes together in your own personal **„Arduino IoT Cloud"**.

The Arduino IoT Cloud is Arduino's physical computing platform. It enables businesses as well as universities, colleges, school classes and other educational institutions to implement IoT projects easily and securely.

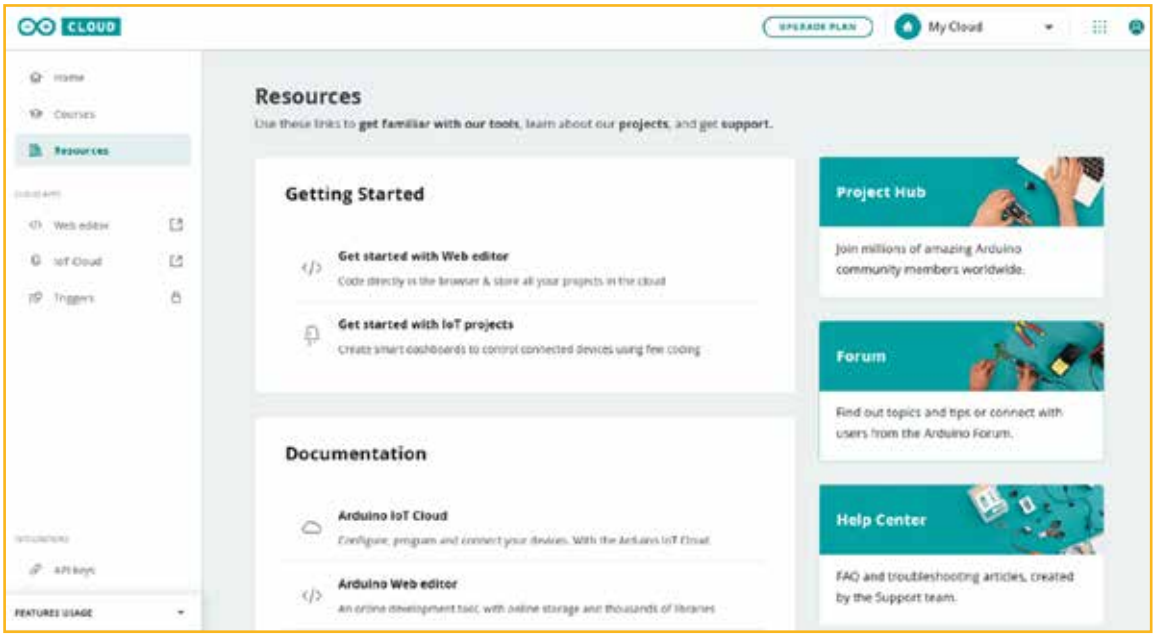
The Arduino Cloud provides the best conditions for experimenting in the IoT environment. A highly simplified data model that is very easy to use and the ability to create attractive browser dashboards with little effort make it relatively easy to get started.

2.3.2 Arduino IoT Cloud briefly explained

The Arduino IoT Cloud is designed to connect sensors, actuators and IoT devices in the cloud and to view and edit their data and states remotely. The Arduino IoT Cloud consists of **easy-to-use, open-source** hardware and software. This means that not only the more than 100 Arduino hardware products such as shields, kits, boards and other accessories can be connected to this cloud, but also third-party products. For beginners, it offers a **complete end-to-end platform** for the Internet of Things (IoT) - including **easy access to microcontrollers and programming.**

With these features, the Arduino IoT Cloud is a **great way to introduce students to connected projects and the Internet of Things.** First projects for beginners can be implemented quickly and safely. But it can also be used for more complex tasks.

In addition, there is always the possibility to use the power of the community. This can be done on the Arduino Project Hub by browsing the projects for suggestions and making them your own, or by accessing FAQs and troubleshooting articles from the Help Centre.



2.3.3 Benefits for education

The Arduino IoT Cloud makes it possible to connect a wide variety of sensors and control units to collect data and control actuators anywhere. The IoT Cloud is the platform where this collected data can be stored, pre-processed, analysed and visualised in one place. It is also possible to control devices from anywhere via the cloud.

Especially for students, the plug-and-play solution with Arduino hardware makes it **easy to get started,** allowing them to connect their devices to the Arduino IoT Cloud quickly and easily without complicated configuration. Clear dashboards, automatic code generation and a wealth of examples and templates provide a **sense of achievement** and show students the tangible benefits of their IoT project.

2.3.4 How safe is it to use the Arduino Cloud?

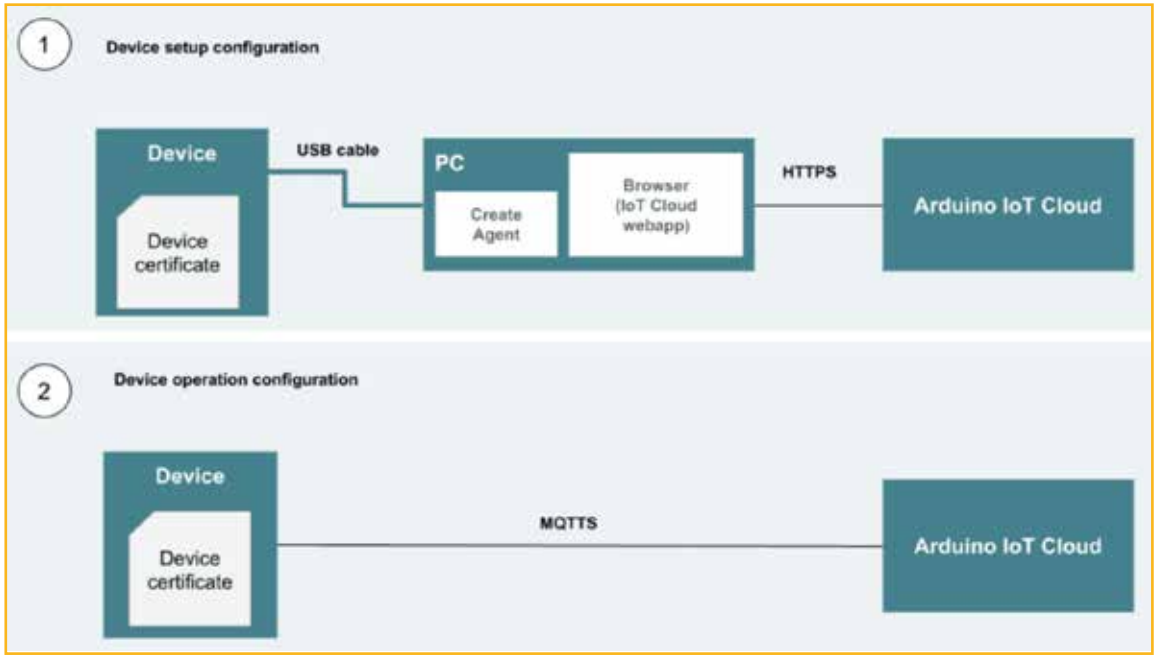
The Arduino Cloud is a **simple and secure** way to connect Internet-based devices to each other or integrating them into a network for data exchange. The cloud servers are used for programming, integrating the devices into the relevant Wi-Fi and exchanging data.

As the programming takes place online on a server and the compilation (translation into machine language) takes place in a secure environment, it is relatively difficult to break into the system once the programme has been loaded into the device. All security-relevant information is stored and processed on the cloud server. So the only vulnerability is access to the cloud.

Make sure that access to the cloud doesn't fall into the wrong hands!

2.3.5 Security Considerations for Arduino Boards

Device onboarding and communication with the Arduino Cloud is ensured to protect the confidentiality of the data exchanged between Arduino devices and the Cloud. The following figure shows the device setup phase and the device operation phase.



Device Setup & Operation Phases
(https://docs.arduino.cc/arduino-cloud/features/security-considerations)

Device setup phase:

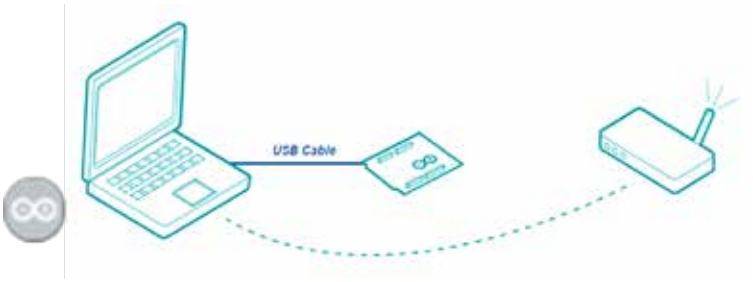
1. The setup phase is initiated by the web browser with a secure connection to the Arduino IoT Cloud over HTTPS requesting to create a new device.
2. The IoT Cloud provides a special sketch that is run on the device to generate:
   - a private and public key
   - a Certificate Signing Request (CSR)
3. The private and public key are stored in the secure element on the device.
4. The CSR is sent from the device to the browser via the create agent.
5. The browser sends the CSR to the IoT Cloud for signing.
6. The Certification Authority of the IoT Cloud signs the certificate and returns the public certificate to the browser.
7. The browser uploads the public certificate to the device via the Arduino Create Agent.

The device establishes a mutual-TLS authenticated connection with the IoT Cloud MQTT server. The mutual-TLS connection is established so that:

1. The MQTT server can establish the identity of the device connecting to it.
2. The device can establish the identity of the MQTT server it is attempting to connect to; to establish such secure connections, devices must be able to connect to *.arduino.cc on port 8884 and 8883.

2.3.6 How does the Arduino IoT Cloud work?

Physical endpoints have the name **„Thing“** in the Arduino IoT Cloud. When you first log in, the user-friendly service prompts you to create your first Thing. As part of the setup process, you will need the **„Arduino Create Agent“**, a tool that enables communication between laptop, cloud, and connected hardware.



The central building blocks of communication between the end device and the cloud service are the memory fields known as **“variables”.**
Depending on the configuration, variables record changes either on the client, on the server, or on both sides. The Arduino Cloud then automatically synchronises the data.

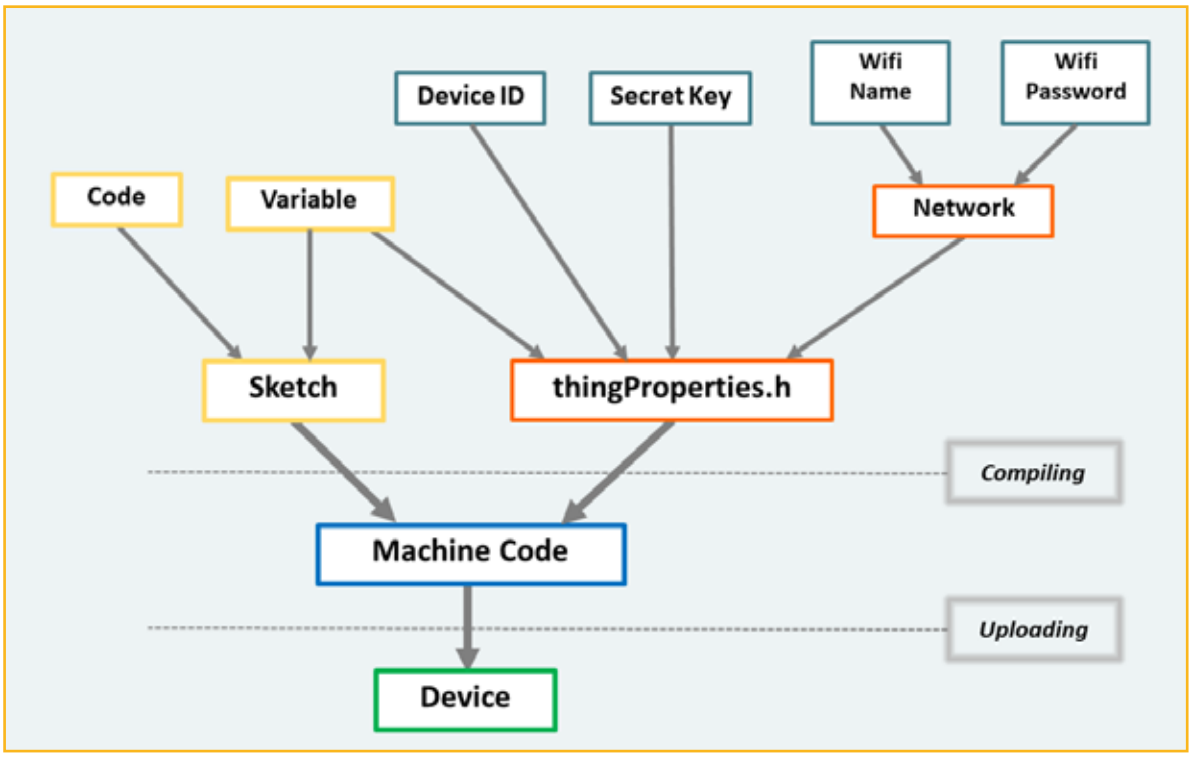
The „real intelligence“ of the cloud service lies in the generated project in the file **„thingProperties.h“,** which provides an initialisation function and the local variables according to the following scheme:


```
// Code generated by Arduino IoT Cloud, DO NOT EDIT.

#include <ArduinoIoTCloud.h>
#include <Arduino_ConnectionHandler.h>

const char DEVICE_LOGIN_NAME[]  = "2cd18f9a-e945-4167-963b-9f7e5618c288";

const char SSID[]               = SECRET_SSID;     // Network SSID (name)
const char PASS[]               = SECRET_OPTIONAL_PASS;    // Network password (use for WPA, or use as key for WEP)
const char DEVICE_KEY[]         = SECRET_DEVICE_KEY;    // Secret device password

void onPumpSwitchChange();
void onLedSwitchChange();

CloudSwitch pump_switch;
CloudTemperatureSensor temp;
CloudRelativeHumidity humi;
bool led_switch;

void initProperties(){

  ArduinoCloud.setBoardId(DEVICE_LOGIN_NAME);
  ArduinoCloud.setSecretDeviceKey(DEVICE_KEY);
  ArduinoCloud.addProperty(pump_switch, READWRITE, ON_CHANGE, onPumpSwitchChange);
  ArduinoCloud.addProperty(temp, READ, ON_CHANGE, NULL);
  ArduinoCloud.addProperty(humi, READ, ON_CHANGE, NULL);
  ArduinoCloud.addProperty(led_switch, READWRITE, ON_CHANGE, onLedSwitchChange);

}

WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);
```

At the end of the file **„thingProperties.h“** you see four variables,

2 analog
- *humi, READ, ON_CHANGE, NULL*
- *temp, READ, ON_CHANGE, NULL*

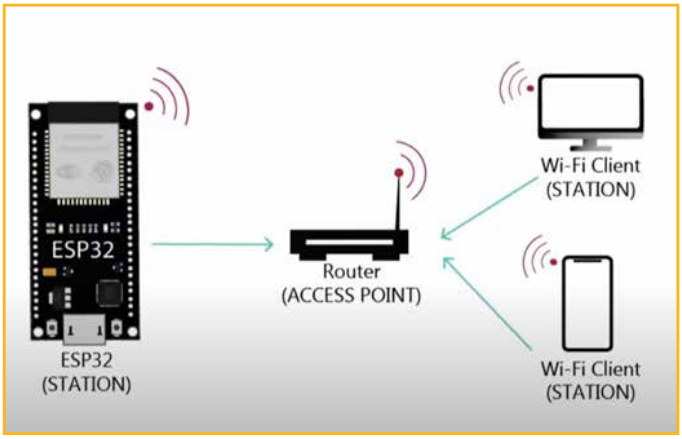
2 digital
- *led_switch, READWRITE, ON_CHANGE, onLedSwitchChange*
- *pump_switch, READWRITE, ON_CHANGE, onPumpSwitchChange*



The illustration clearly shows that a logical connection of the variables to the Sketch and the network is only established via the **".h file".** During compilation, security-relevant and programme-specific information flow together, allowing data to be exchanged via the cloud.
It is therefore impossible for the programmer to create a device that is able to send data to different end devices (computer, mobile phone, tablet, ...) without access to the cloud and without the computing power of the Arduino Cloud Server.

2.3.8 Setting up your own web server

Of course, it is possible to program the ESP boards directly from the IDE editor, but without a web server set up specifically for this purpose, no data can then be exchanged. The device then behaves like a normal Arduino board with an optional data interface (Wi-Fi, Bluetooth, UART, I2C, ...).



More information on this topic can be found under these tutorials:

https://www.youtube.com/watch?v=xPIN_Tk3VLQ

https://www.youtube.com/watch?v=z-I-r3PX2IU

https://www.youtube.com/watch?v=_GohoygyCNE

Warning

Don‘t forget, however, that this approach is far less secure against external attacks than using the Arduino Cloud. There is a theoretical risk that the network and the devices it contains can be taken over and used for criminal purposes.

2.3.9 Handling variables

The values of the variables are stored in different places when using IoT devices and must be regularly synchronised. The normal case is that the microcontroller sends its current values to the cloud and these values are then promptly passed on to other devices or the output units such as the dashboard.

You do not have to explicitly programme the reaction to changes to local variables in the Sketch. Each run of **loop()** performs a check of the states and automatically reports changes to the server:

```
55 void loop() {
56   ArduinoCloud.update();
57   // Your code here
58   humi = dht_sensor.readHumidity();
59   temp = dht_sensor.readTemperature();
60
```

In addition, there are **„onChange“** commands that inform the Sketch about changes to the system state coming from the cloud:

```
91 /*
92   Since PumpSwitch is READ_WRITE variable, onPumpSwitchChange() is
93   executed every time a new value is received from IoT Cloud.
94 */
95 void onPumpSwitchChange()  {
96   // Add your code here to act upon PumpSwitch change
97   if(pump_switch){
98     digitalWrite(PUMP_PIN, HIGH);
99   }else{
100    digitalWrite(PUMP_PIN, LOW);
101  }
102  }
```

Enough theory, let’s get started!

Follow the instructions carefully and try to understand them and their meaning. This is the only way to access this new technology in the shortest possible time!

2.4 STRUCTURE AND SECURITY OF THE ARDUINO CLOUD

Learn how to send data between an ESP32 / ESP8266 development board and the Arduino Cloud IoT.

Introduction

In this tutorial, we will go through the necessary steps to set up an ESP32 / ESP8266 development board with the Arduino Cloud IoT. To test it out, we will send random values from the board to the cloud and set up a switch that can enable an LED on the board.

Goals

The goals of this project are:

- Send data to the cloud from the board. (random values)
- Turn ON/OFF an LED on the board through the cloud.

Hardware & Software needed

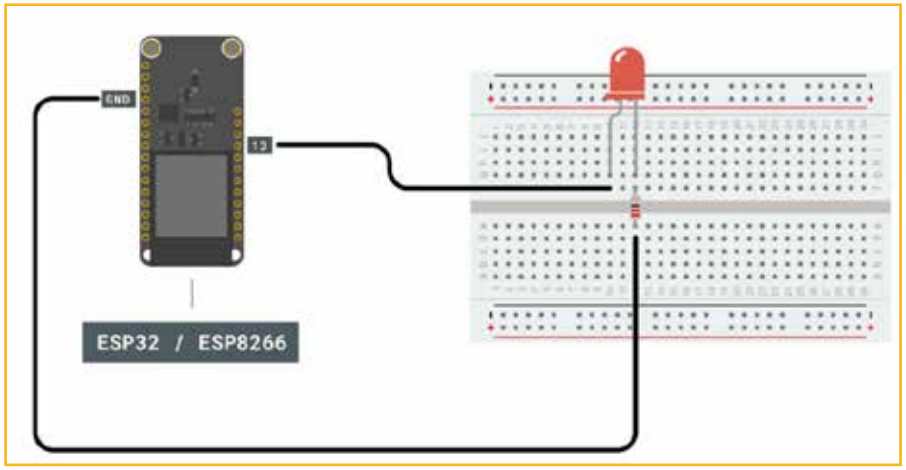
- Arduino Cloud IoT

We will also need the following components for the circuit:

- ESP32 / ESP8266 development board
- LED
- 220- or 100-Ohm resistor
- Breadboard
- Jumper Wires

Circuit

Follow the circuit below to connect the LED to the ESP32 / ESP8266 board.



Make sure you are addressing the correct PIN.

Pin	Code	Arduino alias	Pin	Code	Arduino alias
A0	A0	A0	D6	GPIO 12	12
D0	GPIO 16	16	D7	GPIO 13	13
D1	GPIO 5	5	D8	GPIO 15	15
D2	GPIO 4	4	SD2	GPIO 9	9
D3	GPIO 0	0	SD3	GPIO 10	10
D4	GPIO 2	2	RX	GPIO 3	3
D5	GPIO 14	14	TX	GPIO 1	1

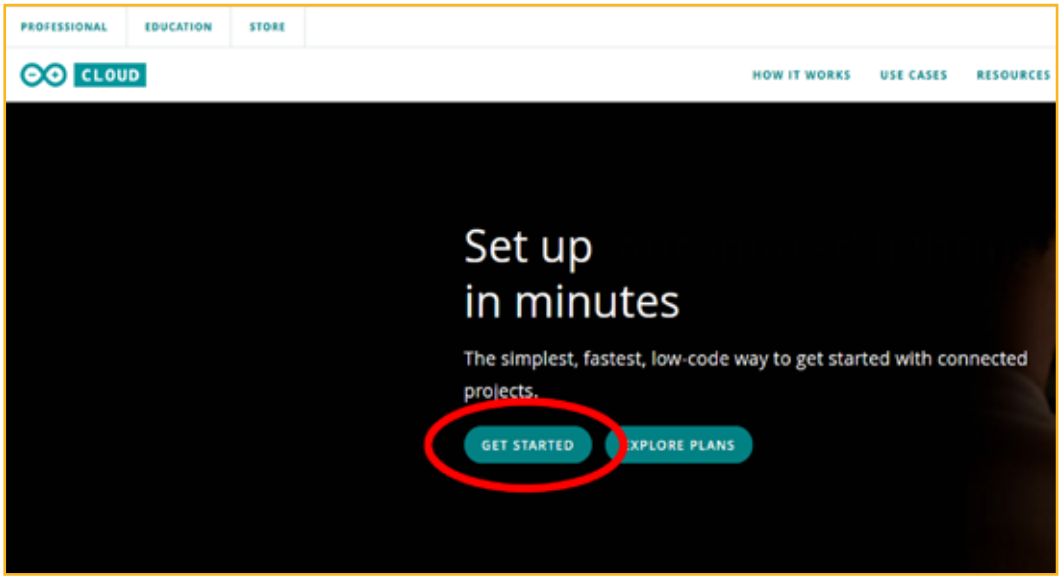
Remember

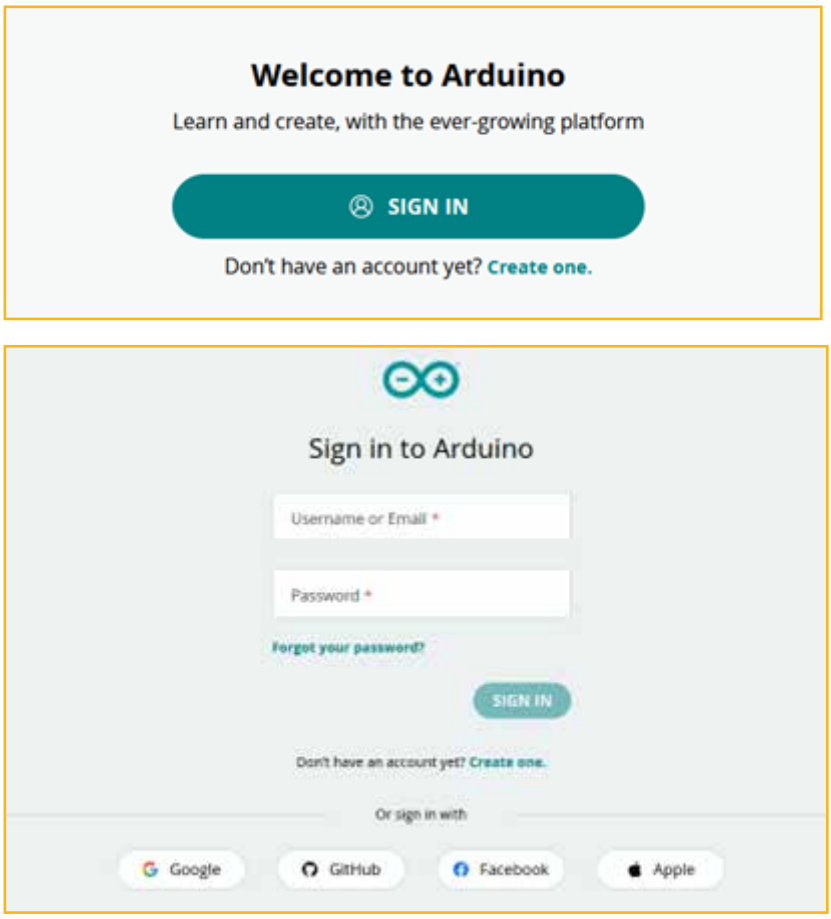
ESP boards have a different PIN designation than the Arduino boards.

Create an account or use our Arduino account - Get started with Arduino-Cloud

https://cloud.arduino.cc/

Sign in or create an account to get started.

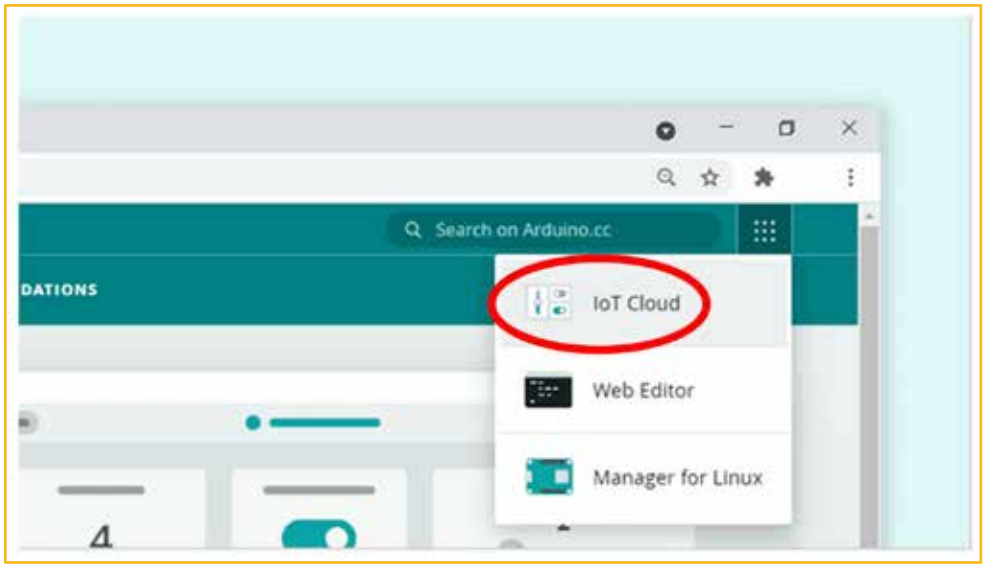




Setting up the Arduino Cloud IoT

To get started, we need to go to the Arduino Cloud IoT. This is also accessible through the menu at the top right.

Navigate to the cloud.



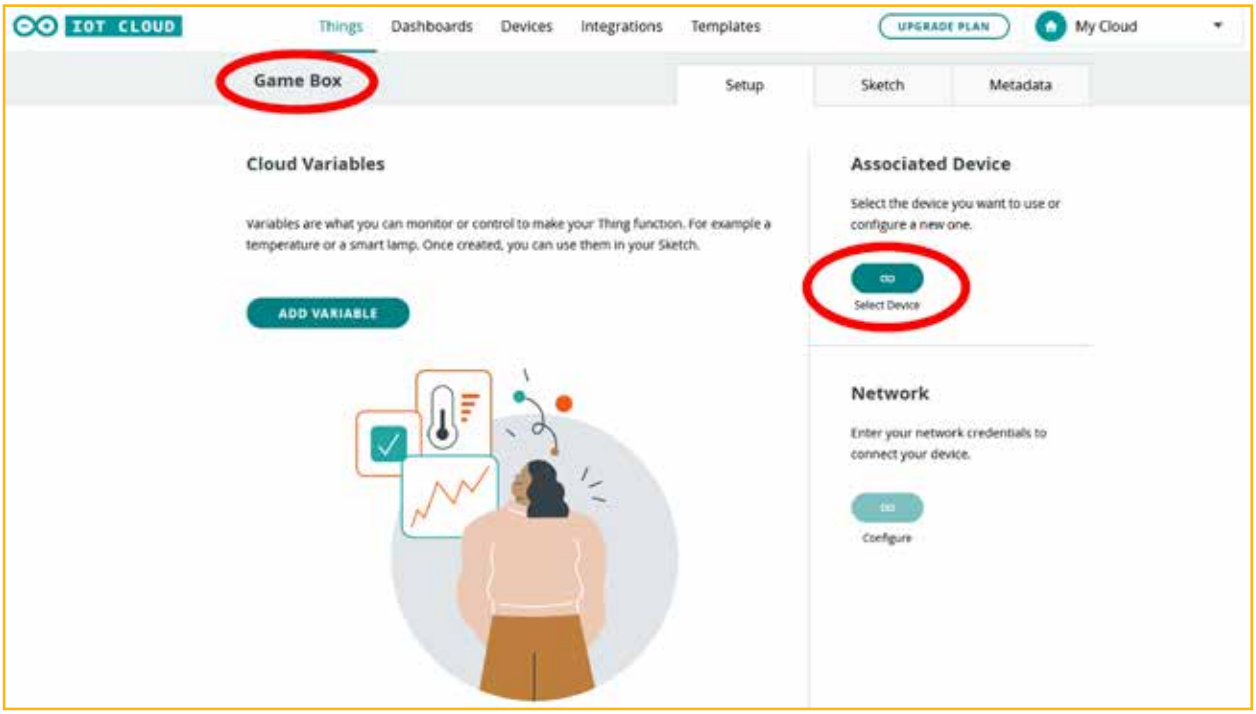
Steps to the first project

Step 1: Creating a Thing

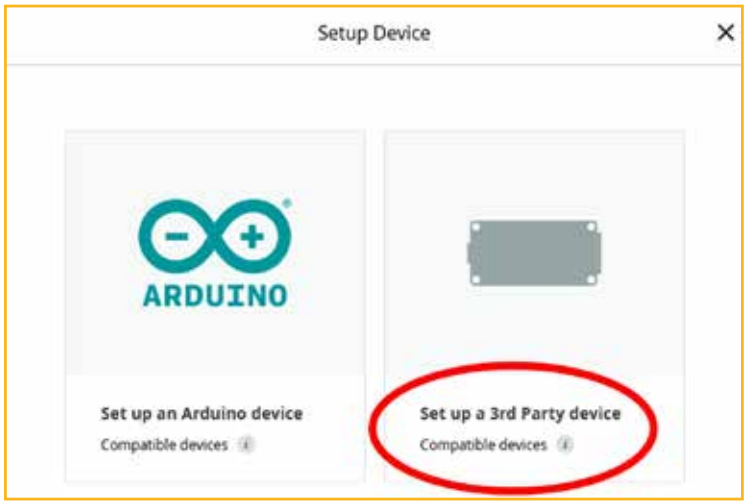
The first step is to create a Thing.

This is done by navigating to the "**Things**" tab. Then, we need to create a new Thing, by clicking on the "**Create Thing**" button. We can then rename our Thing something appropriate to what we are doing, in this case we simply chose "**Game Box**".

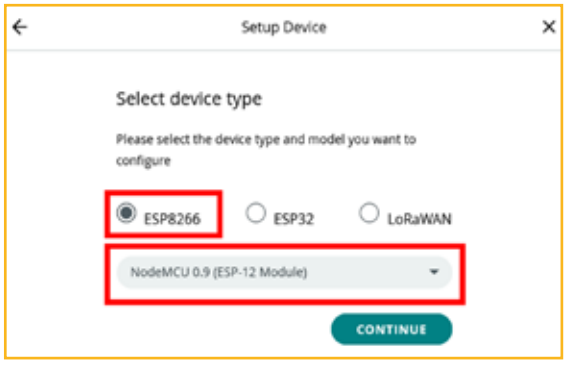
Now, we need to link our device with our Thing. This is done by clicking on the button in the "**Device**" section.



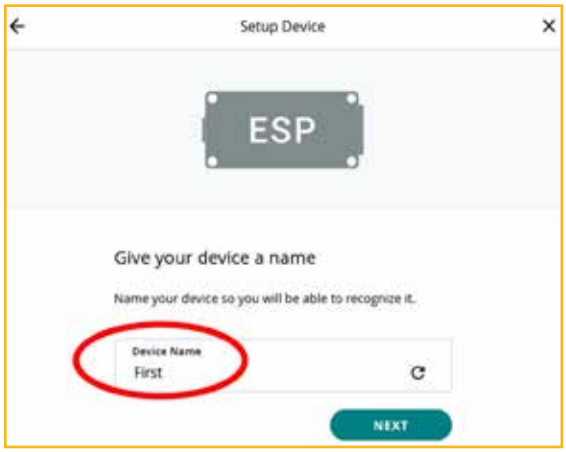
This will open a window. Choose: "**Set up a 3rd Party device**".



We are currently using the ESP8266 in the model version NodeMCU 0.9 (ESP-12 Module).

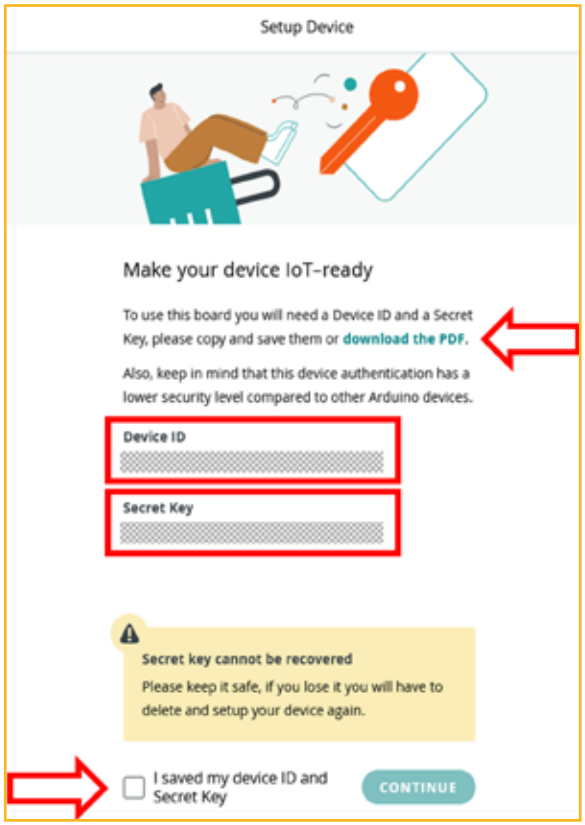


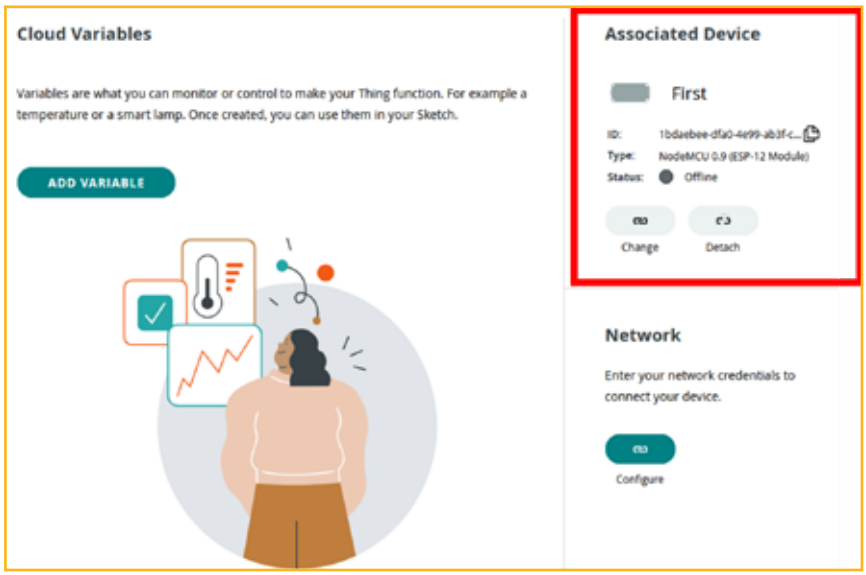
Finally, it is possible to assign a catchy name. We decided on the name „**First**“.



Please note that the secret key cannot be recovered, so make sure you note it down. You can also “**download a PDF**“ with the information. When you have saved it, tick the box at the bottom, and click on “**Continue**“.

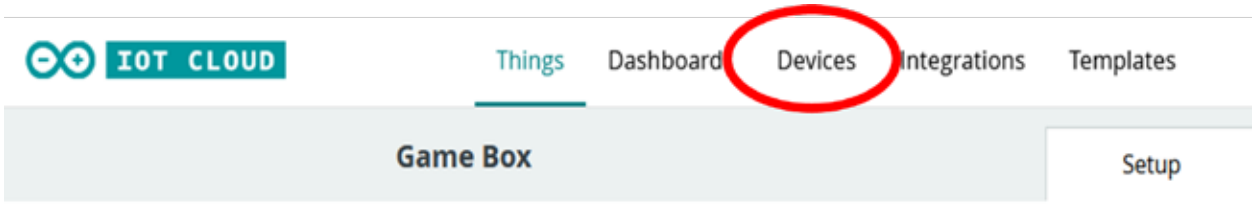
If you have done everything correctly, the linked device appears in the „**Associated Device**“ area. Initially, however, it will have the status „**Offline**“.





Further way to set up a device

You can use this way especially if you want to assign another or a new device to a Thing.



1. Once in the cloud, click on the „**Devices**“ tab. Then, click on the “**Add device**“ button.
2. Then, click on “**Set up a 3rd party device**“.
3. Now we need to select what board we are using. Select either ESP32 or ESP8266, and then choose the board from the drop-down menu.

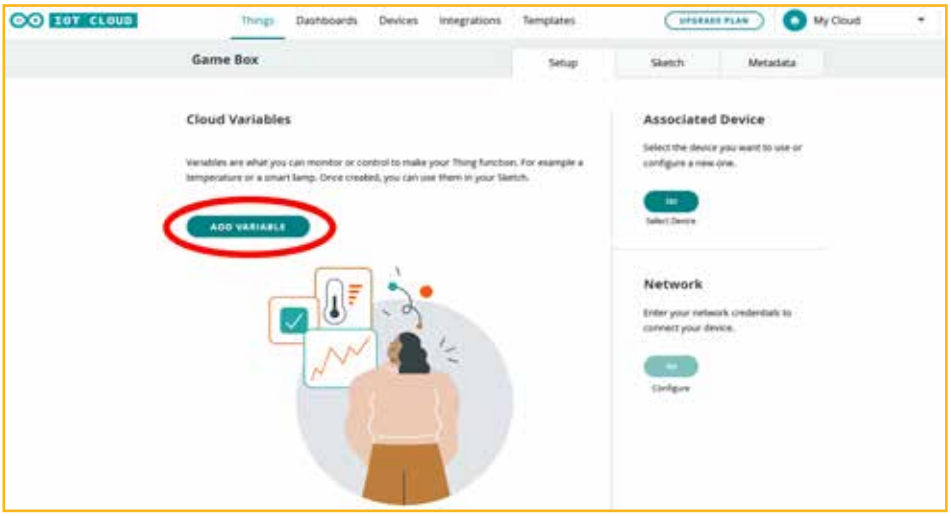
   We are currently using the “**ESP8266 – model NodeMCU 0.9 (ESP-12 Module)**“.
4. Now the board needs a name. We choose “**First**”, but you can have a lot more imagination than that!
5. You will now receive your “**Device ID**“ and “**Secret key**“. Please note that the secret key cannot be recovered, so make sure you note it down. You can also “**download a PDF**“ with the information.

   When you have saved it, tick the box at the bottom, and click on “**Continue**“.

You have now configured your ESP8266 board with the Arduino Cloud IoT. Your device will now appear in the list of devices.

Step 2: Add variables to the Thing

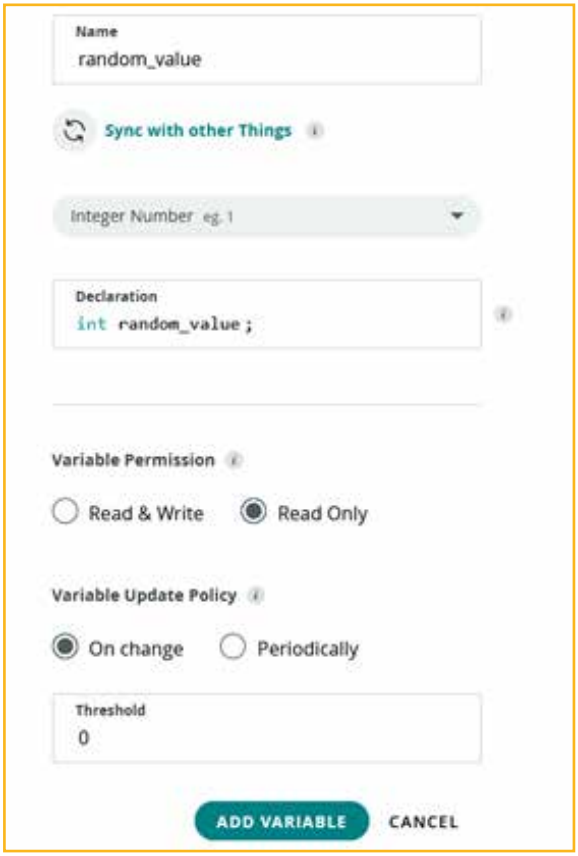
Once the device is linked, we need to create two variables: **random_value** and **led_switch**.



Click on the "**Add variable**" button. This will open a window where you need to fill in variable information. Let's create the "**random_value**" first.

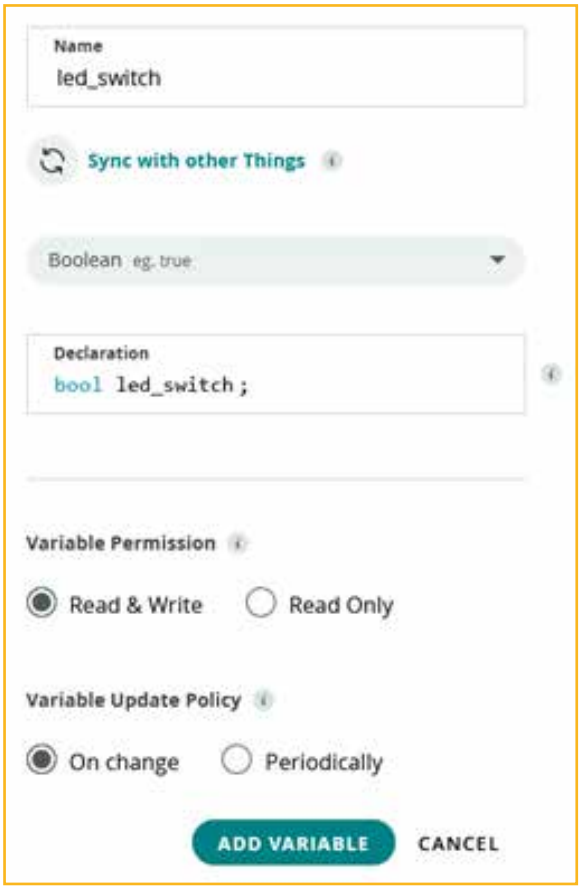
The data type is **int**, permission is **read only** and update policy is **on change**.

Once done, click on the "**Add variable**" button.

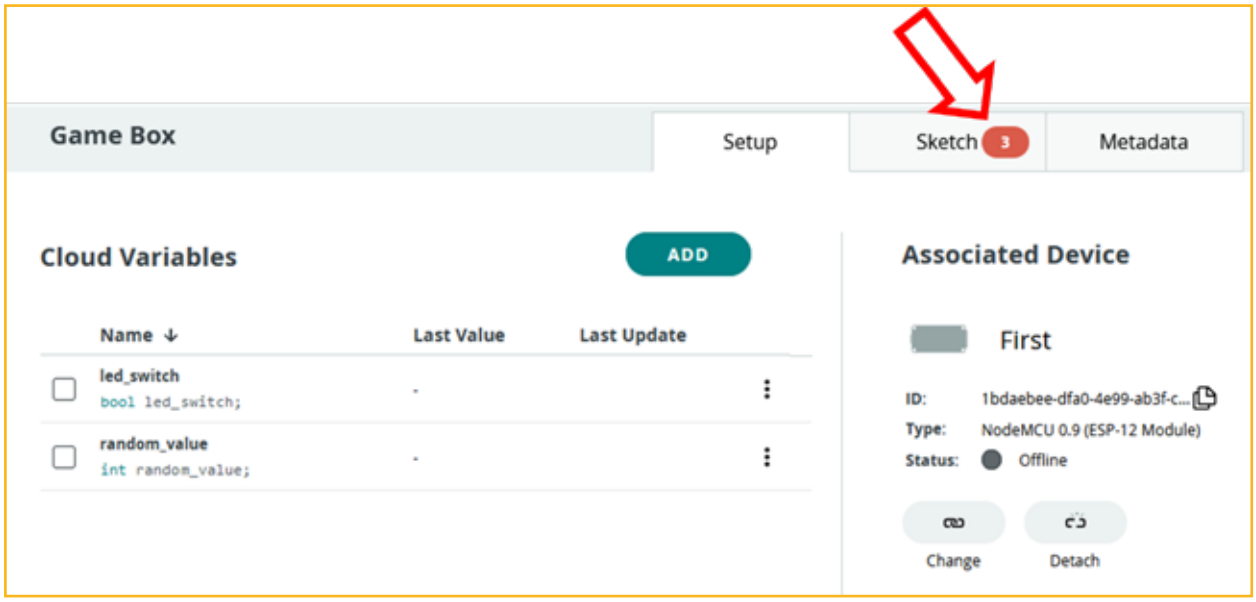


Now, let's also add the "**led_switch**" variable. The data type for this variable is **boolean**, the permission is **read & write**, and update policy is **on change**.

Once done, click on the "**Add variable**" button.

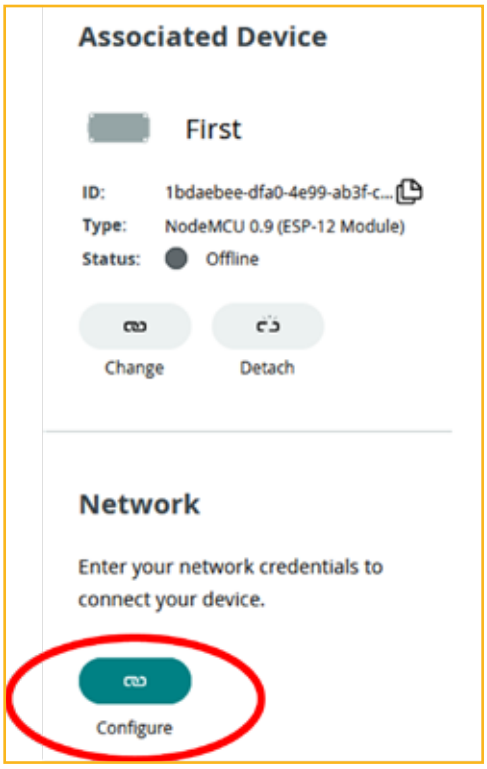


With each change in this area, the number next to the „Sketch" tab changes. In the current screenshot, the number „3" means that three changes have been made to the sketch since the start. In concrete terms, this means that the system generates or changes a sketch during the changes.



Step 2: Add variables to the Thing

Now, we need to enter the credentials for our network, plus the secret key generated during the device configuration. This is necessary to connect our device to a Wifi-Network.



First, click on the button in the "**Network Section**". Open the credentials tab.

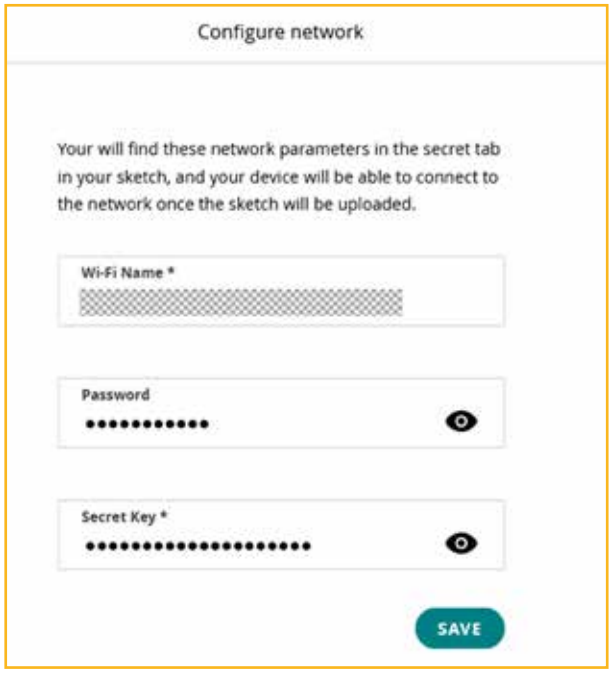
Then, enter the credentials

- network name
- network password and
- secret key.

for the used device.

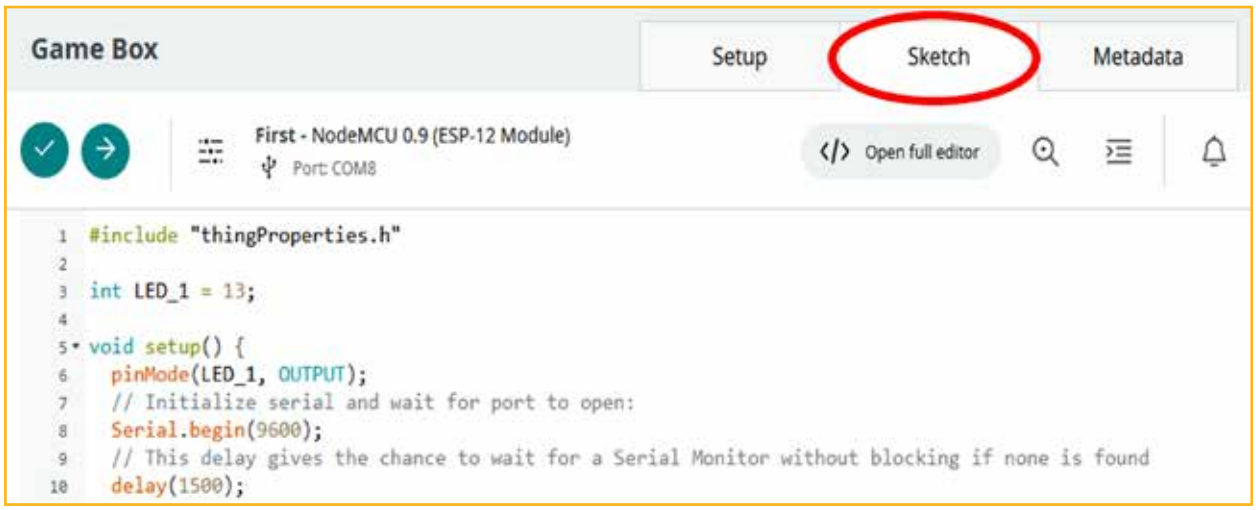
Click „**Save**" when finished.

Here it is also possible to set the access to another Wi-Fi network at any time.



Step 4: Programming the board

The next step is to program the board. To do so, we need to go to the „**Sketch**“ tab.



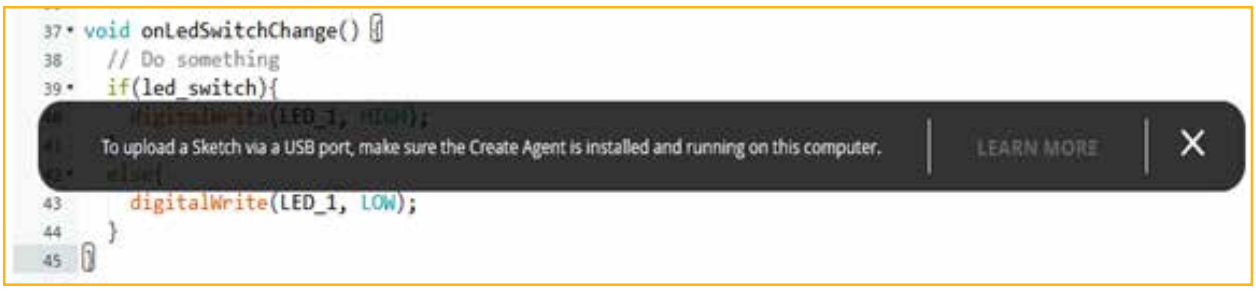
After opening, you will see that the server has already created a programme. However, this is only a skeleton programme that only ensures the connectivity of the device with the cloud. The actual programme still has to be created.

Please note!

When using the Arduino Cloud, it is not possible to program the board directly via the IDE software.
The Sketch is written in an online editor, translated into machine code by the server and loaded into the board connected to the USB port via the computer.
Also make sure that the necessary USB driver is installed. If there are any problems, you can find more information at the following link:
https://support.arduino.cc/hc/en-us/articles/4412955149586-If-your-board-does-not-appear-on-a-port-in-Arduino-IDE

In order for the transmission to the IoT device to work, it is essential to install and start the Create Agent app on the computer.

If you see this notice, you must make sure that the Arduino Create agent is installed and started.



Check your taskbar to see if the Arduino Create agent is working properly.

Installing and using the Arduino Create Agent

The **Arduino Create Agent** bridges the gap between your browser and your boards, allowing users to deploy their sketches to their board using the Arduino Web Editor. The Arduino Create Agent runs a web server bound to localhost that receives requests from the Arduino Web Editor to operate on the boards.

The security of such communication is ensured by cryptographically signing the commands that are sent to the Arduino Create Agent. Whenever the Arduino Web Editor wants to send a command to the Arduino Create Agent, it cryptographically signs the command and appends the signature to the request sent to the agent. The agent verifies the signature and, if it is correct, accepts and executes the command.

To install and use the „**Create Agent App**“, see the following instructions:
https://create.arduino.cc/getting-started/plugin/welcome

Now it is necessary to adapt the sketch to your needs.

You have to complete the code prepared by the system. To do this, type in the sketch „Game_Box“ correctly. You can leave out the comments if you want.

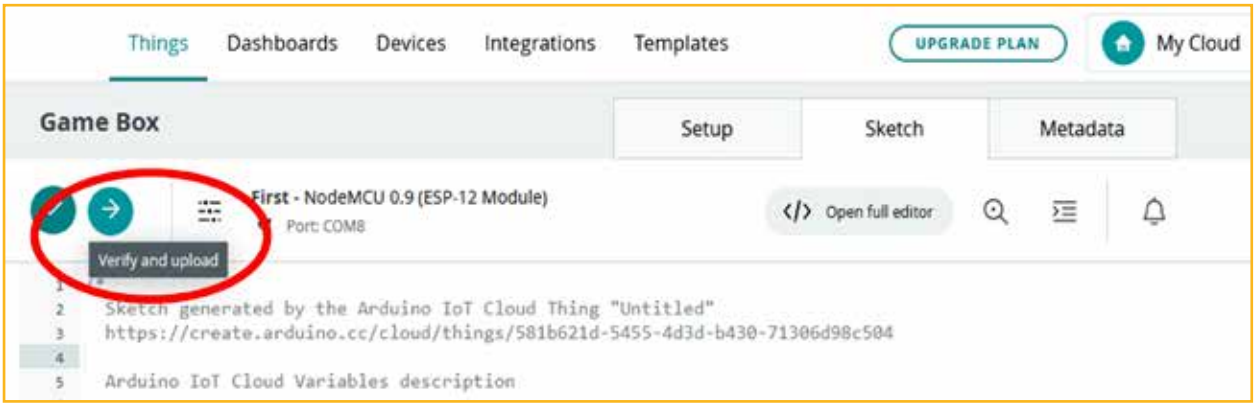
You can also open the sktech in the IDE editor and copy and paste it to the cloud. Make sure that all lines have been transferred correctly.

Here is the sketch you can use. With it, you can turn on an LED and read the random numbers from 1 to 6 (like a dice) generated by the IoT device.

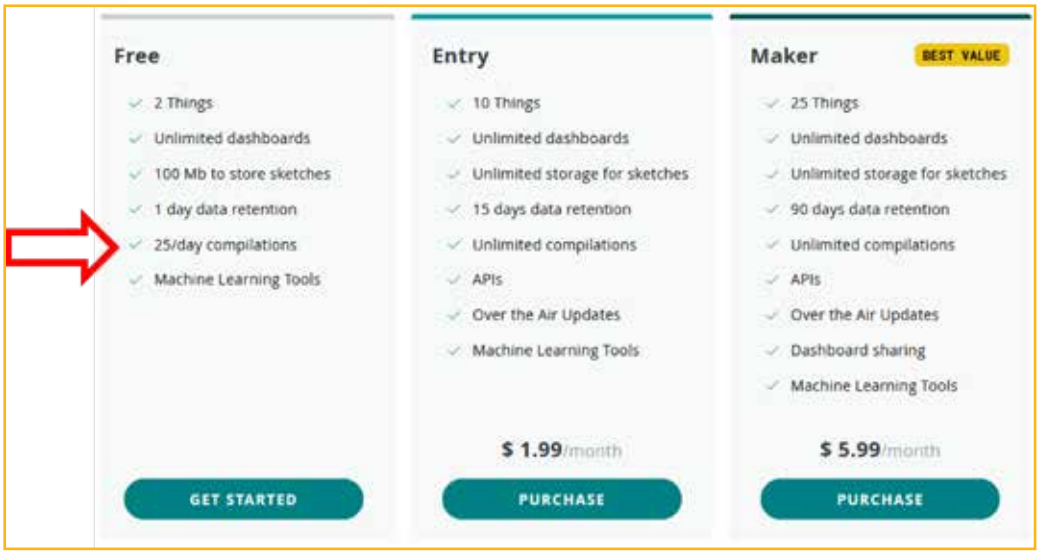
```
#include "thingProperties.h"

int LED_1 = 13;

void setup() {
  pinMode(LED_1, OUTPUT);
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);

  // Defined in thingProperties.h
  initProperties();

  /// Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
     The following function allows you to obtain more information
     related to the state of network and IoT Cloud connection and errors
     the higher number the more granular information you'll get.
     The default is 0 (only errors).
     Maximum is 4
  */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  // Your code here
  random_value = random(1, 6);
  delay(1500);
}


void onLedSwitchChange() {
  // Do something
  if (led_switch) {
    digitalWrite(LED_1, HIGH);
  }
  else {
    digitalWrite(LED_1, LOW);
  }
}
/*
  Since Asd is READ_WRITE variable, onAsdChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onAsdChange()  {
  // Add your code here to act upon Asd change
}
```

01_IoT_First.ino

When you are done, the next step is to compile and upload the sketch.

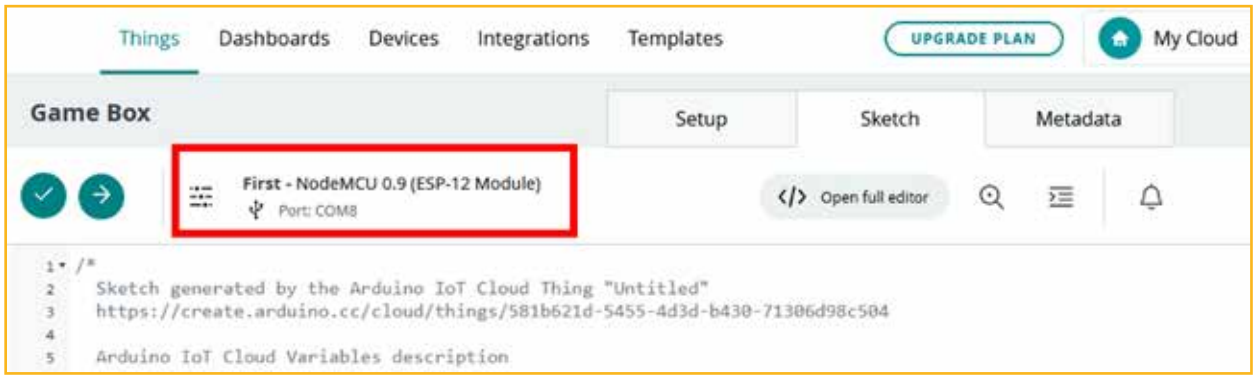


Keep in mind that the free version only allows a limited number of compilations to be created per day.



Status: 01.08.2023

Also make sure that the Arduino Create Agent is started, and the correct board is connected to the computer via USB cable. Here you can see if this is the case.

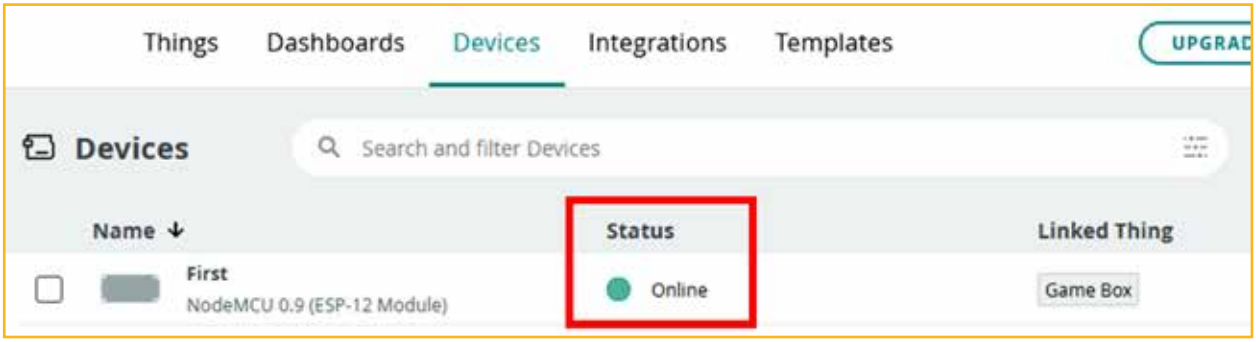


```
32   random_value = random(1, 6);
33   delay(1500);
```

```
Success: Done Uploading Game_Box_apr21a
closing bootloader
flush start
setting serial port timeouts to 1 ms
setting serial port timeouts to 1000 ms
flush complete
Game_Box_apr21a uploaded successfully on board NodeMCU 0.9 (ESP-12 Module) (COM8)
```

If everything has been done correctly and the upload is complete, the board should go online.

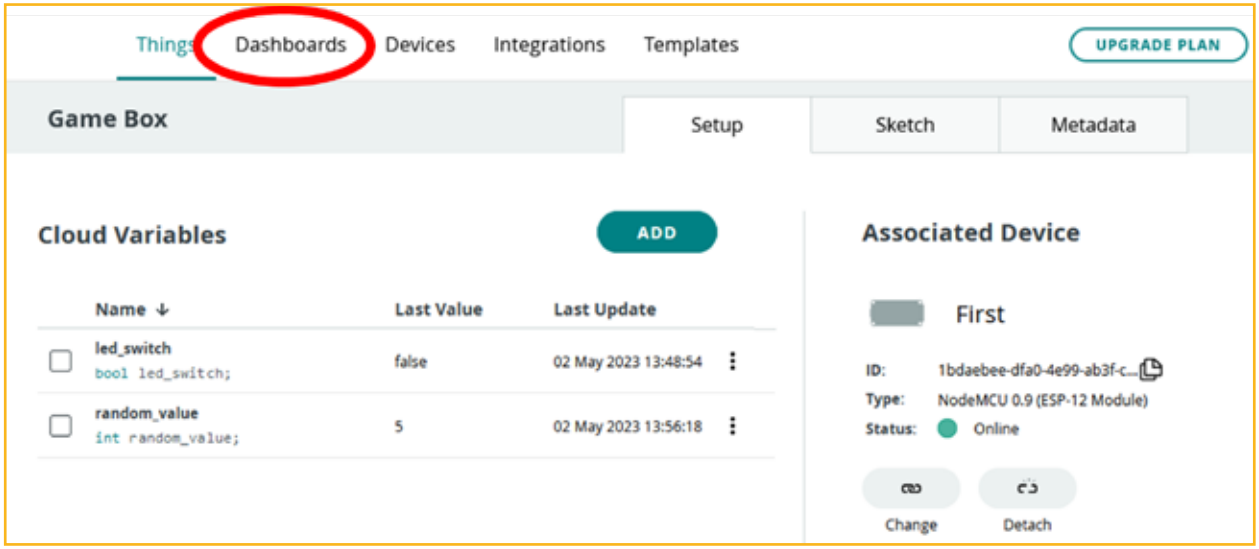
Note that it may take some time for the browser display to switch from „Offline“ to „Online“.



Step 5: Creating a dashboard

Now that the sketch is running on the ESP32 / ESP8266 board, we can move on to the final step, which is creating a dashboard.

First, navigate to the „**Dashboards**“ tab.



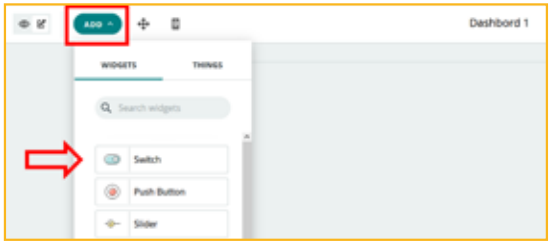
Then click on the „**Create**“ button.

We choose the name „**Dashboard 1**“ for our first dashboard.

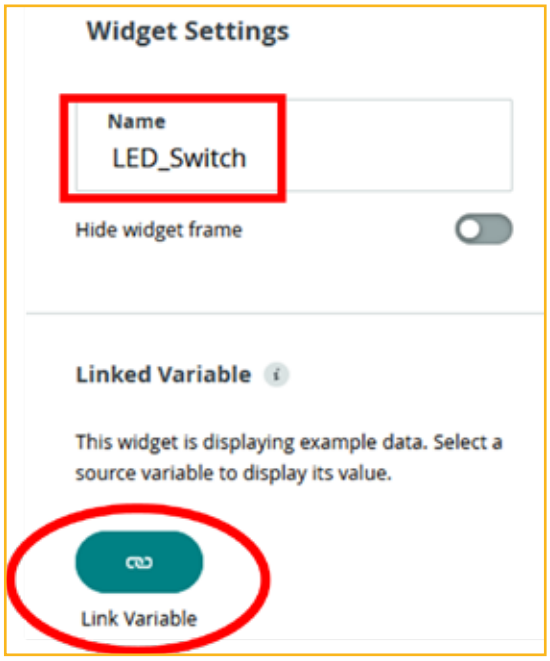
Now we add a widget to turn on the LED. In addition, the status of the LED is to be displayed.

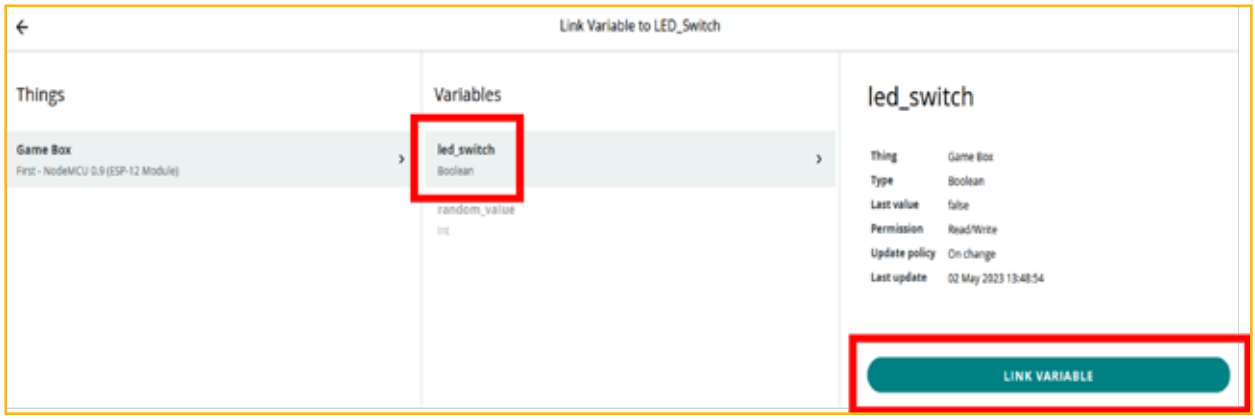
Click on the „**ADD**“ button.

To switch on the LED, we select the „**Switch**“ widget.



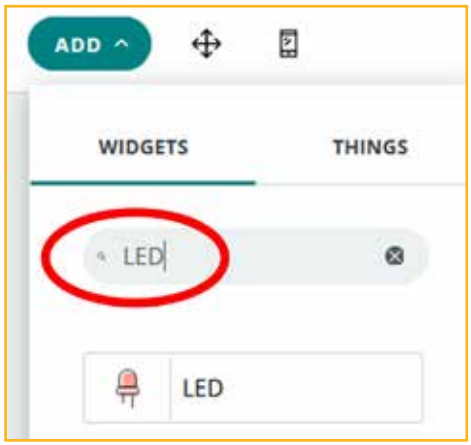
Change the name to “**LED_Switch**” and link the widget to the variable “**led_switch**”.



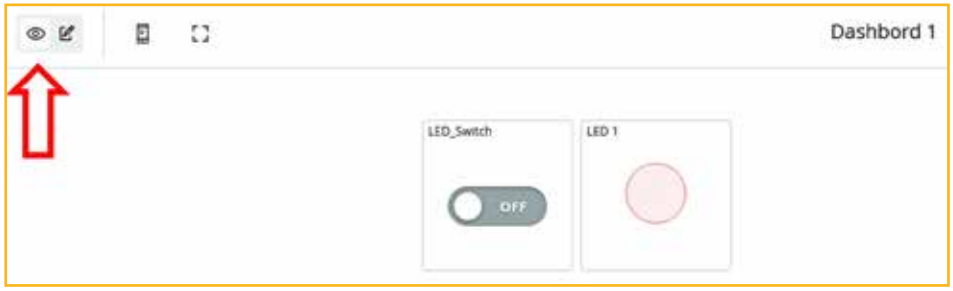


Finish adding the switch by pressing the „**Done**“ button.

To be able to see the status of the LED on the dashboard, we add another widget.
Open „**ADD**“ again and search for the widget „**LED**“.



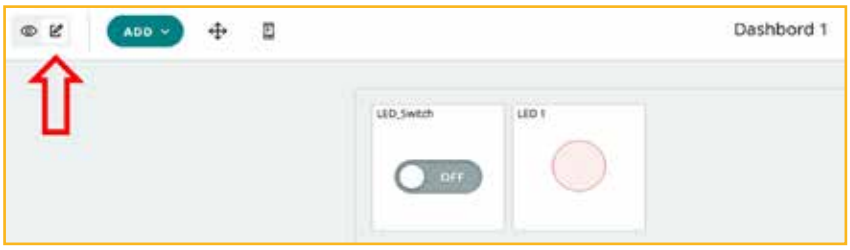
Change the name to „**LED 1**“.
Connect the widget to the variable “**led_switch**” and select the colour “**red**” as the „**LED Options**“.
Close the editing window by clicking on the **eye symbol**.



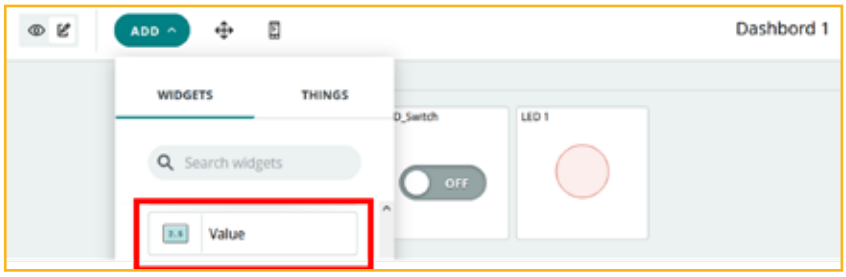
Now you can switch the **LED** on and off by pressing the “**virtual switch**”.

ON and **OFF**

Let‘s now also connect the second variable to the dashboard. To be able to do this, you have to switch to the “**editing mode**”.

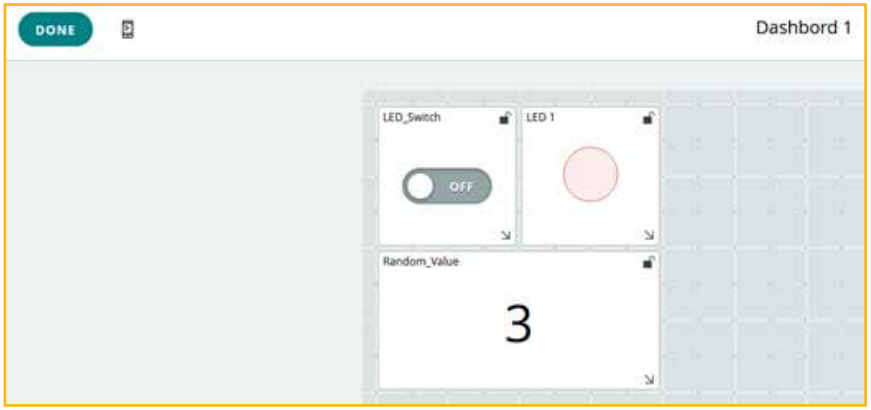


Add the Value widget to your dashboard.



Change the name to „**Random_Value**“ and connect the widget to the variable „**random_value**“. Then click on “**Done**”.

By clicking on the “**Move Icon**”, you can change the design of the dashboard to suit your preferences.

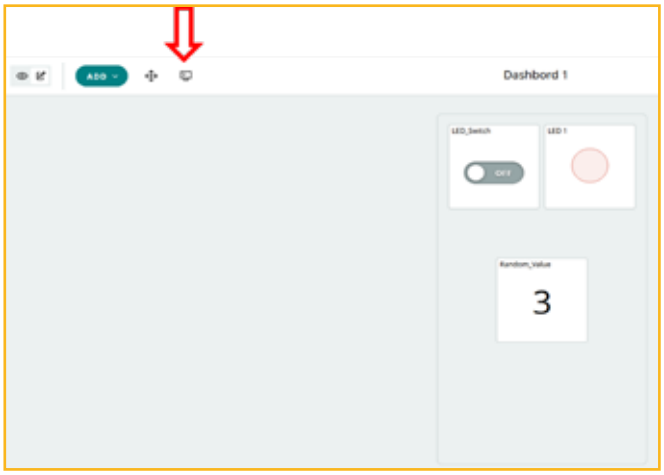


Then click on "**Done**".

The "**value field**" shows the new value of the variable „**random_value**" every few seconds. The values can range from 1 to 6, as set in the sketch, like a cube.

You can also optimise the screen for viewing on a mobile phone.

You can get back by clicking on the "**Screen**" button.



Dashboard on the mobile phone

It is also possible to display and operate the dashboard via the mobile phone.

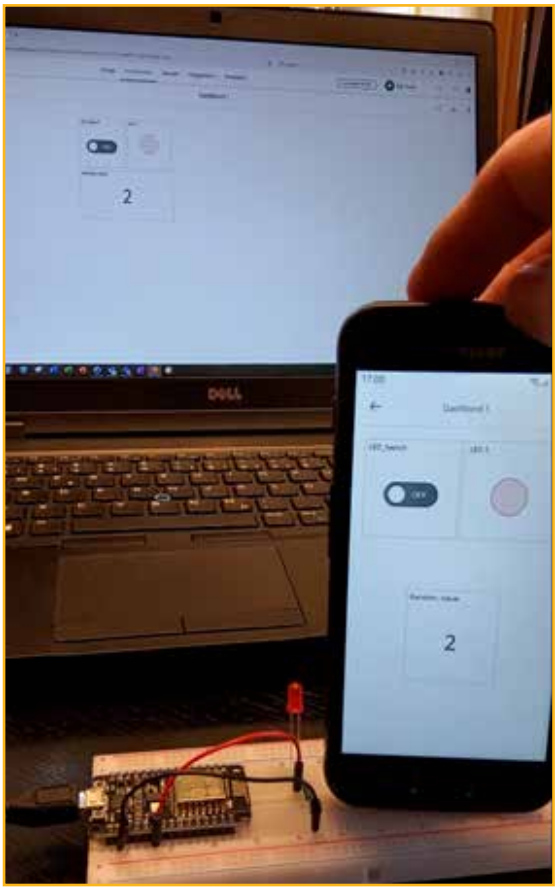
The "**Arduino IoT Cloud Remote**" App (https://docs.arduino.cc/arduino-cloud/getting-started/iot-remote-app) is the perfect companion for the Arduino IoT Cloud. Develop your IoT solution online via a desktop computer and then monitor and control your dashboards on your mobile phone with the Arduino IoT Cloud Remote App.

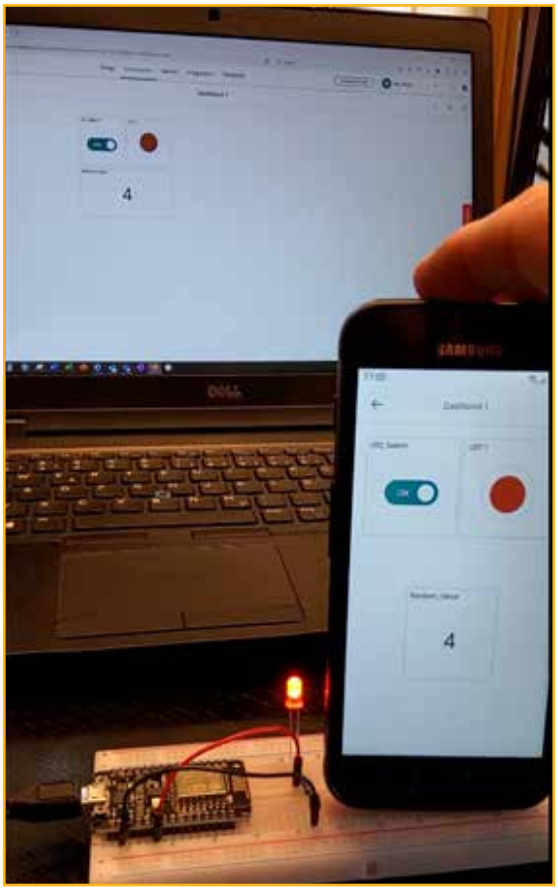
Both the iOS and Android versions are available and can be downloaded for free from the App Store and Google Play, respectively.

Also look at this information:

Control your Internet of Things projects from anywhere with the new Arduino IoT Cloud Remote app.

IoT in use





You can find more information here:

https://docs.arduino.cc/arduino-cloud/getting-started/esp-32-cloud

https://docs.arduino.cc/arduino-cloud/

https://dronebotworkshop.com/arduino-iot-cloud/

https://www.youtube.com/watch?v=xPlN_Tk3VLQ

2.5 EXAMPLE FOR A STUDENT PROJECT: IOT USED TO CONTROL A GREENHOUSE (JOINT PROJECT WITH AGRICULTURE)

The following small IoT projects are designed to prepare greenhouse control (automation technology) as a transfer project to agriculture and to demonstrate it as an activity-based lesson. There are many examples of similar projects in schools around the world.

This topic has been chosen because the construction or replication of this project can be easily reproduced by teachers and students without much effort. In addition, there are colleges in South Africa that specialise in electrical and automation engineering and agriculture.

Through this project, inter-professional and inter-college cooperation can be initiated and, where already existing, deepened. The students should learn to assess the needs, possibilities and limitations of the other professional group.

Of course, other cross-disciplinary topics are also possible, such as:

- Electrical installation in buildings - Building automation with IoT devices (Smart Home)
- Electrical installation in buildings - (Partially) self-sufficient energy supply through PV systems
- Automation technology - water supply in buildings and in agriculture
- etc.

Implementation concept for a greenhouse control system

Aim

Automated controls in greenhouses or autonomous irrigation systems are becoming increasingly important in food production. There are now many schools around the world that study these systems and implement them in student projects.

In this context, the idea of developing and implementing practical student projects in the fields of automation technology and agriculture/water management was born within the framework of TRAINME. The main focus will be on a greenhouse control system to be built step by step.

In a second step, it would be possible to use IoT technology to collect extensive sensory data and, by analysing the data, to conduct experiments in the agricultural sector with regard to growth behaviour. Students in the agricultural sector would learn to assess the importance of different growth factors in order to respond appropriately. In principle, it would also be possible to collect relevant growth data on dispersed arable land using suitable technology.

Human Resources

In order to ensure the safe and speedy implementation of the project and to guarantee its full sustainability, it is essential to generate and use extensive local know-how. Especially the technical side requires a high degree of knowledge, initiative and creativity.

It is suggested that a lead person be appointed to coordinate the project across departments. In addition, at least two teachers from the field of automation are needed for the technical implementation. These teachers should be able to prepare the required learning content in a project-oriented and practical manner and, if necessary, incorporate it into their own teaching. In the course of the project, the coordinator should contact the teachers of the agricultural department at an appropriate time.

Technical equipment

For the first step, small Arduino kits for the teachers and the students are mainly needed. In the further development stage, specific material such as soil moisture sensors, temperature and humidity sensors and other special sensors are needed. On the actuator side, motors, relays, pumps and valves will be used. Water distribution will require a reservoir, hoses and end fittings.

As the project progresses, more robust components are required to automate the greenhouse (industrial PLC, weatherproof electrical enclosures, suitable sensors and actuators, etc.). For the IoT technology, a local wifi network with stable internet access and suitable components must also be set up.

Recommended division of the subject

Due to the scope and complexity of the technical requirements, it is recommended to divide the project into 2 parts:

1. greenhouse control with conventional control system
2. an introduction to the IoT-based connectivity of automation systems with a focus on the integration into the greenhouse control system.

Below you can see a possible breakdown of topics and steps.

Step-by-step approach: Greenhouse control

Step	Theme	Learning Objective	Who?	Examples
Step 1	Arduino programming basics and building simple projects	• Getting to know microcontroller technology • Introduction to the basics of C+ programming • Building simple circuits	Electrical Students	• IPO processing with a microcontroller • Programming microcontrollers • First steps with input and output signals • Switching on an LED with a pushbutton
Step 2	Extension of knowledge to logic and the application of sensors and actuators	• Logically based decisions • if - else structures • Basics of sensor technology • Use of relays and control of motors • Basics of bus systems	Electrical Students	• Using and Adding libraries • Programming decision structures • Reactions to sensory values by controlling suitable actuators using the example of a simple temperature measurement • Display of measured values
Step 3	Sensor technology for the detection of growth factors	• Use and calibration of different sensors for the determination of growth factors • Calibration of the sensors • Indication of these values on displays • Discussion with the agricultural department	Electrical and griculture Students	• Sensors for temperature and humidity • Sensors for soil moisture • Sensors for brightness and UV radiation • Sensors for gases (CO2, O2, ...)
Step 4	Creating a simple irrigation system	• Recording the soil moisture and evaluating the irrigation requirement • Control of the appropriate actuators (pumps, valves)	Electrical and griculture Students	• Sensors for soil moisture • Design of an irrigation system • Control of pumps and/or valves
Step 5	Presentation of project results	• Writing a project report • Presentation to responsible persons and interested parties	Electrical and griculture Students	• Description of the approach • Objectives achieved • Difficulties encountered
Step 6	Transfer to an industrial PLC system	• Transfer to a Controllino-based control system	Electrical Students	• Use of a PLC system based on 24V technology

Step-by-step approach: IoT - Connectivity

Step	Theme	Learning Objective	Who?	Examples
Step 1	Introduction to building simple cloud-based systems	• The principle of connecting an IoT device to an internet-based cloud. • Getting to know basic security mechanisms	Electrical Students	• Integration of IoT devices into an internet-based cloud • Programming in the Cloud • Switching on an LED over computer and cellphone
Step 2	Extension of knowledge to the application of sensors and actuators	• Integration and evaluation of sensor data • Display of measured values on a dashboard • Use of relays and control of actuators	Electrical Students	• Integration of a simple temperature sensor and output to the dashboard (display on the computer and mobile phone) • Activation of a pump via mobile phone
Step 3	Transfer of greenhouse control to the cloud-based system	• Use and calibration of different sensors to determine the growth factors • Display of these values on on-site displays • Integration of the sensors into the cloud • Discussion with the agricultural department • Creation of a suitable dashboard	Electrical and gricultureStudents	• Sensors for temperature and humidity • Sensors for soil moisture • Sensors for brightness and UV radiation • Sensors for gases (CO_2, O_2, ...)
Step 4	Creating a simple irrigation system	• Recording the soil moisture and evaluating the irrigation requirement • Control of the appropriate actuators (pumps, valves)	Electrical and gricultureStudents	• Sensors for soil moisture • Design of an irrigation system • Control of pumps and/or valves • Linking with remote control systems

2.6 SECOND PROJECT – CLIMATE CONTROL IN A GREENHOUSE

The knowledge gained from the first project is now to be implemented in an application-related second project. Now that you know how to connect an ESP32 or ESP8266 to the Arduino Cloud IoT, you are ready to create your first practical applications. Learn how to send data between an ESP32 / ESP8266 development board and the Arduino Cloud IoT to control a greenhouse.

In this second project, students will learn the basic structure of a greenhouse control system. For a goal-oriented implementation, it is necessary to be able to familiarise oneself with agricultural and biological knowledge in addition to sensory (measurement of temperature, humidity, soil moisture, brightness, etc.), actuator and control technology skills. This is an interdisciplinary task. The person setting up and programming the automation must also acquire knowledge of the system to be automated. Otherwise the automation will not be successful.

Introduction

In this tutorial we will go through the steps necessary to use an ESP32 / ESP8266 development board to control a greenhouse.

The aim is to record various measured values such as temperature, humidity, soil moisture and brightness and to activate actions such as irrigation, ventilation and shading based on the measured values. Using a computer or mobile phone, it should be possible to view the measurement data and control the components from any location with an internet connection.

Goals

The goals of this project are:

- Send data (measured values – e.g. temperature, humidity, soil moisture and brightness) to the cloud from the IoT-board.
- Display of these measured values on different terminals.
- Triggering necessary actions (e.g. opening and closing ventilation, controlling irrigation, influencing shading) via the cloud.
- Integration of small logical functions for partial automation, e.g. of irrigation or shading.
- ...

Step-by-step procedure based on tasks

Task 1	Connect to Cloud	✅
Task 2	Switching the light on/off (using an LED)	✅
Task 3	Record temperature/humidity (and show on display)	
Task 4	Create different dashboards for different applications.	
Task 5	Realise control for a water pump and an irrigation system (manual operation)	
Task 6	Automate control for a water pump	
Task 7	Realise control for a shading and ventilation system	
Task 8	...	

Step 1 and Step 2 have already been practised in the first IoT project. It should therefore no longer be a problem to connect an IoT device (ESP32 or ESP8266) to the Arduino Cloud and set up a „Thing“. For repetition and to keep the error rate as low as possible, the steps for setting up are briefly shown again and the procedure is explained in concise form.

Hardware & Software needed

- Arduino Cloud IoT

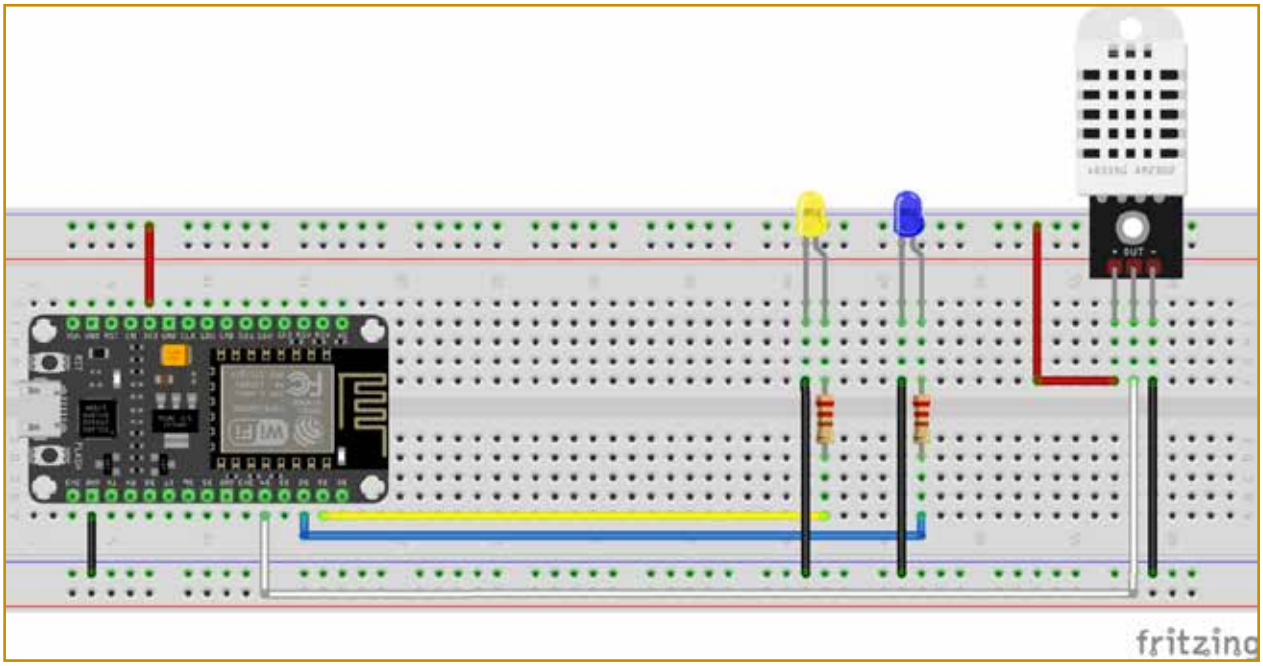
We will also need the following components for the circuit:

- ESP32 / ESP8266 development board
- 2 LED´s (e.g. yellow and blue)
- 2 resistors (220 or 100 Ohm)
- DHT22 or DHT11 (Digital-output relative humidity & temperature sensor-module)
- Breadboard
- Jumper Wires
- (Optional: Display OLED - 1.3“, 128x64, VGSS, I2C)

You need an OLED display if measured values are to be displayed directly near the IoT board. This may make sense for demonstration purposes, but it increases the programming effort.

Circuit

Build the circuit as shown.



Make sure you contact the correct PIN on the ESP board!

Yellow LED (e.g. shading): D1 – GPIO 5 - PIN Adr. 5

Blue LED (e.g. water-pump): D2 – GPIO 4 - PIN Adr. 4

DHT data transmission: D4 – GPIO 2 - PIN Adr. 2

Pin	Code	Arduino alias	Pin	Code	Arduino alias
A0	A0	A0	D6	GPIO 12	12
D0	GPIO 16	16	D7	GPIO 13	13
D1	GPIO 5	5	D8	GPIO 15	15
D2	GPIO 4	4	SD2	GPIO 9	9
D3	GPIO 0	0	SD3	GPIO 10	10
D4	GPIO 2	2	RX	GPIO 3	3
D5	GPIO 14	14	TX	GPIO 1	1

Let us now move on to the necessary steps for the task:

Step 1: Creating a Thing

Create a Thing called “Climate station”.

Add a new device called “Second” and associate the device with the Thing.

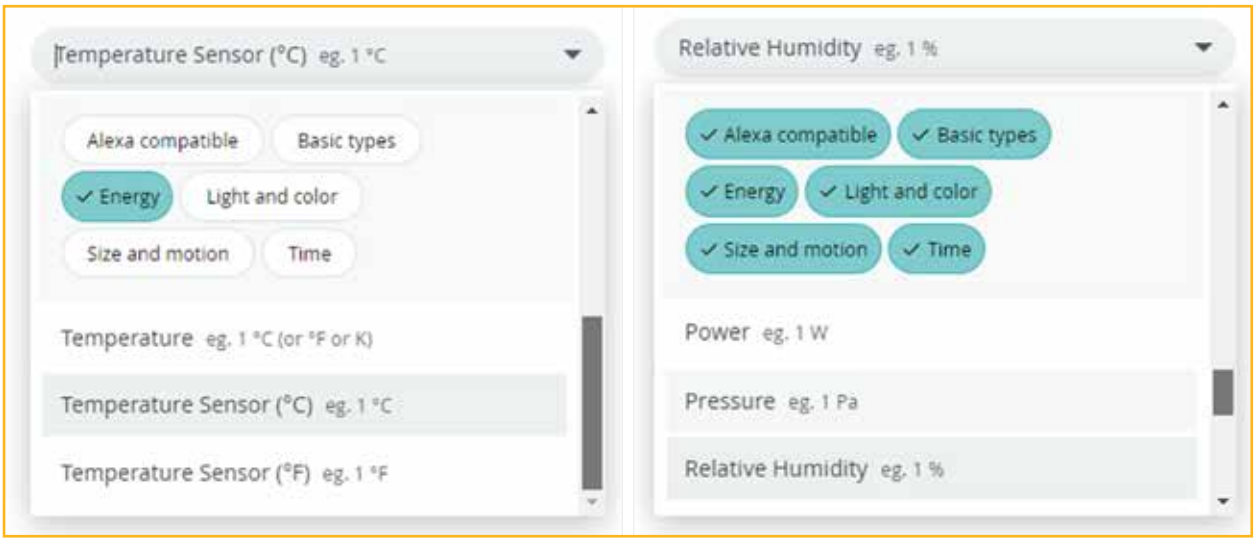
Don‘t forget to save all the necessary safety data.

Step 2: Add variables to the Thing

Add two variables for temperature and humidity:

• Called “temp” / Type: Select Energy then “Temperature Sensor (°C)” / “Read Only” / “On Change”

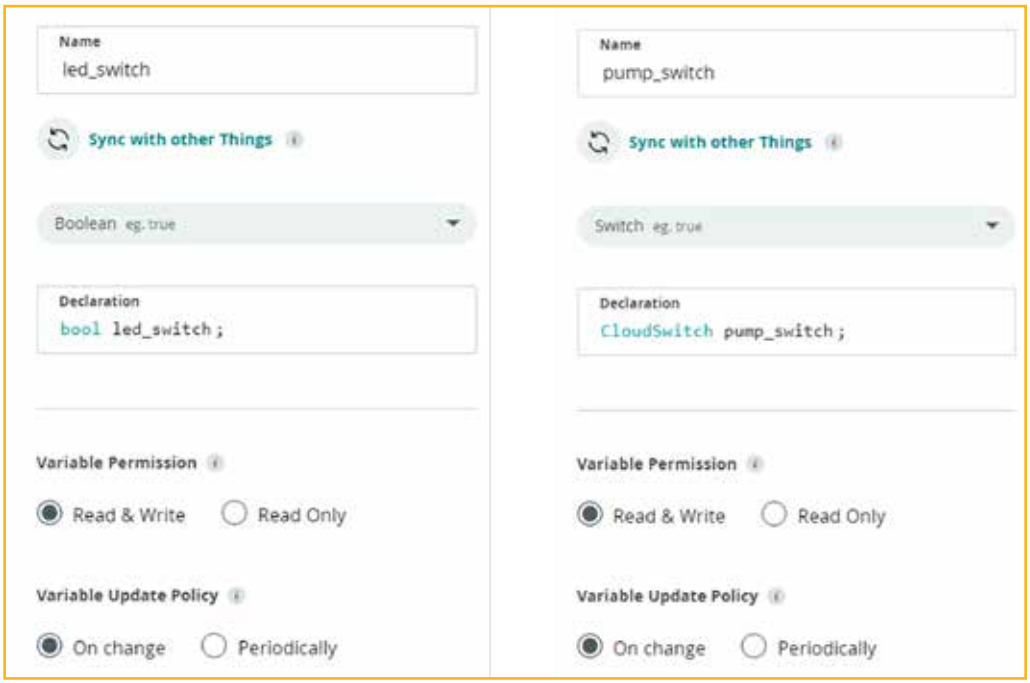
• Called “humi” / Type: Select All then “Relative Humidity eg. 1%” / “Read Only” / “On Change”



In order to be able to carry out necessary switching operations, we add two more variables.

• Called **“led_switch”** / Type: Select All then **”boolean“** or **“Switch eg. true” / “Read & Write” / “On Change”**

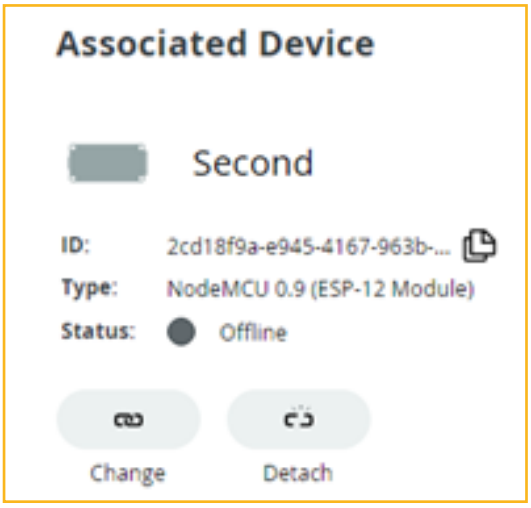
• Called **“pump_switch”** / Type: Select All then **”boolean“** or **“Switch eg. true” / “Read & Write” / “On Change”**



Overview of all variables.

Step 3: Adding credentials

Now, we need to enter the credentials for our network, plus the secret key generated during the device configuration. This is necessary to connect our device to a Wifi-Network.



First, click on the button in the **"Network Section"**.

Open the credentials tab.

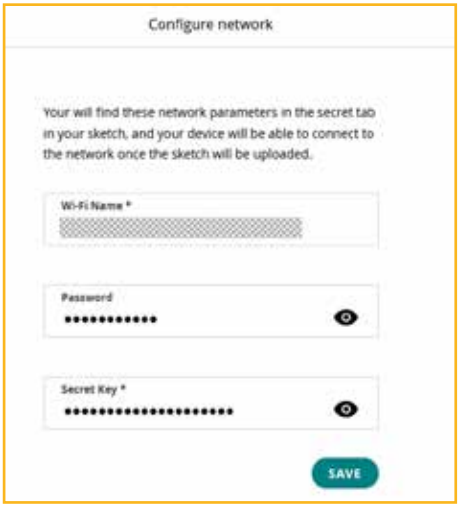
Then, enter the credentials

- network name
- network password and
- secret key.

for the used device.

Click „**Save**" when finished.

Here it is also possible to set the access to another WiFi network at any time



Step 4: Programming the board - Go to "Sketch"

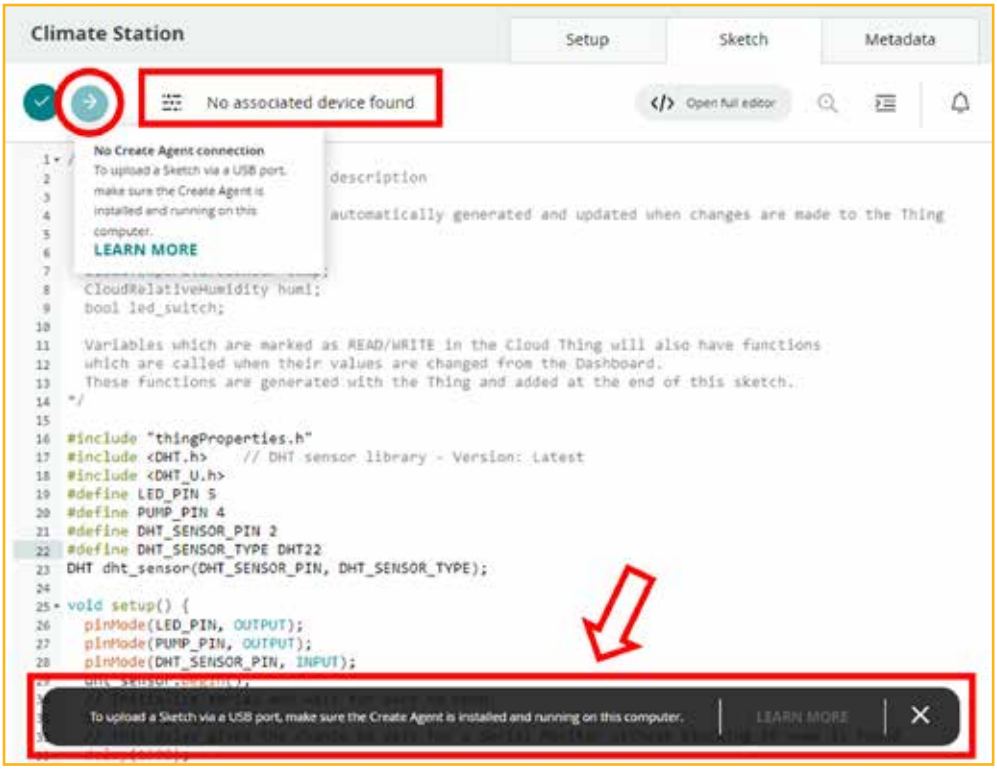
The next step is to program the board. To do so, we need to go to the „Sketch" tab.

You can adapt or enter the sketch here or open a second window for programming via the button „Open full editor". (For experts only!)

Here is the sketch you can use

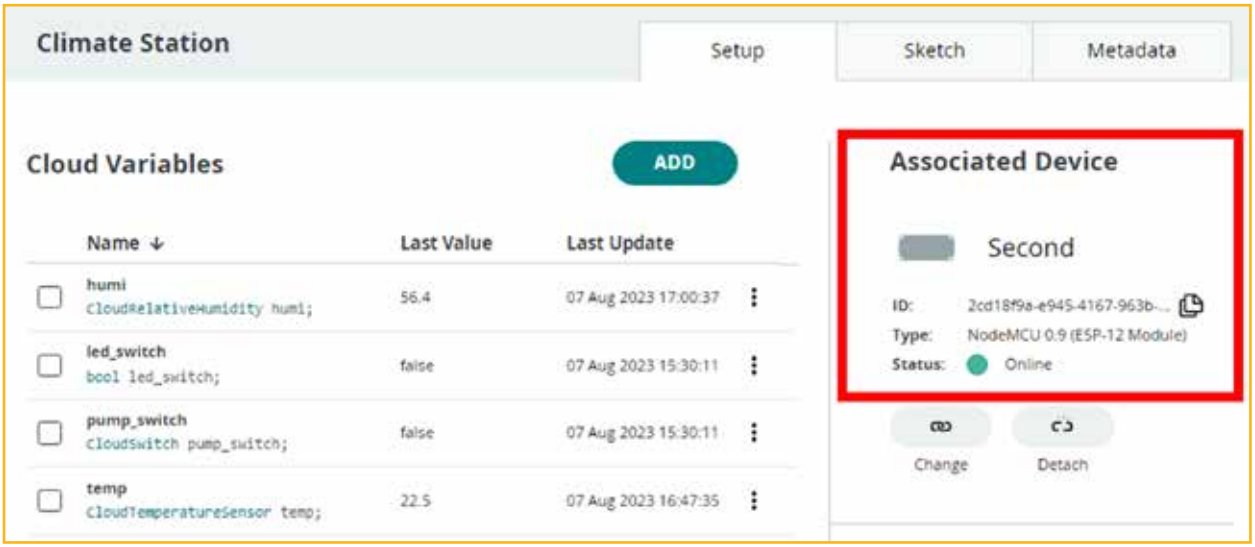
```
/*
  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the Thing

  CloudSwitch pump_switch;
  CloudTemperatureSensor temp;
  CloudRelativeHumidity humi;
  bool led_switch;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"
#include <DHT.h>    // DHT sensor library - Version: Latest
#include <DHT_U.h>
#define LED_PIN 5
#define PUMP_PIN 4
#define DHT_SENSOR_PIN 2
#define DHT_SENSOR_TYPE DHT22
DHT dht_sensor(DHT_SENSOR_PIN, DHT_SENSOR_TYPE);

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(PUMP_PIN, OUTPUT);
  pinMode(DHT_SENSOR_PIN, INPUT);
  dht_sensor.begin();
  // Initialize serial and wait for port to open:
  Serial.begin(115200);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);

  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
     The following function allows you to obtain more information
     related to the state of network and IoT Cloud connection and errors
     the higher number the more granular information you'll get.
     The default is 0 (only errors).
     Maximum is 4
  */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  // Your code here
  humi = dht_sensor.readHumidity();
  temp = dht_sensor.readTemperature();

  if(isnan(humi) || isnan(temp)){
    Serial.println("Failed to read from DHT Senor!");
  }
  else{
    Serial.print("Temperature:  ");
    Serial.print(temp);
    Serial.print("°C");
    Serial.print("  |  ");

    Serial.print("Humidity:     ");
    Serial.print(humi);
    Serial.print("%");
  }
  delay(2000); //31:34 min !!!

}

/*
  Since LedSwitch is READ_WRITE variable, onLedSwitchChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLedSwitchChange()  {
  // Add your code here to act upon LedSwitch change
  if(led_switch){
    digitalWrite(LED_PIN, HIGH);
  }else{
    digitalWrite(LED_PIN, LOW);
  }
  }

/*
  Since PumpSwitch is READ_WRITE variable, onPumpSwitchChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onPumpSwitchChange()  {
  // Add your code here to act upon PumpSwitch change
  if(pump_switch){
    digitalWrite(PUMP_PIN, HIGH);
  }else{
    digitalWrite(PUMP_PIN, LOW);
  }
  }
```

When you are done, the next step is to compile and upload the sketch.



If the message „**No associated device found**“ is displayed next to the “**Compile and Upload Button**”, then there is no connection from the computer to the device via USB cable and/or the Arduino Create Agent is not installed or started.

And when the upload is done correctly, the status display for the “**Associated Device**” switches to „**Online**“.



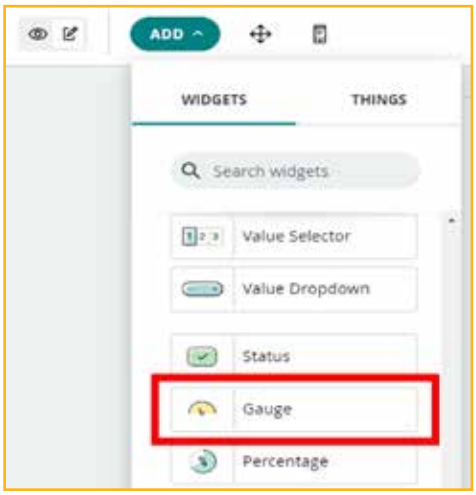
Step 5: Creating a dashboard

Now that the sketch is running on the ESP32 / ESP8266 board, we can move on to the final step, which is creating a dashboard.

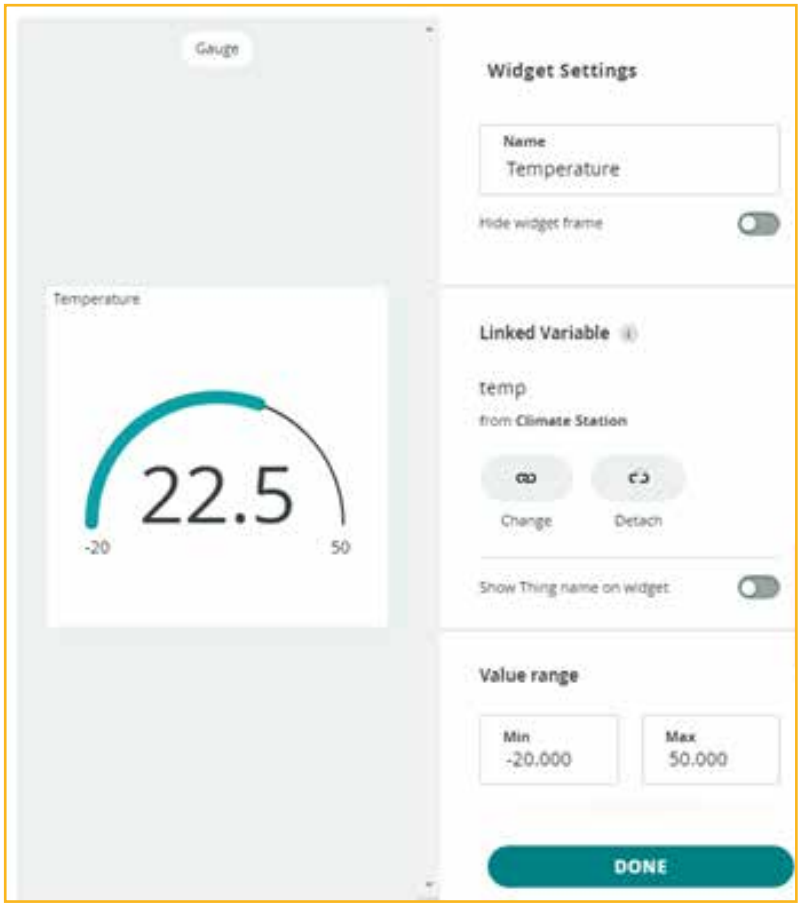
Navigate to the „**Dashboards**“ tab. Click on the „**Create**“ button.

Give the new dashboard the name “**Climate Control**”.

Click on the “**Add**” – Button and select a widget called “**Gauge**”.



Set the “Widget Settings”:



Name: “Temperature”

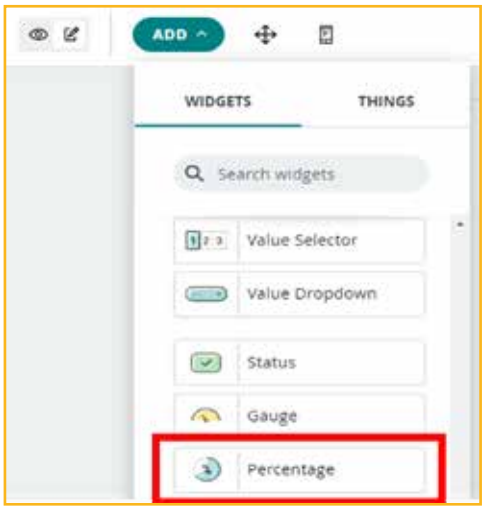
Linked Variable: “temp”

Value range: “Min -20 / Max 50”

or any you like.

Finish adding the widget by pressing the „Done“ button.

To display the humidity, select the widget "**Percentage**".



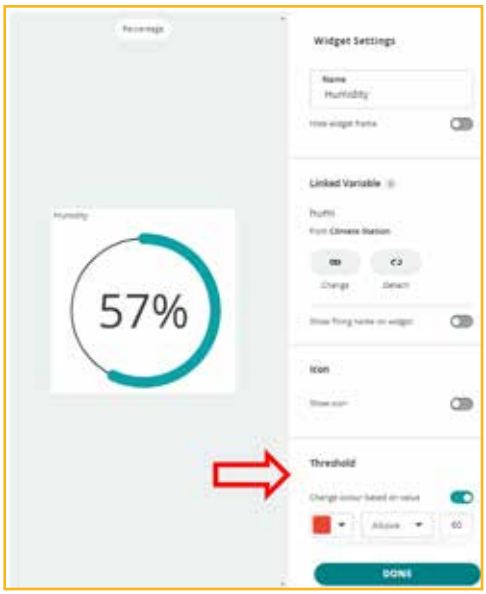
Name: "Humidity"

Linked Variable: "humi"

Threshold:

If you want the colour of the display to change above a certain threshold, switch on this function, select a colour and enter a value.

Finish adding the widget by pressing the „**Done**" button.



Add two more switches to the dashboard, as you have learned in the previous chapter.

Name the two switches "**Light Switch**" and "**Pump Switch**".

Since it is not possible to select the colours yellow and blue for LEDs in the dashboard, we use the colours red and green here.

This dashboard is optimised for use on the screen. If you want to use the functions on the mobile phone, you can also generate a mobile layout.

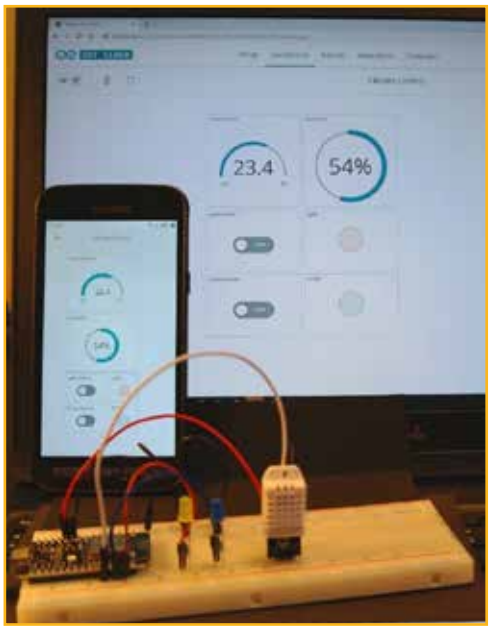
Use this button to switch between the two views.

Look at the first results

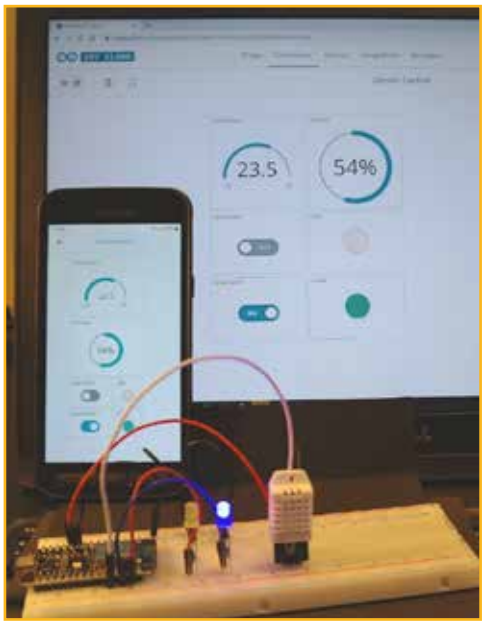
You can view the temperature and humidity both on your computer in the browser view and in the IoT app on your mobile phone.

The humidity is displayed in green because the value is still below 60%. Remember: we have set a threshold value of 60% in the „Humidity“ widget.

The two actors „Light Switch“ (yellow LED) and „Pump Switch“ (blue LED) are switched off. It would be possible to switch the respective output in each representation.



Now the temperature has risen slightly to 23.5 °C, and the pump (blue LED) has been switched on via the app or on the computer.



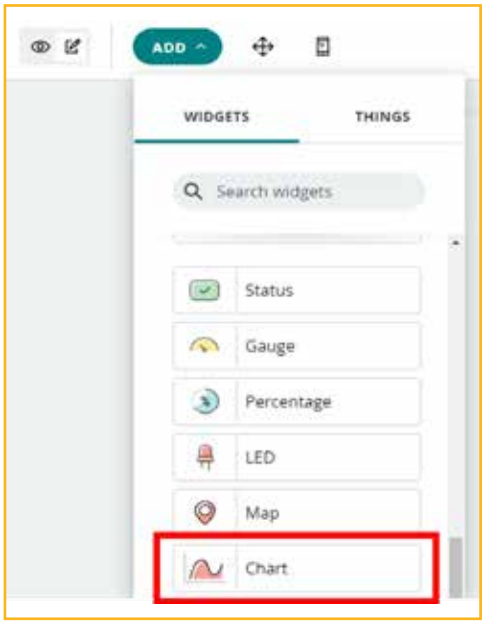
Since we have inserted a threshold value at 60% for the variable „Humidity“ in the dashboard, the colour of the display changes from green to red when this value is exceeded.

Other ways of displaying values in the dashboard

Another interesting way of presenting values is in the form of a chart. Charts can be used to present data on any topic. This is possible in the Arduino Cloud with the help of the „Chart“ widget. The charts used here show the values over time.

Create a new Dashboard called “**H&T Chart**”.

Click on the “**Add**” – Button and select a widget called “**Chart**”.



Name: “**Temperature**”

Linked Variable: “**temp**”

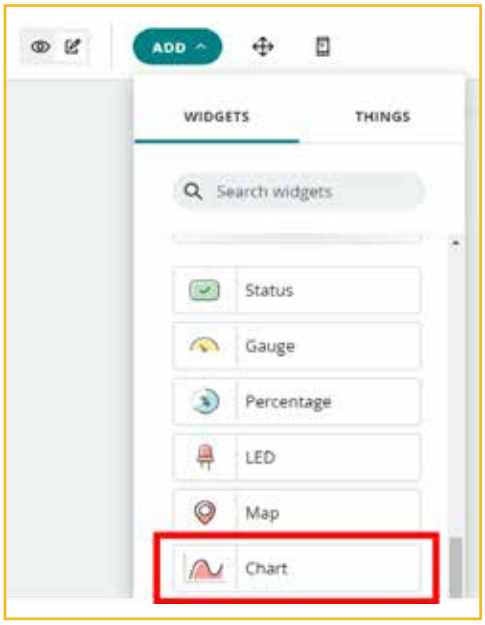
from “**Climate Station**”

Data points interpolation:

“**Spline**” or “**Line**”

-whatever you prefer.

Finish adding the widget by pressing the „**Done**“ button.

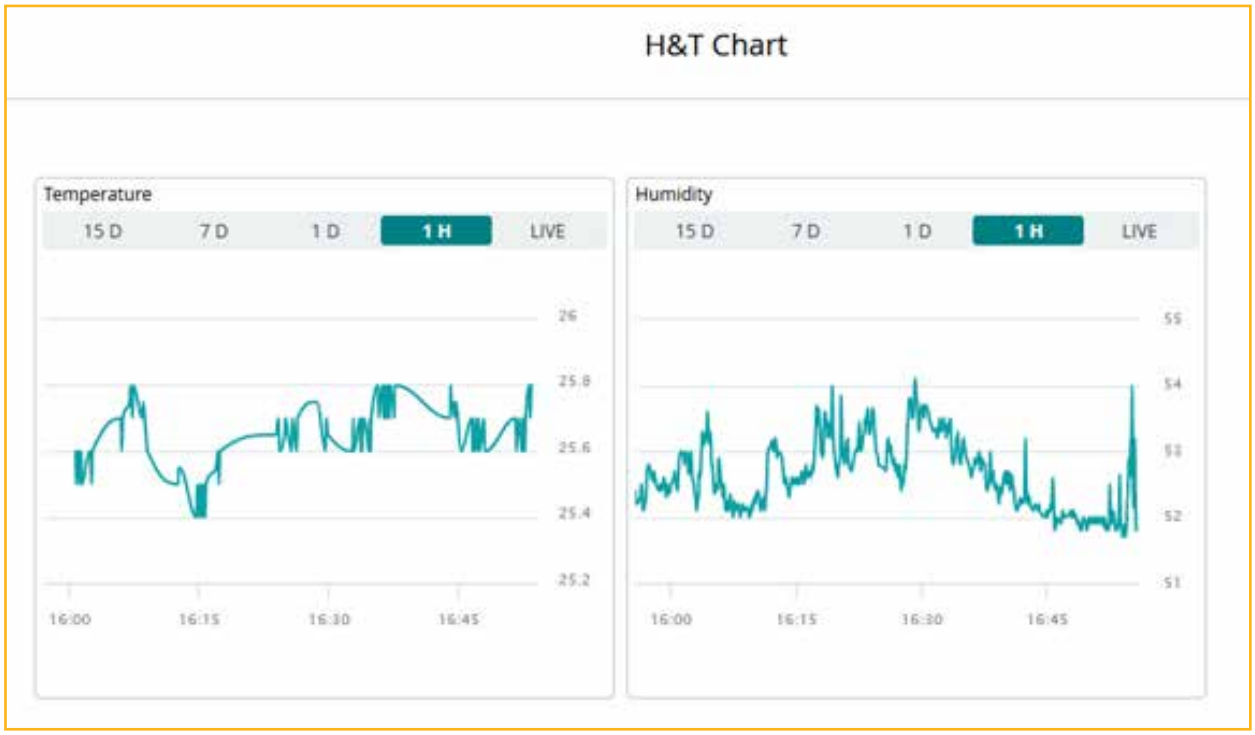


Do the same with a chart for humidity and look at the results.

Results

This diagram shows the course of the values over 1 hour. It is also possible to extend the observation period to a day or longer. You only have to switch to the daily display via the tab „1D".

However, a prerequisite for a correct display is that the data is available over this period, i.e. the device in question is already in operation over this period and is supplying data.



Next step – Control the soil moisture

In order to have a positive influence on plant growth and at the same time use the resource water as sparingly as possible, it is necessary to know the moisture content of the soil. Normally, it is the task of the gardener or the farmer to assess this and to ensure suitable conditions. However, with the spread of automation technology into many areas, it is possible to hand over tasks to technology in agriculture as well.

The basic aim in developing a soil moisture monitoring system is to ensure that the plants are watered at the right time and in the right amount. If the soil moisture is too low, there is a risk that the plants will be damaged, and if the soil moisture is too high, there is also a risk that the plants will be damaged.

Most plants grow well in soils with a moisture content between 20 and 60 %. A moisture content above 60 % can harm the plant. However, the 60 % threshold can vary.

Some experts believe that most flowers, trees and shrubs need a moisture content of 20 % to 40 %, while for vegetables you should maintain a soil moisture content of 40 % to 80 %.

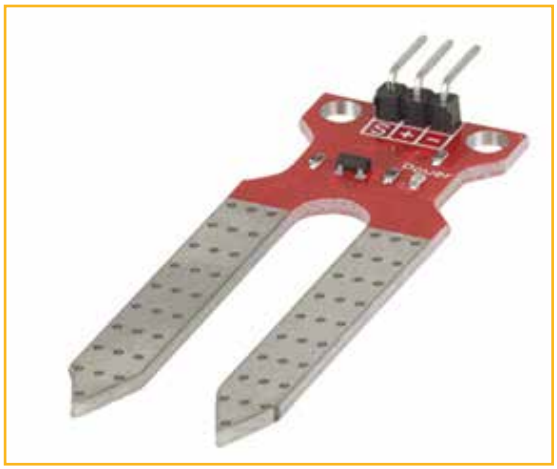
Simple measurement of soil moisture

To control and manipulate soil moisture, it is necessary to measure it.

There are several inexpensive sensors available for measuring moisture levels. Most Arduino kits include a simple moisture sensor with 2 tips.

Technical data

- Power supply: 3.3V / 5V DC
- Output signal: 0V to 1.7V / 0V to 3V
- Current consumption less than 20 mA



Connections

- Power supply (+)
- Ground (-)
- Analog sensor output (S)

This simple sensor is based on the principle of conductivity and is suitable for measuring both soil moisture and water level in a container. The two large exposed pads act as electrodes for the sensor and are used to measure resistance. The more water there is in the soil, the better the conductivity between the pads, resulting in lower resistance and higher voltage at the output.

How a resistive sensor works

A voltage is applied to the two contacts of the sensor. The higher the humidity between the two contacts, the better the current can flow from one contact to the other. This value is electronically processed in the sensor and transmitted to the microcontroller in the form of an analogue signal. In this way, numerical values from 0 to approx. 800 can be recorded and classified accordingly (800, as the full value - max. 1023 - cannot be reached due to the voltage divider via the supply voltage). The calibration depends essentially on the electrical conductivity of the liquid to be measured. This is known as a resistive sensor.

This sensor is very suitable for school projects and first experiments.

Capacitive sensor vs. resistive sensor

For the greenhouse project, a sensor will be used. This sensor uses a different measuring method. Moisture sensors can be either resistive or capacitive.

A capacitive sensor module uses capacitance instead of resistance to determine the water content of the soil. This type of sensor usually requires less power than a resistive sensor and is less susceptible to corrosion. A capacitive moisture sensor measures changes in capacitance caused by changes in the dielectric. Capacitive measurement essentially measures the dielectric that the soil forms, and water is the most important factor affecting the dielectric in this case.

The main disadvantage of the resistive forked sensor is that the probes inserted into the ground must be made of conductive bare metal and the electric current flowing between them will, over time, lead to corrosion of the probes by electrolysis.

The capacitive sensor improves this situation because the sensor metal embedded in the soil can be covered with solder resist to minimise corrosion. In addition, there is no electrical current flowing through the soil to cause electrolysis. The biggest weaknesses in the life of the sensor are the uncoated cut faces of the PCB, which can absorb moisture over time, and the exposed electronics on the top of the sensor when splashed with water.

To protect the electronics, it may be possible to encapsulate them with a suitable material. Clear nail polish or a similar material could be applied to the edges as an additional protective coating without seriously affecting the performance of the probe.

The analogue „Capacitive Soil Moisture Sensor V2.0“ is to be used for the greenhouse project.

Connections

- GND: Ground, must be common with the microcontroller
- VCC: 3.3V – 5.5VDC (May be powered from a digital output pin on a microcontroller)
- AOUT: Analog output usually connected to an analog input on a microcontroller

There is a 3-pin PH2.0 type JST connector on the sensor. One end of the supplied cable plugs into the connector, the other end is a standard Dupont 3-pin female connector. The cable is colour-coded with black for ground, red for VCC and yellow for AOUT.

To be noted during use

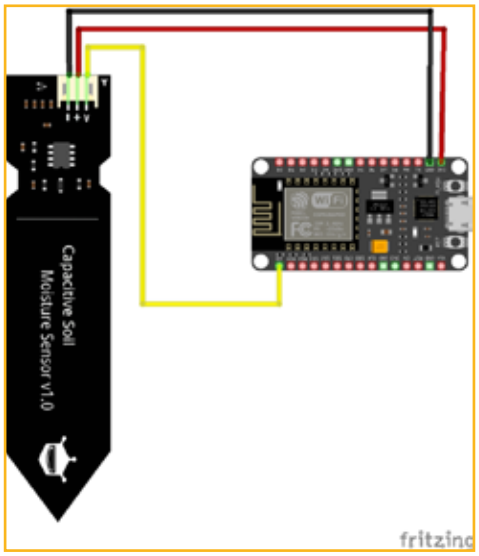
The probe is inserted into the soil to be monitored. The horizontal line indicates the maximum depth to which the probe should be inserted. The green area is the recommended depth to insert the probe. It is important that the electronics at the top of the probe do not come into contact with water or splashes to avoid damage.

Sensor calibration

The measurement range between wet and dry soil is determined experimentally by measuring the soil under different conditions and configuring the software to recognise when the soil is too dry, too wet or just right.

It is not possible to directly determine the actual moisture content of the soil from the measurements, but it is relatively easy to define basic ranges that are considered „too dry“, „too wet“ and „just right“.

First, a simple program can be used to monitor the output of the probe under various conditions and calibrate the settings. Simply connect the sensor output to A0 and apply power and ground. If necessary, you can connect the sensor connector to the microcontroller with male/male jumper wires. After downloading, open the Serial Monitor window to see the output of the sensor.



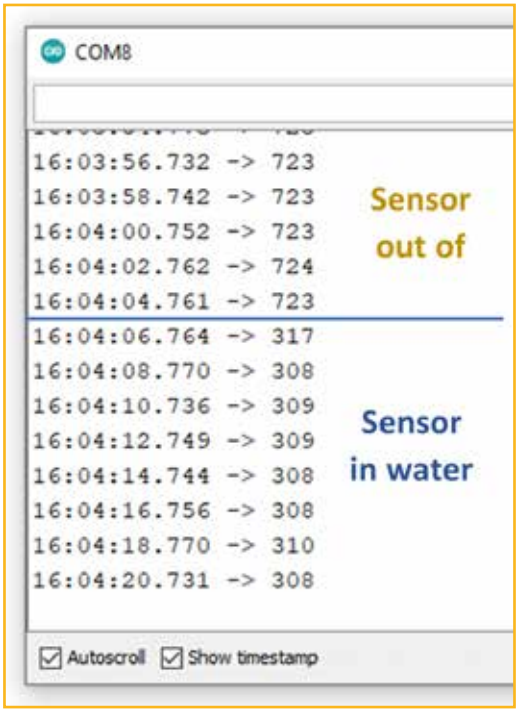
Simple programme for displaying the current sensor values

01_Moisture_Calibrate

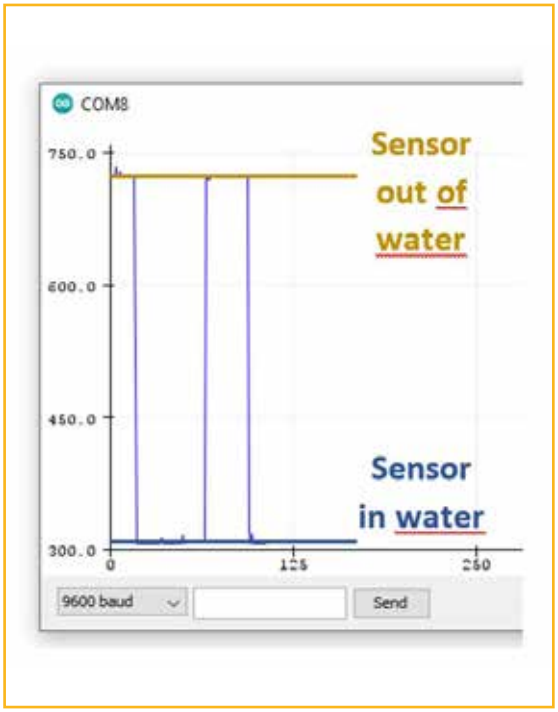
```
void setup() {
  Serial.begin(9600); // open serial port, set the baud rate as 9600 bps
}

void loop() {
  int val;
  val = analogRead(0); //connect sensor to Analog 0
  Serial.println(val); //print the value to serial port
  delay(2000);
}
```

01_Moisture_Calibrate.ino

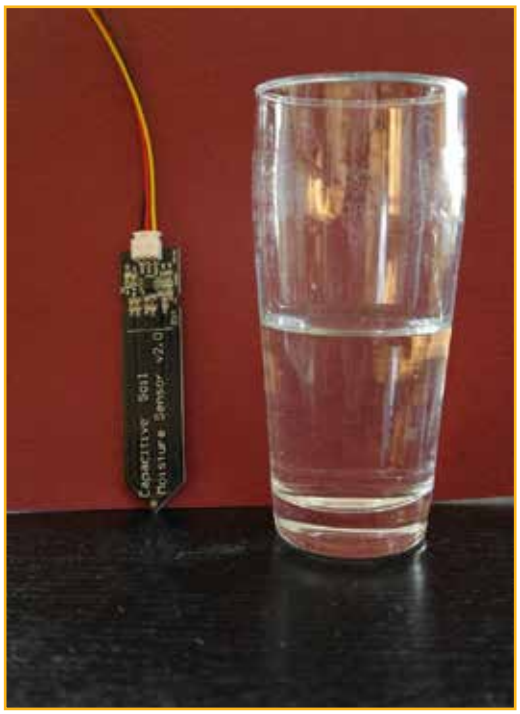
Start the Serial Monitor or Serial Plotter



Serial Monitor



Serial Monitor



Sensor out of water

Sensor in the water

Specifying the measurements

This procedure is used to determine the possible measurement limits. The measured values when the probe is dry and free in the air (driest value) and the other extreme is immersed in a glass of water up to the max line only represent the edge points.

By following this procedure, only the approximate maximum limits for the expected measurements can be determined. At the wet end of the scale, the water glass may not be quite as wet as very wet soil due to factors (e.g. texture of the water) other than the moisture present. Therefore, it is important to repeat the measurements at the field site for a functioning irrigation control.

Sensor under real conditions

Capacitive sensor vs. resistive sensor

For the greenhouse project, a sensor will be used. This sensor uses a different measuring method. Moisture sensors can be either resistive or capacitive.

A capacitive sensor module uses capacitance instead of resistance to determine the water content of the soil. This type of sensor usually requires less power than a resistive sensor and is less susceptible to corrosion. A capacitive moisture sensor measures changes in capacitance caused by changes in the dielectric. Capacitive measurement essentially measures the dielectric that the soil forms, and water is the most important factor affecting the dielectric in this case.

The main disadvantage of the resistive forked sensor is that the probes inserted into the ground must be made of conductive bare metal and the electric current flowing between them will, over time, lead to corrosion of the probes by electrolysis.

The capacitive sensor improves this situation because the sensor metal embedded in the soil can be covered with solder resist to minimise corrosion. In addition, there is no electrical current flowing through the soil to cause electrolysis. The biggest weaknesses in the life of the sensor are the uncoated cut faces of the PCB, which can absorb moisture over time, and the exposed electronics on the top of the sensor when splashed with water.

To protect the electronics, it may be possible to encapsulate them with a suitable material. Clear nail polish or a similar material could be applied to the edges as an additional protective coating without seriously affecting the performance of the probe.

The analogue „Capacitive Soil Moisture Sensor V2.0“ is to be used for the greenhouse project.

Connections

- GND: Ground, must be common with the microcontroller
- VCC: 3.3V – 5.5VDC (May be powered from a digital output pin on a microcontroller)
- AOUT: Analog output usually connected to an analog input on a microcontroller

Using a potted plant, measure the soil under 3 basic conditions:

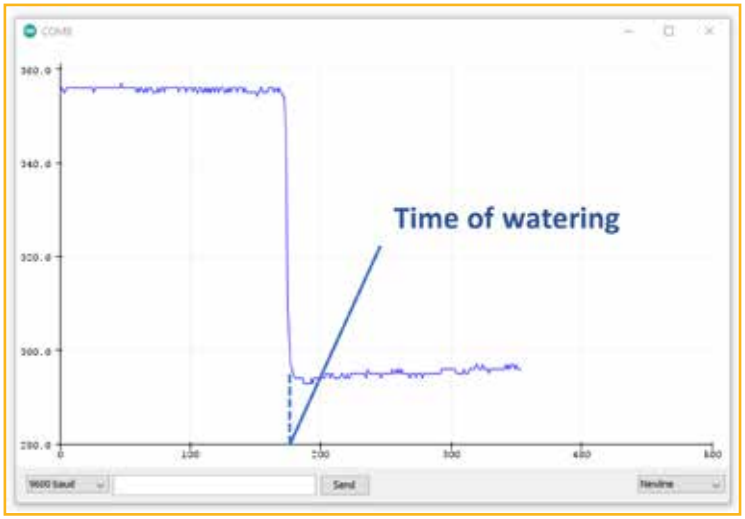
1. When dry enough so that the plant should be watered
2. When watered so it has the desired amount of moisture that would be considered ideal for the plant
3. When just watered heavily so the soil is on the too wet side and not ideal for the plant.

Using these 5 measurements (in the air, in the water glass and on the plant), the ranges for each of the 3 conditions can be determined initially and then refined once the plant is put into operation. It also depends a little on how dry or humid a particular plant species wants to be overall.

Examples for measured values

Dry in open air	710 - 730
Dry soil needing water	> 360
Ideal soil moisture	290 - 360
Just watered soil	290
In cup of water	270

Measurement in a flower pot



In a next step, the following areas can be defined in an irrigation control programme on the basis of this data:

1. < 280 is too wet

2. = 290 – 360 is the target range

3. > 360 is dry enough that we should water the poor plant

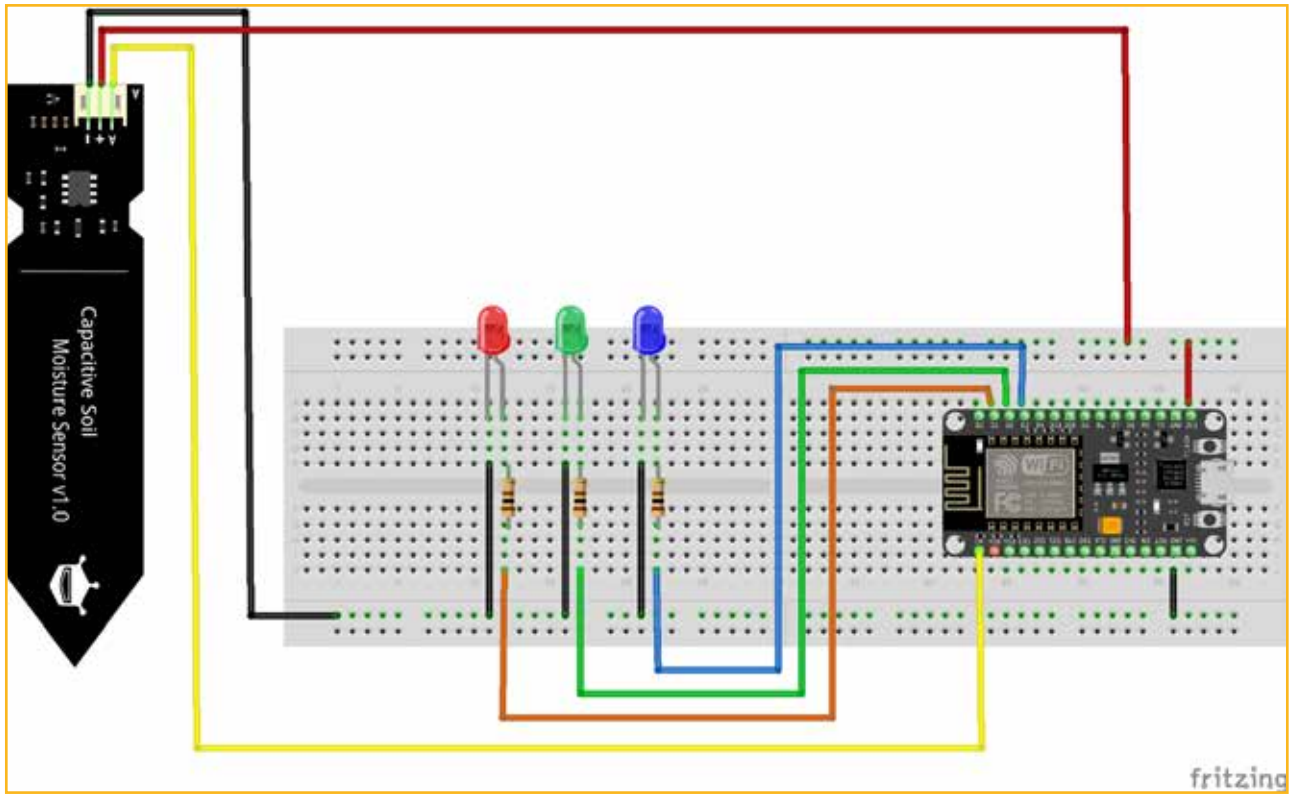
By simply adding an LED indicator with 3 LEDs to the existing programme:

- "Green" meant that everything was ok
- "Red" indicates that the plant is too dry
- "Blue" indicates that the soil is too wet

it is possible to control and visualise the moisture of the soil.

For this case, extend the small programme for reading out the values by this function.

Circuit



To realise this function, you can use either the “if ... else” - or the “switch ... case” - control structure. Try to find out which is better suited for the solution.

Further tasks

This simple programme reports the value or roughly shows the soil moisture via the 3 LEDs, but does nothing else.

1. The trigger point (lighting up of the red LED when the soil becomes too dry) could be used to switch on a small water pump to automatically water the plant.
2. In addition, the measured value could be shown via a suitable display. Consider how to convert the value evaluated by the analogue input to a percentage or another unit.

Tip: If you assume a linear behaviour of the input signal, the “map()” – function is useful to transfer the read-in values to practicable output values.

Hardware needed

- pump (solar or aquarium pump)
- relay
- external power supply
- hose
- water tank
- OLED - or LCD display

Notes on energy saving for battery-powered devices

Due to the very low power consumption of the Capacitive Soil Moisture Sensor V2.0 (approx. 5 mA), power could be supplied via a digital output pin. This makes it possible to switch the measurement on and off via the output pin, as the sensor may only need to be read every few hours or at fixed times of the day.

Note:

> When the power is switched on, the sensor takes a while to stabilise to a reading, so it is a good idea to wait at least 10 seconds after switching on with a delay () before taking a reading and making a decision.

You can find more information here:

https://arduino-tutorials.net/tutorial/capacitive-soil-moisture-sensor-arduino

The structures shown should serve as a basis and framework for your own developments. Try to create and implement a content-adapted and goal-oriented lesson plan according to your circumstances through extensions and additions. You will see what joy and interest the pupils will develop for the contents.

Have fun developing more IoT - Applications!

Chapter 3

Basic knowledge of sensor technology

3.1 INTRODUCTION TO SENSORS

This chapter gives you a brief introduction to the topic of sensors. This technology and an understanding of its basics are essential for all types of automation.

Without appropriate sensor technology, automation cannot and will not succeed!

3.1.1 MAN - MACHINE ARE THERE ANY RECOGNISABLE SIMILARITIES?

In order to be able to read this text, you depend on three organs of your body:

- the eye to recognise the characters
- the hand to hold the documents
- the brain to understand the content

All human activities are connected to the

- sensory organs for perceiving the outside world,
- muscles to perform actions and to the
- central nervous system for processing information and controlling actions.

In the course of technological development, humans have

- machines to facilitate physical work,
- computers to support mental activity, and
- sensors to record environmental situations.

The interaction of these three tools makes it possible to have a huge variety of routine human tasks in almost all fields carried out by technical devices and equipment.

3.1.2 PLACEMENT OF THE SENSOR TECHNOLOGY

Interaction of sensory organs and processing

And what about the technical realisation of sensory-physiological processes?

The sensory organs of living beings initially perceive nothing more than physical or chemical stimuli - light, sound, heat, moisture etc. Only after further processing of these influences in the central nervous system do the sensory impressions and, in interaction with the thought processes, the „mental effects“ and the reactions of the organism arise.

In simplified terms, this is called the IPO principle:

For this reason, it is not only sensible but even urgently necessary to also equip technical machines (robots) with the ability to perceive the senses, to equip them with „sense organs“, so-called sensors. The necessity of this development is thus rooted in the trend towards ever more comprehensive automation of industrial processes.

Importance of sensor technology for technical development

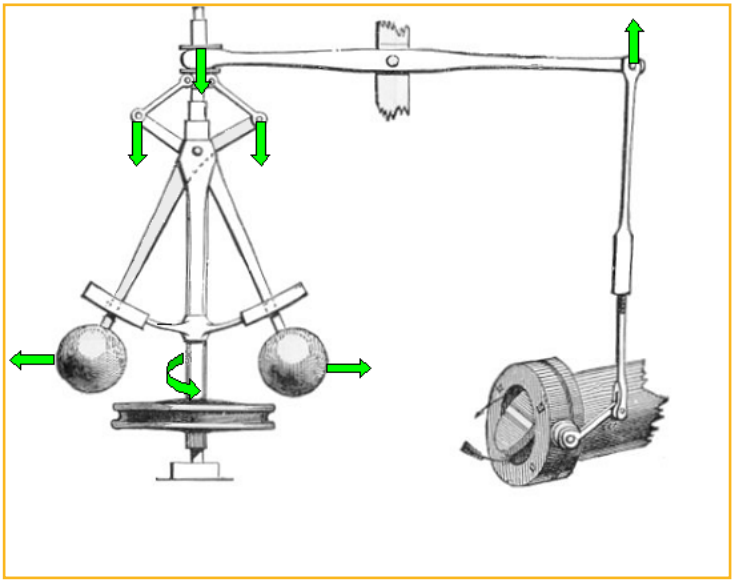
In the industrial revolution, it was important to build tools and machines that could relieve humans of heavy and monotonous work, or even make previously unfeasible work possible in the first place, and last but not least - do this work faster and better than humans could with their „bare hands“.

Sensors have become important components of automation technology in recent years. The manufacturing structures of industry have changed in such a way that the direct operation and monitoring of processes by humans is decreasing and being taken over by process controls.

The increasing automation and control of manufacturing structures, which are becoming more and more complex, requires components that make it possible to capture the data and information belonging to the manufacturing process. The human senses must be emulated in automation by sensors that are capable of sensing the environment.

Origin of sensory elements

When the early phase of industrialisation began, engineers could not limit their designs to productivity alone; they also had to consider its reliability, interference resistance and the protection of the operator. Many useful devices were invented to prevent damage to the equipment and to the operators, e.g. centrifugal governors on steam engines, coolant control on machine tools, two-hand operation on presses or punches.



From Wikimedia Commons, the free media archive

Thus, the first elements were created that allowed a simple „reaction“ of the machine to unforeseen or undesired influences as well as changes in the functional sequence. With the further progress of technology, especially with the development of more complex machine tools and robots, these conventional aids were no longer sufficient.

If an industrial robot is to pick out only certain parts from different individual parts or guide a welding torch along a joint that does not exactly match the pre-calculated course, it needs sensory recognition systems.

Without sensors no robots

Basically, sensors correspond to the sensory organs of humans, even if they look quite different and are partly based on other functional principles. Without such devices, robots are of limited use; automation in industry would very soon reach its limits.

Equipping industrial robots and automated processes with sensors as „technical sense organs“ is thus an unavoidable condition for their use in those areas where humans cannot yet be replaced by automation today, because the proper course of this work is decisively controlled by the human sense organs and flexible reaction.

In the meantime, there are also sensors for environmental effects and technical variables for which humans have no sensory organs.

These include values of

- field strengths (electrical and magnetic),
- electrical voltage and current,
- ionising radiation (e.g. radioactivity)
- and much more.

Term „sensor”

The term „sensor“ is a linguistic creation of our time. It only found its way into reference works in the seventies.

Whereas in the past a wide variety of terms were used, such as sensor, initiator, proximity switch, transmitter, probe, switch and transducer, today the term „sensor“ is increasingly gaining acceptance in international terminology. This refers to all components that serve to record measured values in the broadest sense.

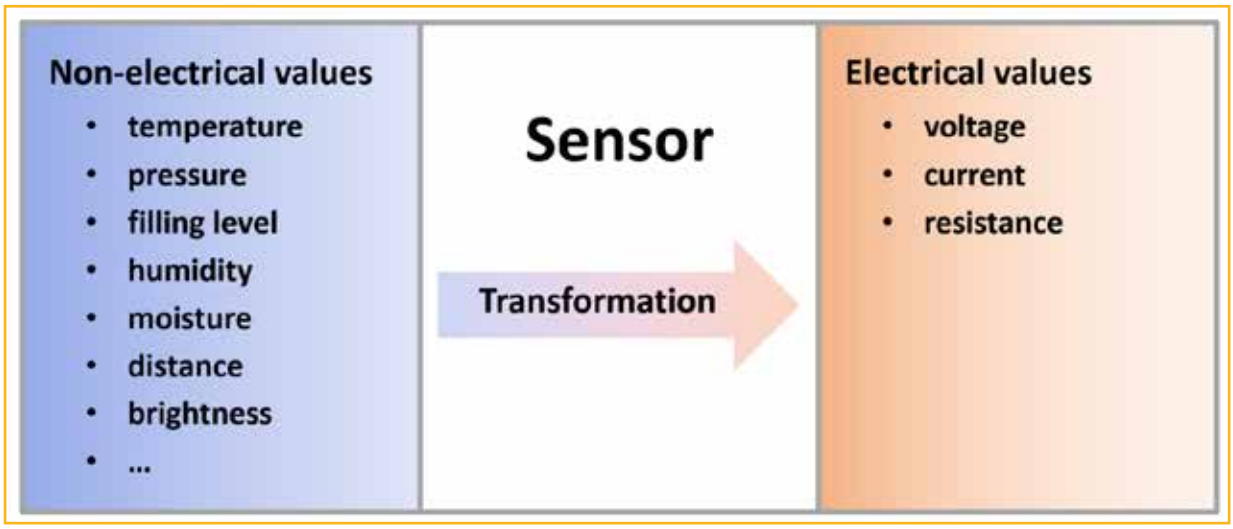
The linguistic origin lies in Latin (sensus: feeling, sensation). The terms sensitive, sensible, sensitisation or sensitometry, which are also derived from this, have been in use for a long time and describe - especially in medicine and psychology - characteristics, states or processes that have something to do with feeling, sensation and sensitivity.

„Sensor“ is - literally translated - a „feeler“, a „sensing device“, and this sense is increasingly understood in technical terms today.

3.1.3 FUNCTION AND CLASSIFICATION OF SENSORS

Sensors transform „environmental parameters“ into a processable electrical signal by means of physical, chemical or biological effects. Only electrical signals/parameters can be used by electronic controllers.

Sensors transform a non-electrical value into an electrical value.



Sensors can be divided into passive and active sensors.

Passive sensors

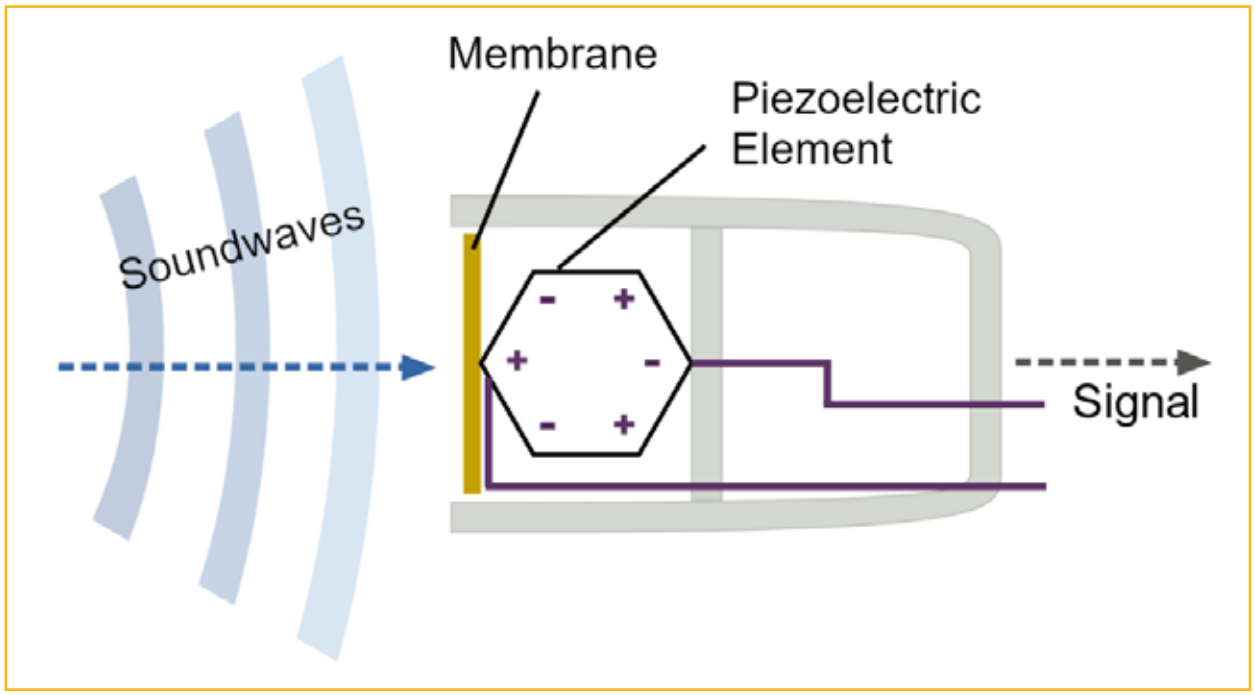
Passive sensors contain passive components (e.g. resistors, coils and capacitors) whose electrical properties are changed by the measured variable. They change their electrical property, e.g. resistance or inductance, under the influence of the non-electrical or physical variable, e.g. pressure, temperature or the magnetic field. In order for the electrical variable to be influenced by the components, an external auxiliary energy in the form of a current or voltage source is required.

Active sensors

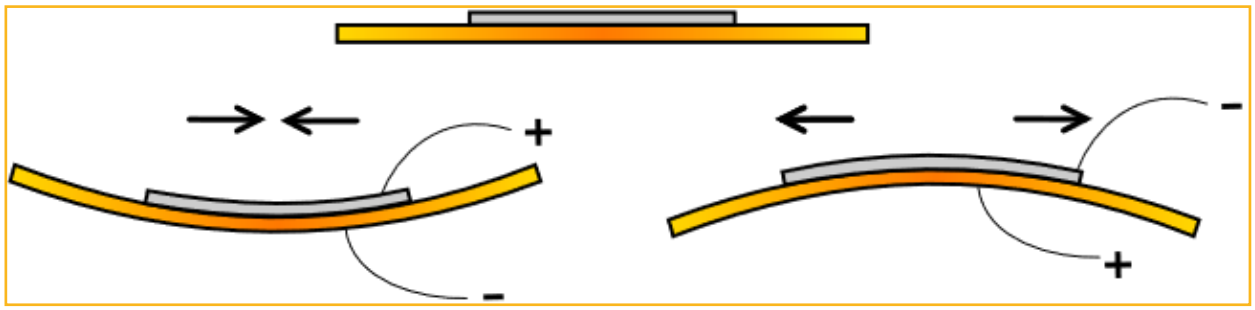
Active sensors generate an electrical signal due to the measuring principle, e.g. electrodynamic or piezoelectric. These sensors are therefore voltage generators themselves and do not require any electrical assistance energy. They convert mechanical, thermal, light or chemical energy directly into electrical energy.

Active sensors are usually voltage generators such as thermocouples, photoelements or piezoelectric elements. If necessary, the output signal must be amplified, linearised or calibrated by a transmission element or a measuring amplifier.

Active sensors can also be used as actuators by reversing the physical measurement principle, e.g. a dynamic microphone can also be used as a loudspeaker.



Galak76, CC BY-SA 3.0, via Wikimedia Commons



From:
PiezoBendingPrinciple.gif: Sonitron Supportderivative work: Gregors (talk) 14:24, 17 February 2011 (UTC) - PiezoBendingPrinciple.gif, CC BY-SA 3.0

Classification by the principle of action

Sensors can be classified according to the operating principle on which the sensor is based. There are a variety of sensory approaches for each active principle.

Examples

Principles of action	Example
mechanical	pressure manometer, expansion lever, spring balance, lever balance, thermometer, ...
thermoelectrical	thermocouple (temperature measurement)
resistive	strain gauges (DMS), semiconductor DMS, Pt100, ...
piezoelectrical	acceleration sensor, pressure sensor, force sensor, ...
capacitive	pressure sensor, rain sensor, humidity sensor, ...
inductive	inclinometer, force sensor, position sensor, ...
optical	CCD sensor, photocell, light barriers, proximity switch, ...
acoustical	level sensor, ultrasonic flow meter, ...
magnetic	hall sensors, reed contact, ...
...	...

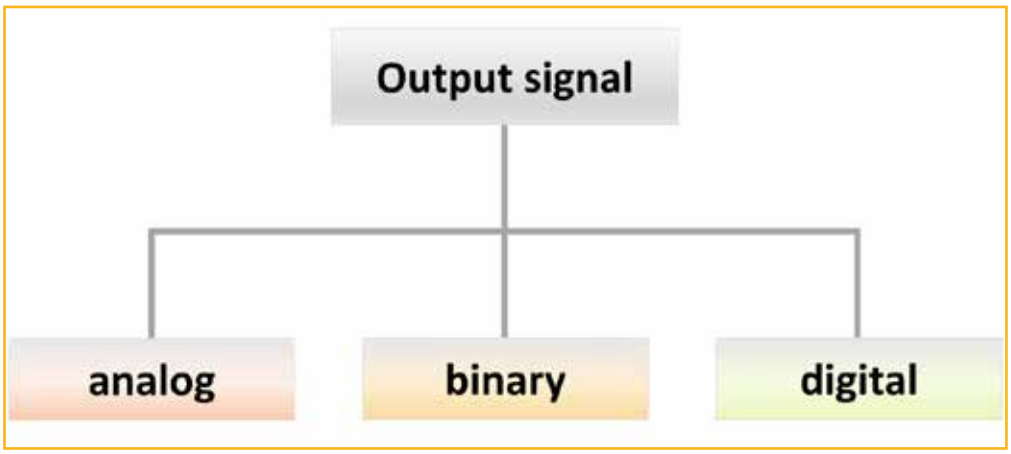
Source: https://de.wikipedia.org/wiki/Sensor

This list is not complete.

Classification by signal output

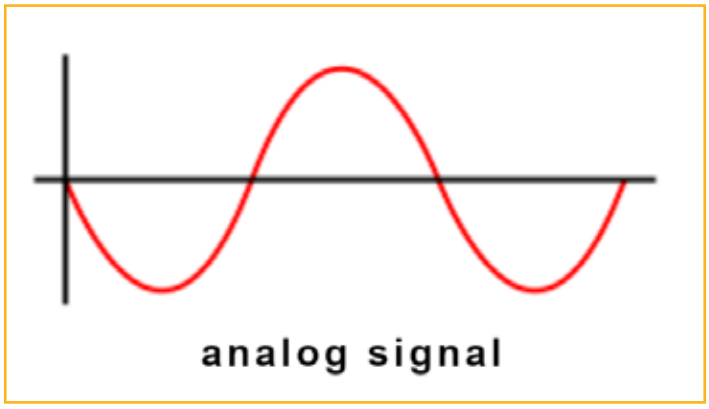
Sensors can also be subdivided according to the type of output signal. This is especially important for the type of further processing.

A distinction is made:



Analog sensor

Analog sensors generate a continuous analog output signal. This output signal is proportional to the measured variable. With suitable output devices, it can be displayed as a function of time, with the signal constantly assuming one of the many possible values.



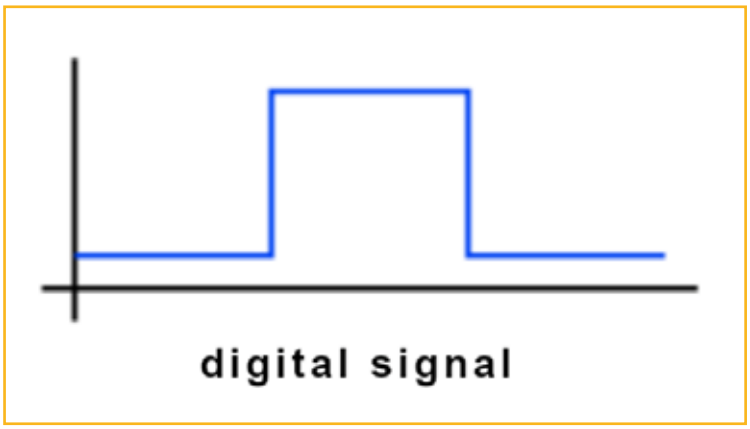
Analog sensors are often used in applications that require higher accuracy and/or lower noise interference than digital sensors. For example, temperature measurements for precision instruments may require an analogue sensor because they can detect small changes in the measured parameter. Analog sensors are also very suitable for tracking constantly changing physical quantities.

Another advantage of analog sensors is their ability to measure physical phenomena over a wide range of values. For example, thermometers can measure temperatures that vary from very low to very high depending on the type used.

Digital sensor

Digital sensors are sensors that measure a physical quantity and convert it into an immediately usable digital signal. The conversion of the analogue measured value is done by means of an integrated analogue-to-digital converter (ADC).

The output signal is given as a sequence of binary values (high signal level = 1 or a low signal level = 0). If this sequence of numbers is coded and transferred to a bus system, the digital values can be sent to the processing unit and processed directly. Various buses such as CAN bus, Profibus, Industrial Ethernet, I2C or USB can be used to transmit the measured value to a digital interface.



Binary sensor

Basically, a binary sensor is nothing more than a switch. It can only accept the values 0 and 1 at the output. This type of output is very often used for proximity switches, light barriers, presence detection, etc.

3.2 APPROACH IN CLASS

The topic of „sensor technology“ is a perfect choice for technical lessons, because sensors have become an indispensable part of many everyday objects. In the practical handling of sensors, the students are given a basic understanding of how technology works and how it is used, and are thus taught everyday skills in the use and evaluation of technology.

3.2.1 APPROACH AND GENERAL CONDITIONS

If everyday competence is the aim, the central content of the lessons should be, in addition to the (concrete) functioning of selected sensor elements, the conversion of a physical measured variable into an electrical variable. This includes above all the calibration of the sensor element, i.e. the determination of the functional relationship between physical input and electrical output variable, in order to be able to use the sensor element sensibly.

In addition to the students‘ own experimental measurements and approaches, it is useful for the teacher to set tasks that build on each other. The tasks should lead the students‘ knowledge in the right direction.

When developing a sequence of lessons or small projects, the following considerations should be taken into account:

- To activate students, they should experiment on their own and independently.
- In order for the students to become aware of the function and importance of the sensor elements, they should explicitly explore them themselves.
- In order to address all students equally, in our experience it has proven successful to have sensor elements explored right at the start of the topic.
- In a second step, the sensor elements studied are used to build a device themselves; here, an everyday reference can easily be made.
- A project should include as many different types of sensors as possible. (e.g. construction of a weather station, control of a greenhouse, sorting device for different objects, ...).

Particularly useful for getting started are

- switching (binary) sensors

(push-buttons, reed contacts, light barriers, motion detectors, inductive and capacitive proximity sensors, incremental encoders, ...),

- resistive sensors

(temperature dependent resistors, light dependent resistors - LDR, pressure / force dependent resistors - DMS, ...) and

- analog sensors

(ultrasonic sensors, active temperature sensors, humidity sensors, ...).

3.2.2 HOW SHOULD STUDENT PROJECTS ON THE TOPIC OF SENSORS BE DESIGNED?

Understanding how sensors work is particularly important when using them.

Approach to the introduction:

1. Awaken interest, get to know different sensor elements:

   What is the senor used for? Where is it used?

2. Theoretical basics, how the sensor elements work:

   How does the sensor work? What physical principles are behind the principle?

3. Technical basics of the sensors used:

   Types, data sheets, connections, ...

4. Experimental learning about the properties of the sensor:

   First experiments through basic set-ups, analysing first output values of the sensor.

5. Set tasks that build on each other in a practical way:

   Use of the sensors in practical applications (everyday devices) with increasing difficulty and complexity.

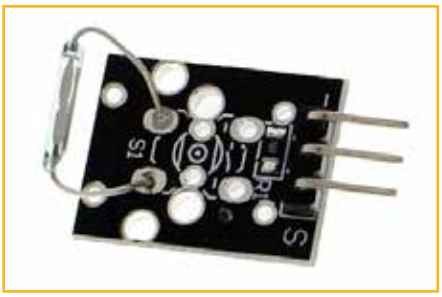
6. Documentation and presentation of the results and learnings to an appropriate audience.

3.2.3 SELECTION OF SUITABLE SENSORS

It is unnecessary and perhaps even confusing to try to give an overview of all the different types of sensor elements. The selection criteria should be availability, price and technical/physical complexity.

Small selection of sensors (especially suitable for Arduino projects)

Magnetic reed module (Reed switch)



Infrared Sensor
(Proximity Switch)

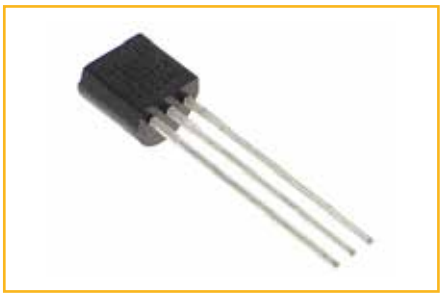
Distance sensor, obstacle detection sensor - Infrared (IR)

Distance sensor
(Inductive proximity switch)

Laser light barrier
(M12JG-30N1 – NPN)

Ultrasonic sensor
(Distance sensor - HC-SR04)

Temperature sensor TMP36



DHT22
Temperature and humidity sensor

Capacitive moisture sensor

Photo resistor LDR 10K

You can find more information here:

https://arduinogetstarted.com/arduino-tutorials

https://lastminuteengineers.com/tmp36-temperature-sensor-arduino-tutorial/

3.2.4 TEACHING UNIT 3: USING A TEMPERATURE SENSOR

After the theoretical introduction to the topic of sensors, an example will be used here to show how the students can learn about the function and application of sensors. A simple and cheap active sensor is used for temperature measurement. The TMP36 sensor outputs a voltage signal proportional to the temperature at one socket.

The worksheet is also available as a Word document called „03_Worksheet.docx“ and can be adapted to suit your needs.

Please note!

The following worksheets (written in blue colour) are available for download here:

Task: Measuring a temperature with an active sensor

Learning Objectives:

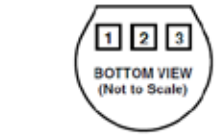
- how to use a temperature sensor (TMP36)
- using the “map” - command
- output of the values (serial monitor, serial plotter)
- averaging of values
- application of “if – else” commands (interlaced and not interlaced)

Required equipment:

Arduino / breadboard / cables / temperature sensor TMP36 / external power supply, resistors, LEDs (green, red, blue), piezo speaker, OLED display

Description of the sensor:

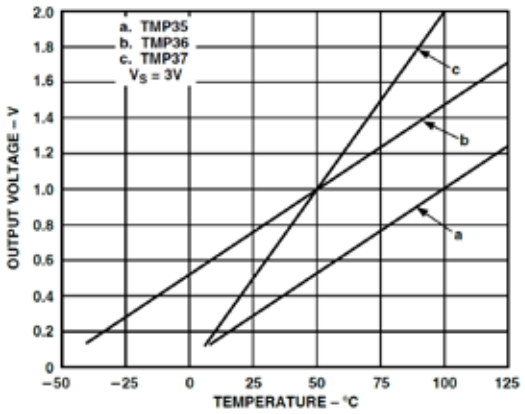
The sensor has three terminals: 5V, GND, and the pin for the temperature signal. On this pin, the sensor puts out a voltage between 0 and 2.0 volts.



PIN 1, $+V_S$; PIN 2, V_{OUT}; PIN 3, GND

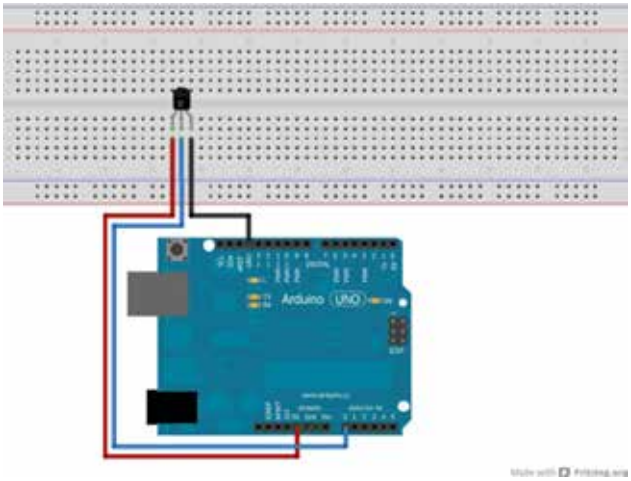
The sensor is supposed to be quite precise (±2°C) between -40°C and 150°C, according to the manufacturing. If an external power supply (if possible 9V battery or 9V power supply) is used, the sensor accuracy is higher.

The voltage on this pin 2 must be read out by the microcontroller board and then has to be converted into a temperature value.



TPC 1. Output Voltage vs. Temperature

Connection diagram:



CAUTION:

If the sensor is connected incorrectly, it will be destroyed.

Explanations:

The sensor is connected to the analog Pin A0.

Example for a simple sketch:

01_Temperature_measurement

```
int TMP36 = A0;
int temperature = 0;
int temp[5];
int wait = 500;


void setup()
{
Serial.begin(9600);
}

void loop()
{
temp[0] = map(analogRead(TMP36), 0, 410, -50, 150);
delay(wait);
temp[1] = map(analogRead(TMP36), 0, 410, -50, 150);
delay(wait);
temp[2] = map(analogRead(TMP36), 0, 410, -50, 150);
delay(wait);
temp[3] = map(analogRead(TMP36), 0, 410, -50, 150);
delay(wait);
temp[4] = map(analogRead(TMP36), 0, 410, -50, 150);

temperature=(temp[0]+temp[1]+temp[2]+temp[3]+temp[4])/5;

Serial.print(temperature);
Serial.println(" degree");
}
```

Under the variable **“temperature”** the temperature value will be saved.

To get good values we have to read out some values to get the mean value out of them. The square brackets **“[5]”** create five different variables at one time: **„temp[0]“**, **„temp[2]“**, ...till... **„temp[4]“**. So with this spelling [5] we will save some memory space. The value after “wait” sets the distance between each measurement.

In the setup we are going to start the serial communication, to see the temperature values on the serial monitor. The microcontroller will send values to the computer.

Serial.begin(9600);

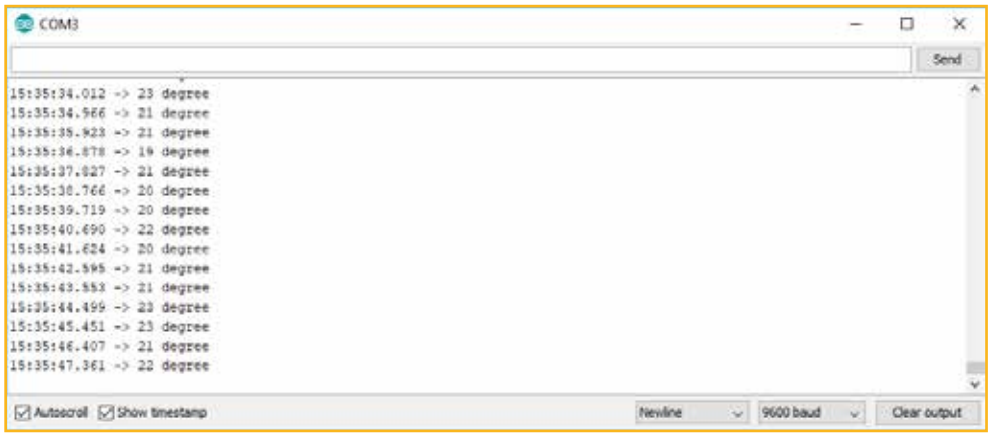
Using the Serial Monitor and the Serial Plotter

Command to send the temperature values to the computer and show them on the serial monitor or the serial plotter:

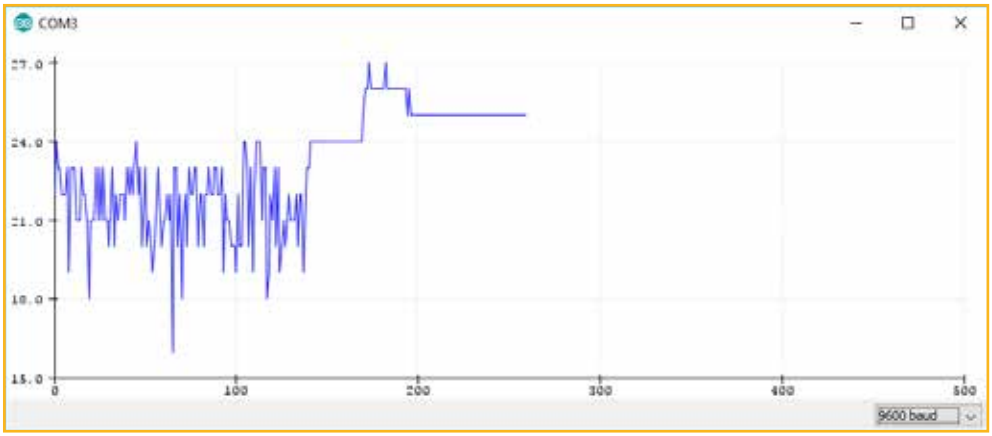
Serial.print(temperature);

Serial.println(„ degree“);

The serial monitor and the serial plotter can be started at “Tools” in the Arduino software.



Serial Monitor



Serial Plotter

It is better when the temperature is read out serval times. To get good values we have to read out some values to get the mean value out of them. In between the measures is always a little break. The square brackets "[5]" create five different variables at one time: „temp[0]", „temp[2]", ... till ... „temp[4]".So with this spelling [5] we will save space in the memory.

```
int temp[5];  //save space for five integer variables with the name "temp"

temp[0] = map(analogRead(TMP36), 0, 410, -50, 150);
...
temp[4] = map(analogRead(TMP36), 0, 410, -50, 150);
```

Explanations:

The sensor is connected to the analog Pin A0.

Example for a simple sketch:

Additional information

Arrays

Basically, arrays are not a separate type of variable, they are a grouping of several variables of one type.
int value[5] = {10,12,32,46,50};
In the example, an array of type „int“ is created first. The 5 in square brackets after the variable name determines the number of memory locations that the array provides. The number of memory locations is also called the length of the array.
In a program one can access the memory locations of the array by indexed queries.
The first digit in the array is the digit 0:
value[0] contains the value 10
In use: analogWrite(ledPin, value[0]);

We will take a closer look on the used command: “map(a,b,c,d,e)”

This is a “Map command”. With this command it is possible to change a read out value (a) from one region between (b) and (c) into a region between (d) and (e).

In our case this means:

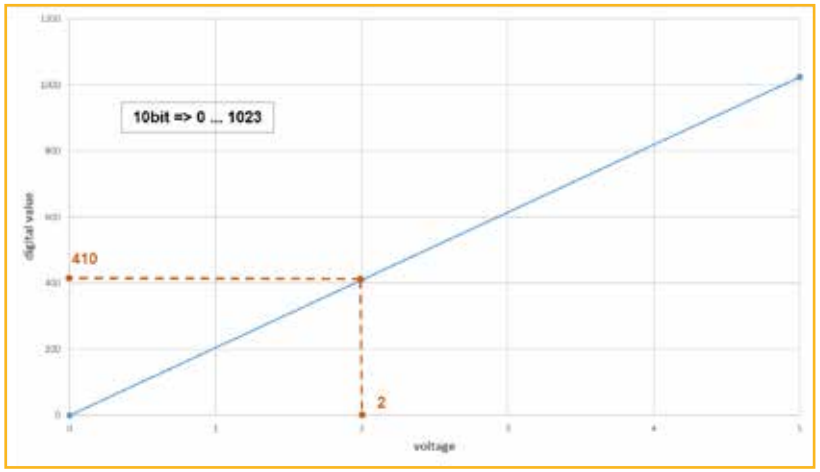
a = value to be converted

b = minimum measuring range

c = maximum measuring range

d = minimum output value

e = maximum output value



The sensor value is read out right in the map command “analogRead(TMP36)”. The value should be between 0 and 410 (= values between 0V and 2V at the analog port). The sensor outputs this voltage values if it measures temperatures between -50°C and 150°C. With the “map command” these values get converted into degree values between -50°C and 150°C.

```
temperature=(temp[0]+temp[1]+temp[2]+temp[3]+temp[4])/5;
```

In this line, all five values are summarized and divided by five. This average value is saved under “temperature”.

Tasks

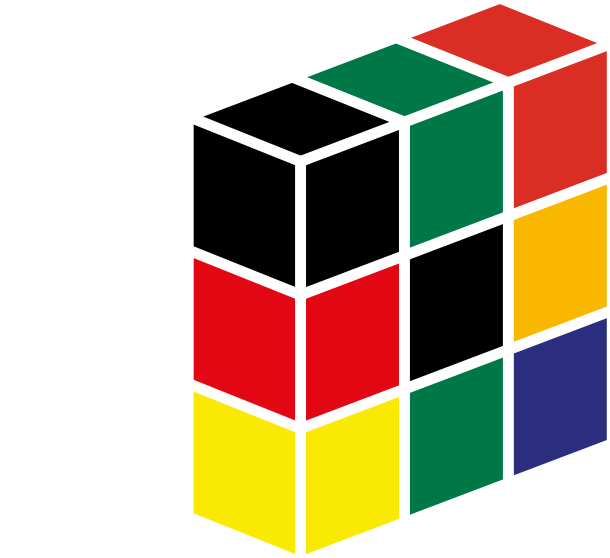
Task 1: Complete the incomplete program „01x_Temperature_measurement“ so that it uses the temperature sensor to measure the current temperature 5 times, form an average of these values and output them via the serial monitor or the serial plotter on the screen.

Task 2: Use the “if-else” function to switch a green or a red LED on or off if the temperature is correct (green LED) or too high (red LED).

Task 3: Continue to use the “if-else” function (interlaced) to control a third LED (blue). The blue LED should light up when the temperature falls below a certain value, the red LED should be activated if the temperature exceeds a higher value. The intermediate temperature range should be indicated by the green LED lighting up.

Task 4: A too low (blue LED) and a too high temperature (red LED) should be indicated by a short beep (0.5 s) via a loudspeaker.

Task 5: Display of the measured value on an OLED display.



TRAINME

This textbook has been developed within the framework of the TRAINME & TRAINME 2 programme (Modular Training of South African TVET Lecturers in Mechanical and Electrical Engineering). The TRAINME programmes have been designed by the Inter-Company Training Centre in Eastern Bavaria (ÜBZO) and the University of Stuttgart (Dept. BPT) on behalf of the German Federal Ministry of Education and Research (BMBF) and the DLR Project Management Agency (DLR-PT) in cooperation with the South African Department of Higher Education (DHET).

Powered by:

ÜBZO
Überbetriebliches Bildungszentrum in Ostbayern gemeinnützige GmbH

University of Stuttgart
Germany

Supported by:

higher education & training
Department: Higher Education and Training
REPUBLIC OF SOUTH AFRICA

SPONSORED BY THE
Federal Ministry of Education and Research