ALL-IN-ONE MICROCONTROLLER EDUCATION KIT

ARD-SET02

TABLE OF CONTENTS

TABLE OF CONTENTS

1. GENERAL INFORMATION

Dear Customer,

Thank you for choosing our product. Below, we will show you what you need to bear in mind when commissioning and using.

Should you encounter any unexpected problems during use, please do not hesitate to contact us.

These instructions were written using Arduino IDE 2.3.5.

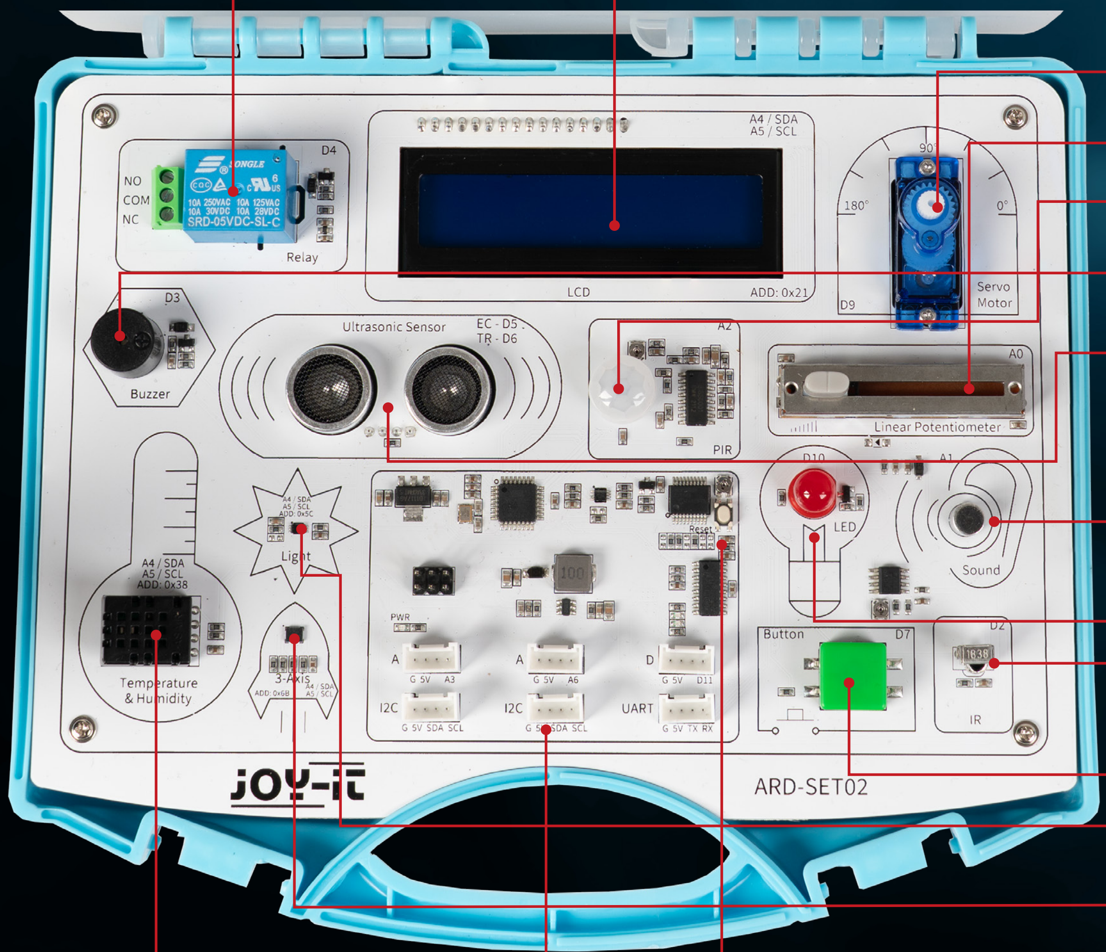
This manual has been automatically translated.

2. DEVICE OVERVIEW & PIN ASSIGNMENT

The All-in-One Microcontroller Education Kit is a comprehensive and beginner-friendly learning platform designed to introduce users to the world of electronics, programming and embedded systems. The compact kit is housed in a robust plastic case and features a fully integrated main board with an ATmega328P-compatible microcontroller. Thanks to its enclosed and pre-wired design, no additional hardware assembly is required, allowing users to start learning and experimenting right away. With the most important components already integrated, you save time and effort on wiring. The All-in-One Microcontroller Education Kit features a wide range of sensors and actuators, allowing you to start your projects right away with a variety of modules. With the integrated breadboard, you can quickly build and realise your own projects.

A wide range of sensors and actuators are already integrated into the system, allowing important microcontroller concepts such as digital and analogue inputs, signal processing and peripheral control to be explored in practice. Typical applications include measuring environmental data, detecting motion and distance, controlling outputs such as LEDs, buzzers, relays and servo motors, and displaying information on an integrated LCD display. This diversity makes the kit suitable for both structured teaching and creative, project-based learning.

Programming and power supply are provided via an easily accessible USB-C port on the housing. This modern interface ensures a reliable connection and simplifies daily use in classrooms, laboratories or at home.



Relay Pin D4

16x2 LCD screen
SDA Pin A4 & SCL Pin A5
I^2C address: 0x21

Servomotor Pin D9

Linear potentiometer Pin A0

PIR motion sensor Pin A2

Buzzer Pin D3

Ultrasonic sensor
Echo pin D5 & trigger pin D6

Sound sensor Pin A1

LED Pin D10

Infrared sensor Pin D2

Button Pin D7

Light sensor
SDA pin A4 & SCL pin A5
I^2C address: 0x5C

Acceleration sensor
SDA pin A4 & SCL pin A5
I^2C address: 0x6B

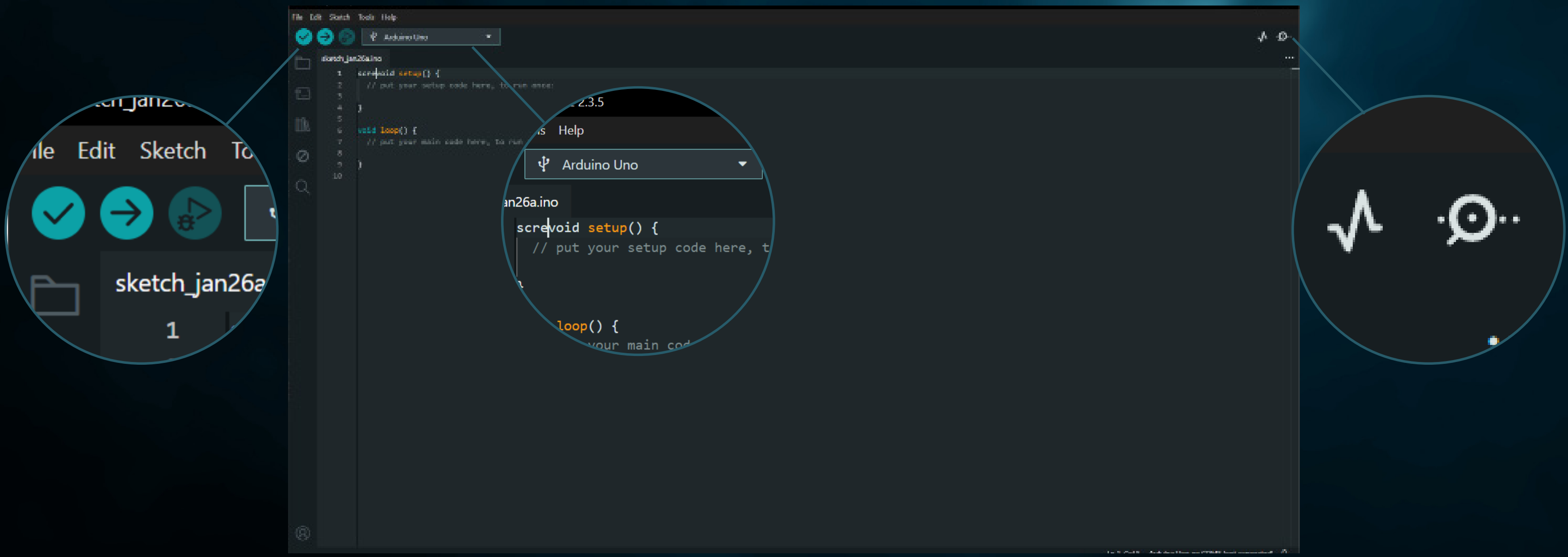
DHT20 Temperature & Humidity Sensor
SDA Pin A4 & SCL Pin A5
I^2C address: 0x38

Free interface connections

Reset button

3. ARDUINO IDE

Thanks to its ATmega328P-compatible architecture, the kit can be programmed using familiar development environments such as the Arduino IDE. You can download this from here.



To programme the device, connect the supplied USB-C cable to your computer.

ATTENTION! In the Arduino IDE, you must select the correct port and board (as shown in the illustration). For the All-in-One Microcontroller Education Kit, select Arduino Uno as the board.

You can transfer the code to the education kit using the **upload button**.
The **serial monitor** is used to exchange data between the learning kit and the computer. In the examples, the serial interface is started with `Serial.begin(9600);` in **setup()**, where **9600** baud specifies the transmission speed. The serial monitor can be opened in the Arduino IDE via the **magnifying glass icon** in the top right-hand corner. To ensure that the output is displayed correctly, the baud rate must also be set to **9600** in the serial monitor.

From basic configuration to project implementation, this guide will walk you through the entire process. Our guide includes easy-to-understand explanations and useful tips to help you quickly and effectively develop your microcontroller skills.

4. MODULES IN DETAIL

All modules available on the All-in-One Microcontroller Education Kit are explained individually below with sample codes.

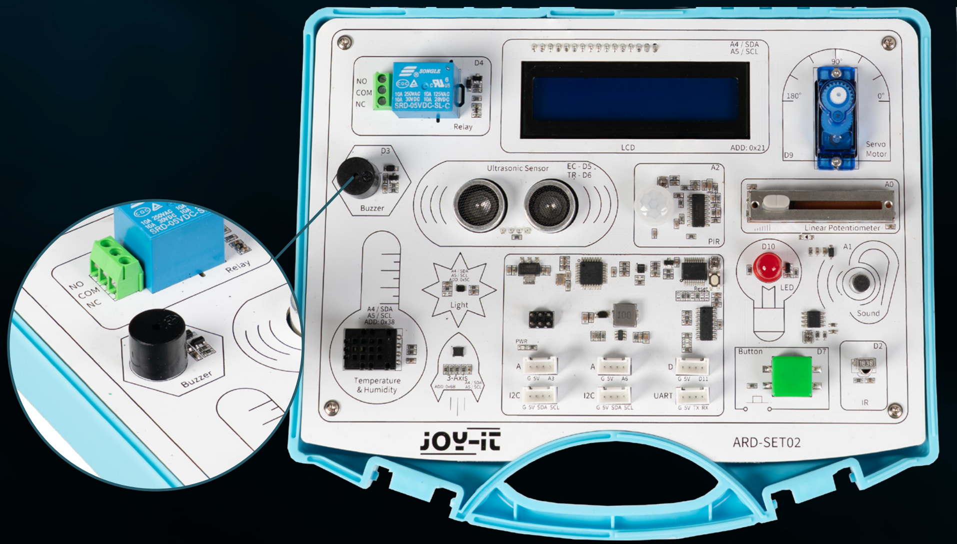
HERE you can download all sample codes for using individual modules.

External libraries are required to use some modules. Download the libraries via the IDE using the library manager. There you can install the specified libraries using the search function.

4.01. BUZZER

A buzzer or buzzer generates an audible signal, similar to a loudspeaker. Unlike a loudspeaker, however, it is only suitable for a limited frequency range, meaning that it does not produce a good sound for playing music or speech. It is ideal for generating loud warning tones in the form of beeps. Whenever an electrical device emits a warning tone, it is almost always a buzzer. Examples include alarm clocks, smoke detectors and seat belt reminders in cars.

The buzzer is connected to pin D3.

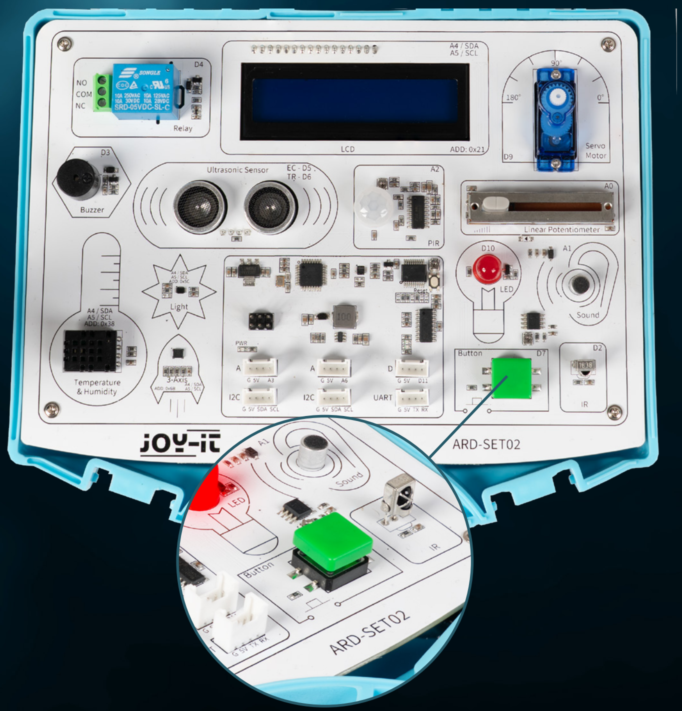


In the code example, the songs Für Elise by Ludwig van Beethoven and Frère Jacques by Jean-Philippe Rameau are played on the buzzer. To do this, the different frequencies for each playable note were defined, as well as the temporal sequence of the notes of the respective song.

4.02. BUTTON

Buttons are interactive elements in user interfaces that perform a simple but essential function: user input. They are used to initiate a wide range of commands and actions in digital environments.

The button is connected to pin D7.



In the code example, the button is configured as an input. The status of this input can then be queried in the programme. If the eductaion kit detects a low signal at the corresponding pin, the code knows that the button has been pressed. If, on the other hand, there is a high signal, the button is considered not to have been pressed.

4.03. RELAY

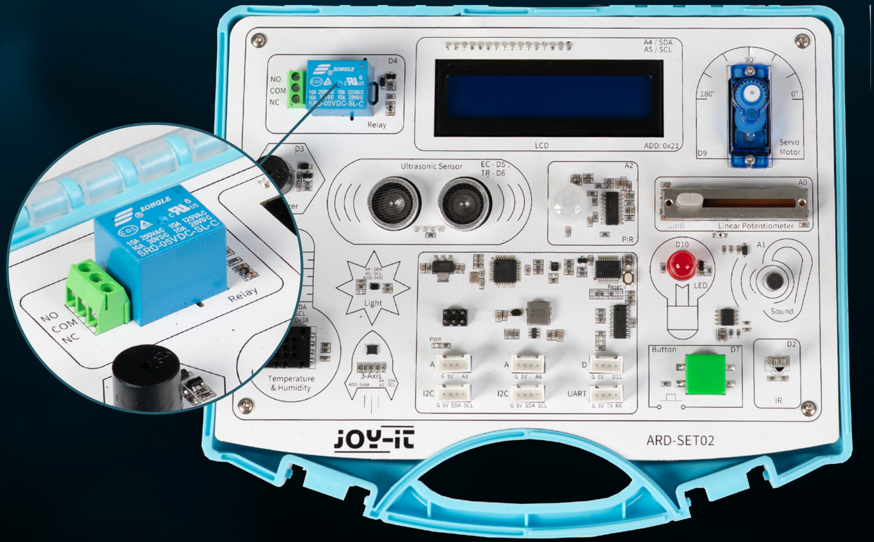
Relays are among the oldest electromechanical components and function like electrically controlled switches. A small control signal at the input can be used to switch a significantly larger load at the output on or off. This allows lamps, motors or other devices with higher power requirements to be safely controlled via a microcontroller or a education kit, for example.

Inside a relay is a coil. When activated, it generates a magnetic field that moves a mechanical switch contact. This contact then connects or disconnects the output terminals. As soon as the control signal is switched off again, the contact springs back to its original position.

A relay usually has three connection terminals: ***COM*** *(****Com****mon),* ***NO*** *(****N****ormally* ***O****pen) and* ***NC*** *(****N****ormally* ***C****losed). COM is the common connection. NO is not connected to COM in the idle state and is only closed when the relay is activated. NC is connected to COM in the idle state and is disconnected when activated.*

Stripped cable ends can be inserted into the terminal socket and securely fastened by tightening the screws. This allows the three connections to be used easily and reliably.

The relay is connected to pin D4.



Attention! Work on electrical systems with voltages of 60 volts or higher may only be carried out by qualified electricians. Persons without the appropriate training are strongly advised to only switch the low voltages of 3 & 5 V available on the board using the relay. Improper handling can result in serious injury or death due to heat, fire or electric shock.

Please adhere to the safety regulations and consult a specialist if you are unsure.

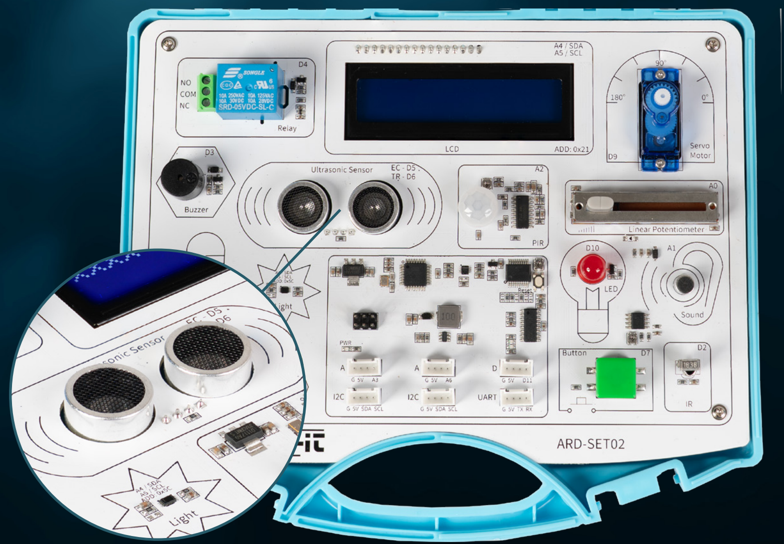
In the code example, the relay is set up as an output. When the corresponding pin is activated, the relay engages and the contact changes its position: the connection between COM and NC is disconnected, while COM is connected to NO. When the pin is deactivated again, the relay returns to its initial position. COM is reconnected to NC and the connection to NO is disconnected.

4.04. ULTRASONIC SENSOR

The ultrasonic sensor can measure distances using ultrasound. For this purpose, the sensor has a trigger and an echo pin. The sensor emits an ultrasonic signal via the trigger pin. It then waits until the echo pin responds, indicating that the ultrasonic sensor has received the signal back. The distance can be calculated based on the elapsed time.

The ultrasonic sensor is connected to pin D5 with the echo pin and to pin D6 with the trigger pin.

Important! The ultrasonic sensor can only measure distances accurately between 2 cm and 3 m. Measurements above or below this range are not reliable.

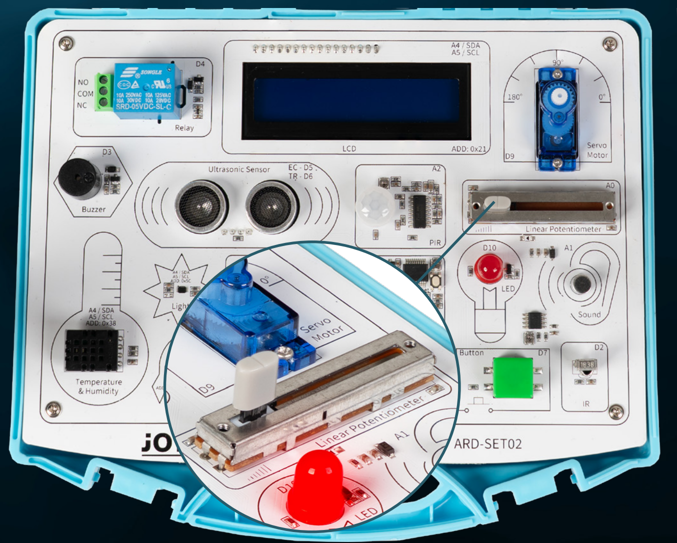


In the code example, the trigger pin is declared as an output and the echo pin as an input. First, the trigger pin is set to a high voltage for 10 µs to trigger the measurement. The code then waits for the echo pin to receive a high value, and the elapsed time is stored. This time can be used to calculate the distance. The code then checks whether the calculated distance is within the valid value range and outputs it.

4.05. POTENTIOMETER

A linear potentiometer is an adjustable resistor that can be used to continuously change electrical values. It is often used to adjust voltage, volume, brightness or position.
The kit measures the voltage supplied by the potentiometer at the analogue input. This voltage depends directly on the position of the potentiometer and changes linearly between 0 V and 5 V. The kit's integrated analogue-to-digital converter (ADC) converts this voltage into a digital value between 0 and 1023. This allows the current position of the potentiometer to be easily evaluated in the programme and used for control or regulation tasks.

The potentiometer is connected to pin A0.



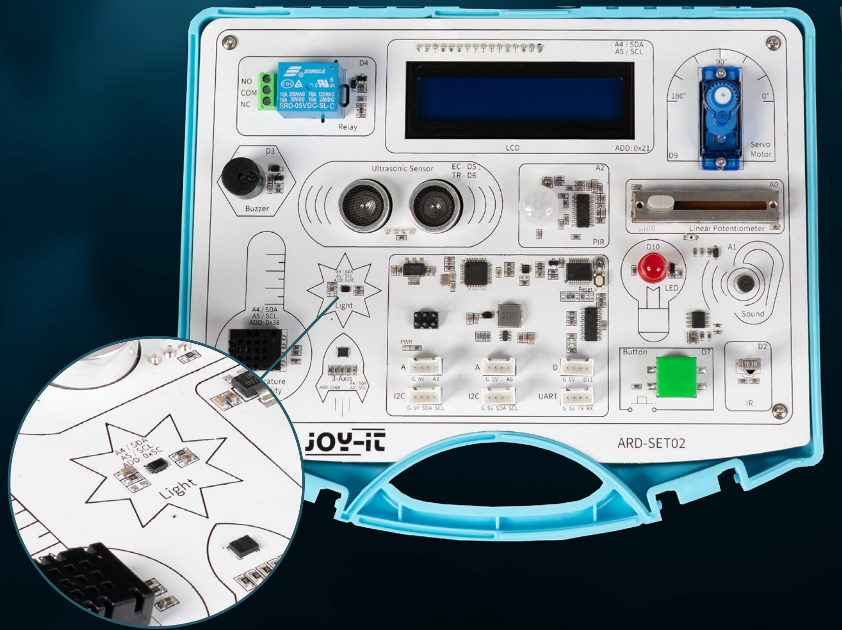
In the code example, pin A0 is declared as an input and read out analogously in a loop. The value read out is between 0 and 1023 and is output via the console.

4.06. LIGHT SENSOR

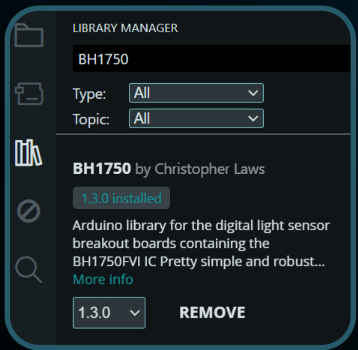
The BH1750 is a digital light sensor that makes it very easy to measure ambient brightness. The measured value is output directly in lux, which makes it particularly beginner-friendly – no conversions are necessary.

The sensor is connected to the kit via the I^2C bus. You can find more information on this in chapter ***4.15 INTERFACES*** *. The BH1750 is ideal for projects such as automatic lighting control, display brightness control or ambient light measurement.*

The light sensor is connected via I^2C and can be addressed using the address 0x5C.



In the code example, the light sensor is controlled using the **BH1750** library.
This was created by Christopher Laws and published under an MIT licence.
You can find and install the library in Library Manager under BH1750.



In the code example, communication with the light sensor is set up via the library. To do this, the communication mode, the I^2C address and the I^2C communication are transferred to the library. Afterwards , the measured lux value can be output using the `readLightLevel()` method .

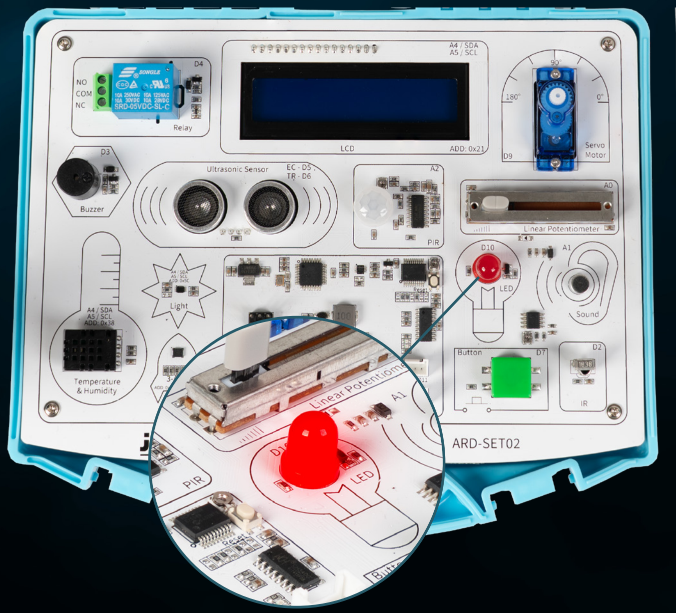
4.07. LED

A red LED is one of the simplest ways to visualise the kit's output. It is often used to indicate conditions such as on/off, status messages or warnings. An LED is ideal for experiments and helps to test programmes and circuits quickly and easily.

<u>The LED can be controlled digitally or via PWM:</u>
If the digital output is set to HIGH in the programme, current flows through the LED and it lights up. If set to LOW, the circuit is interrupted and the LED goes out.

*With **PWM control (pulse width modulation)**, the brightness of an LED **can** be continuously adjusted with the kit. Although a digital output only knows on or off, this allows for smooth brightness control. To do this, the kit switches the output on and off very quickly. The longer the LED is switched on during a switching cycle, the brighter it appears. If it is only switched on briefly, it appears darker. This happens because the effective voltage at the LED is influenced by the PWM signal.*

The LED is connected to pin D10.



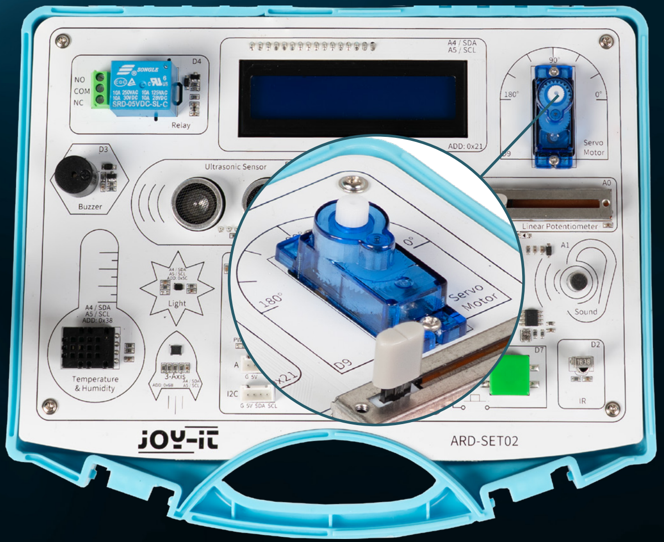
In the code example, the LED is switched on and off digitally. To do this, the LED pin is declared as an output.

4.08. SERVO MOTOR

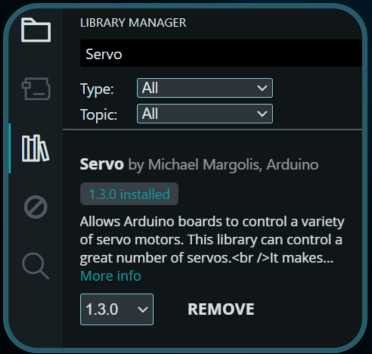
A servo consists of an electric motor with a gearbox and control electronics. On the output side of the gearbox there is a gear wheel onto which the servo horn is mounted. The servo can move the axis within a range of approximately 180°. Servos are used in model making, for example to control the wing or rudder position of an aeroplane or ship. Servos are also increasingly being used in automotive engineering to close doors automatically, for window regulators, mirrors and other adjustable elements.

A servo motor is controlled by the kit via a PWM signal to move to a specific position (0° to 180°). Unlike with LEDs, PWM is not used here to set the brightness, but rather the angle of the servo.

The servo motor is connected to pin D10.



The servo motor is controlled in the code example using the **Servo** library. This was created by Michael Margolis and Arduino and published under the GNU Lesser General Public Licence v2.1.
You can find and install the library in the Library Manager under Servo.



In the code example, the servo motor is controlled via the library. To do this, the pin is connected to the library. In the example, the servo motor rotates between its maximum values.

4.09. SOUND SENSOR

A sound sensor is used to detect noises in the environment, such as clapping, speaking or other sound events. It is well suited for simple projects such as noise controls, clap switches or volume detection.

The sensor consists of a microphone and a small evaluation circuit. It is an analogue sensor, which means that it provides a continuous signal that reflects the volume of the detected sound. The measured value changes depending on the intensity of the sound.

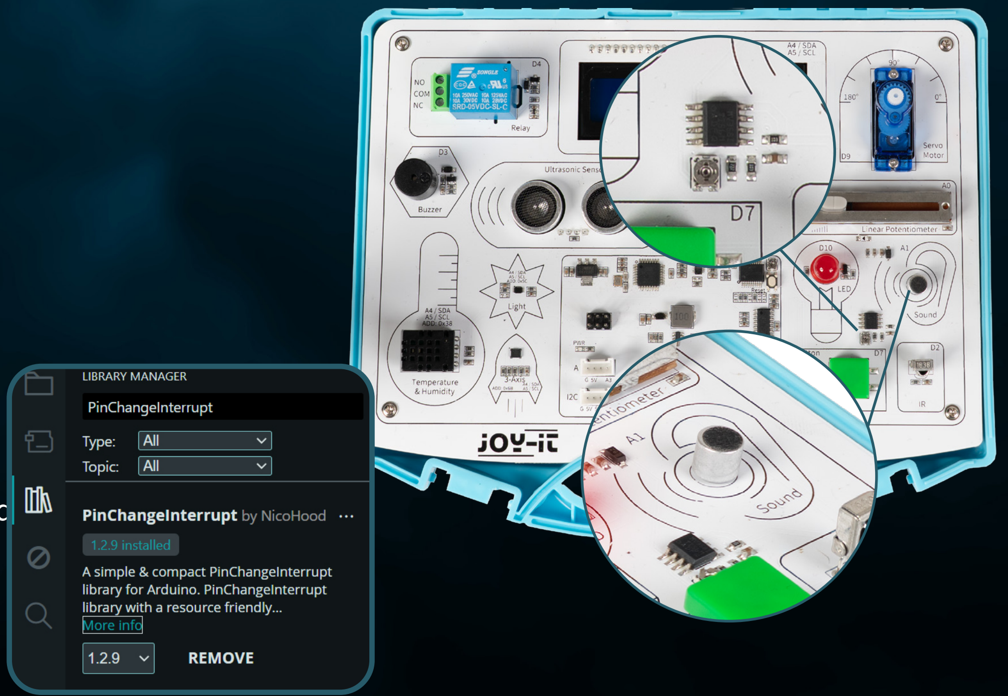
The module also has an integrated potentiometer. This small rotary control can be used to adjust the sensitivity of the sensor. This allows you to specify the volume at which the sensor should respond, which is particularly helpful in environments with varying background noise levels.

The sound sensor is connected to pin A1.

A sound sensor can generate very short sound pulses, for example when clapping or cracking. If the kit only queried the sensor regularly in the programme, these short signals could be overlooked.

An interrupt triggers immediately as soon as the state of a pin changes (LOW ↔ HIGH). The kit thus reacts directly to the detected sound, regardless of the current programme sequence.

The library **PinChangeInterrupt** is used, as the sound sensor is not connected to a specific pin for interrupts. This library makes it possible to use a pin on a port as an interrupt. It was created by NicoHood and published under the MIT licence.



In the code example, the sound sensor pin is connected to a pin-change interrupt as an input. When the sensor detects a sound, the code automatically executes the `soundRecognised()` method .
The sensor itself continues to operate analogously and delivers a continuous signal that reflects the current volume. In this example, however, the analogue signal is only evaluated by the kit as a digital switching signal: it is therefore read as either high or low. If the noise exceeds a certain volume threshold, which can be set using the potentiometer, the output state changes – and it is precisely this change that triggers the interrupt.

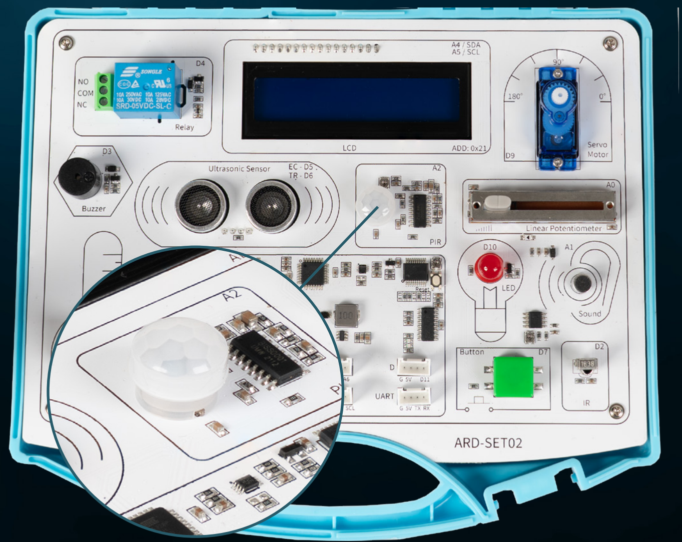
4.10. MOTION SENSOR

A PIR (passive infrared) motion sensor is used to detect the movement of people or animals. It reacts to changes in heat within its detection range and is ideal for motion detectors, alarm systems or automatic lighting controls.

The sensor does not detect movement itself, but rather a change in infrared heat radiation that occurs when a warm object moves in front of the sensor. As soon as movement is detected, the signal output switches to HIGH.

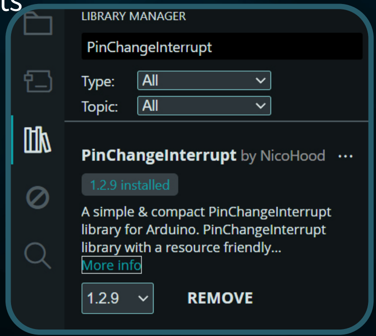
The sensor also has a potentiometer that can be used to adjust the sensitivity of the sensor.

The motion sensor is connected to pin A2.



For the same reasons as with **SOUND SENSOR**, the **PinChangeInterrupt** library is used here.

Important! The sensor has a 2–3 second cool-down period before it detects the next movement. In addition, the PIR sensor requires a short warm-up time after switching on before it can reliably detect movement.

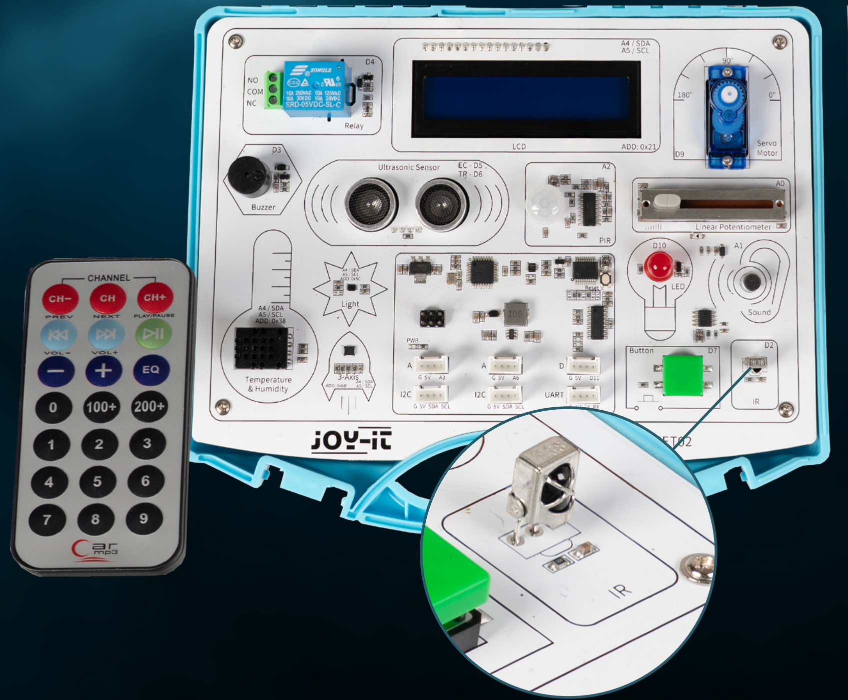


In the code example, the motion sensor pin is linked as an input to a pin-change interrupt. When the sensor detects motion, the code automatically executes the method `movementRecognized()`.
This method queries the current status of the pin. This status indicates whether a movement has just been registered (signal active) or whether the sensor has returned to standby mode after its cool-down phase. During this short pause, the sensor does not respond to new movements in order to prevent false alarms.

4.11. INFRARED SENSOR WITH REMOTE CONTROL

An infrared sensor (IR receiver) is used to receive signals from a remote control and evaluate them with the kit. This allows projects such as LEDs, motors or menus to be conveniently controlled without contact.

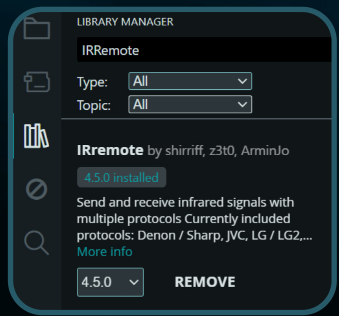
When a button is pressed, the remote control sends infrared light pulses with a specific code. The IR receiver detects these pulses and transmits a digital signal to the kit. Each button on the remote control has its own code, which the kit can distinguish.



The infrared sensor is connected to pin D2.

Note! To use the remote control, you must first insert two AAA-1.5-V-batteries into the remote control. Ensure that the polarity is correct.

The infrared sensor is controlled in the code example using the **IR remote** *library*. This was created by shirriff, z3t0 and ArminJo and published under the MIT licence. You can find and install the library in the Library Manager under IRremote.



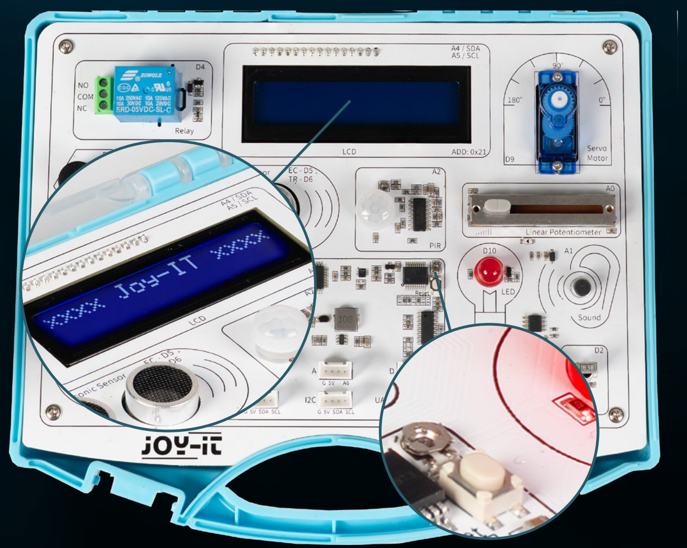
In the code example, the encoding of the remote control is defined using the library. The pin of the infrared sensor is also passed on to the library as input. The method `decode()` returns whether the sensor has received an infrared signal. The received value can be accessed via the variable `.decodedIRData.decodedRawData`. The method `getKey()` converts the received value into the name of the button on the accompanying remote control. Further reading is enabled by the method `resume()`.

4.12. 16X2 LCD SCREEN

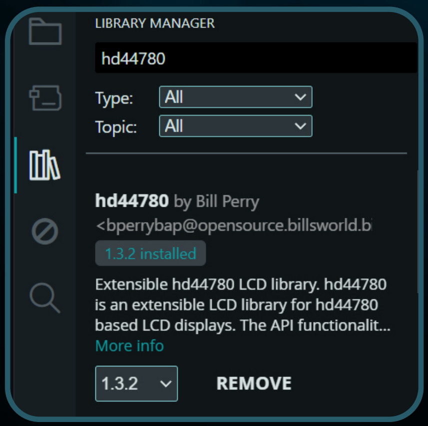
A 16 × 2 LCD screen is a text display that can show 16 characters per line and 2 lines. It is often used to clearly display measured values, status messages or menus. The display is usually based on the HD44780 controller and can be controlled directly with the kit.
*Communication takes place via the I²C bus, where the display is addressed via a fixed I²C address. You can find more information on this in chapter **4.15 INTERFACES**.*

The kit also includes a small potentiometer for adjusting the contrast of the LCD screen. This rotary control can be used to adjust the display contrast to suit individual preferences. If the contrast is set too low, the characters will be difficult or impossible to read; if it is set too high, dark bars will appear on the display. The display can be optimally adjusted by turning the knob carefully.
The potentiometer is located to the left of the red LED on the kit and is therefore easy to find.

The 16x2 LCD screen is connected via I²C and can be addressed using the address 0x21.



The 16×2 LCD screen is controlled in the code example using the **hd44780** library. This was created by Bill Perry and published under the GNU General Public Licence v3.0. You can find and install the library in the Library Manager under hd44780.



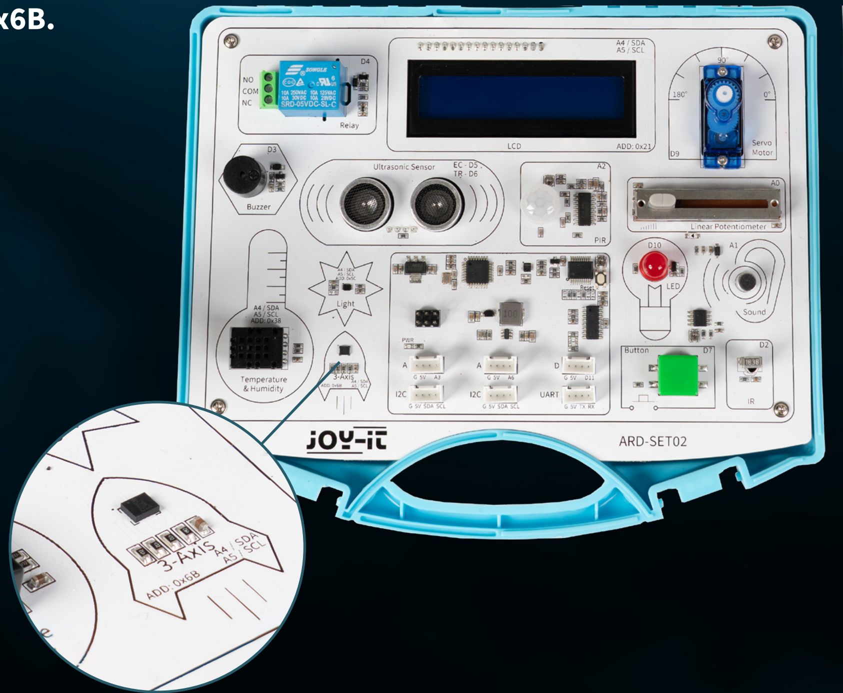
In the code example, the I²C communication and the I²C address of the LCD are given to the library. The library has many ready-made methods for controlling the screen. You can find more information about this in the code example.

4.13. ACCELERATION SENSOR

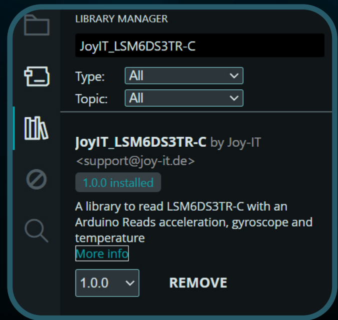
*The LSM6DS3TR-C is a combined accelerometer and gyroscope sensor. It can detect movement, tilt and rotation and is ideal for projects such as position detection, pedometers, motion controls or gesture recognition. The sensor is connected to the kit via the I²C bus. For more information, see chapter **4.15 INTERFACES**.*

The sensor measures linear acceleration along the three spatial axes (x, y, z) and detects both movements and the effect of gravity. This allows inclinations, changes in position, vibrations and shocks to be detected. The sensor also provides measurements when stationary, as it detects gravity.
It also measures the rotational speed around the three axes (x, y, z). These values are output in degrees per second and provide information about how fast the sensor rotates around an axis. This allows rotational movements, rapid changes of direction and rotations to be recorded precisely.
In addition, the sensor can also measure temperature, allowing temperature dependencies of the acceleration and gyroscope measurements to be taken into account.

The acceleration sensor is connected via I²C and can be addressed using the address 0x6B.



The acceleration sensor is controlled in the code example using the **JoyIT_LSM6DS3TR-C** library. This was created by us and published under the MIT licence. You can find and install the library in the Library Manager under JoyIT_LSM6DS3TR-C.

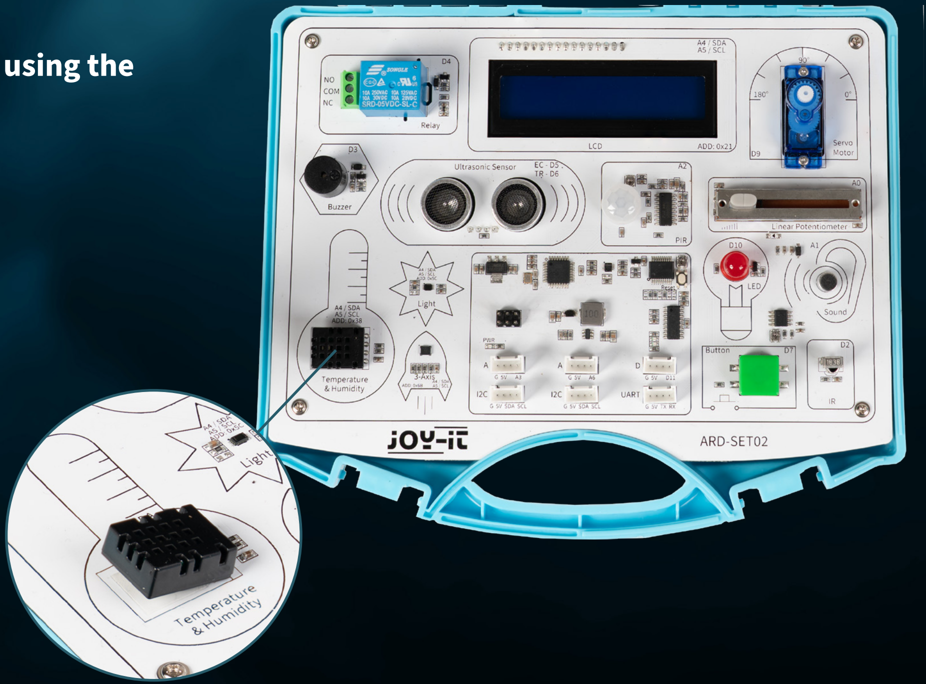


In the code example, the acceleration sensor is controlled via the library. The library also provides objects in which the measured values are stored. `Acceleration` is used to store the acceleration values and `Gyroscope` to store the rotational speed values. In both objects, the values of the individual axes can beaccessed , *for example,* via `.x`. The temperature is stored in a transferred `float`.

4.14. TEMPERATURE AND HUMIDITY SENSOR

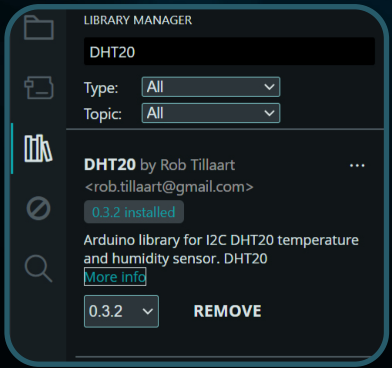
*The DHT20 is a digital temperature and humidity sensor. It reliably measures the ambient temperature and relative humidity and provides the measured values digitally. The connection is made via the I^2C bus, whichis described in more detail in the chapter **4.15 INTERFACES**. The DHT20 typically provides the temperature in degrees Celsius and the humidity in percent (%) relative humidity. The measured values are already processed internally and can be used directly in the programme, for example for display on a screen or for controlling fans and heaters.*

Its compact design and simple control make it ideal for applications such as weather stations, room climate monitoring or smart home projects.



The temperature and humidity sensor is connected via I^2C and can be addressed using the address 0x38.

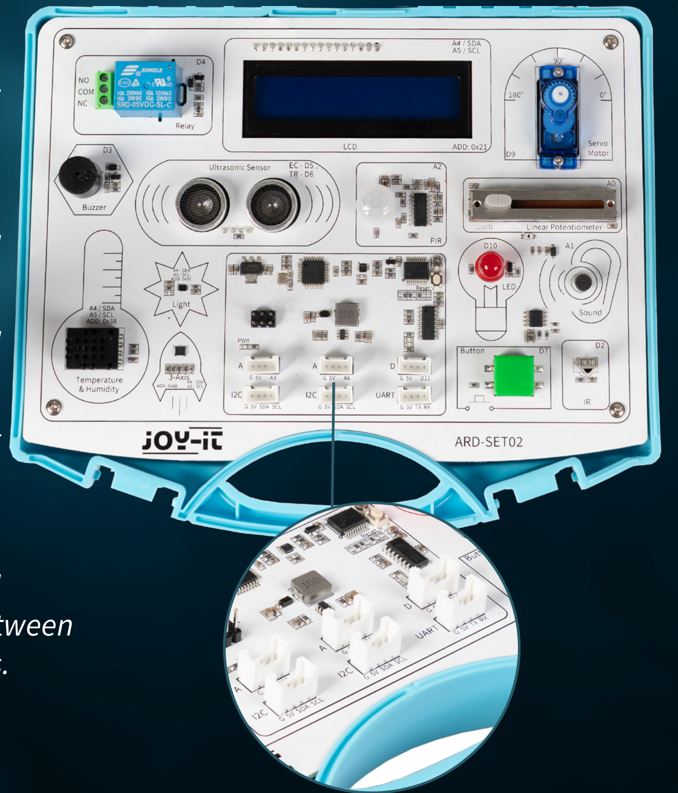
The temperature and humidity sensor is controlled in the code example using the **DHT20** library. This was created by Rob Tillaart and published under the MIT licence. You can find and install the library in the Library Manager under DHT20.



In the code example, the DHT20 is controlled via the library. To do this , the I^2C communication is passed on to the library. By calling the method `.read()`, the sensor reads the data internally. The methods `.getHumidity()` and `.getTemperature()` then return the measured humidity and temperature.

4.15. INTERFACES

Interface connections play a crucial role in the world of electronics, similar to buttons in user interfaces. They enable communication and power supply between different electronic components. Our kit therefore features the following connections in the interface area:



I²C(Inter-Integrated Circuit): *I²C is a serial two-wire interface consisting of a data line (SDA – Serial Data) and a clock line (SCL – Serial Clock). Multiple devices can communicate with each other via these two lines. A so-called master (e.g. a microcontroller such as the education kit) controls the communication and addresses individual slave devices (e.g. sensors and actuators, such as the temperature and humidity sensor) in a targeted manner.*

Each connected device has its own address. The master sends this address to determine which component is currently to be used for data exchange. This allows several sensors, displays or other modules to be operated in parallel on the same two lines.

UART (Universal Asynchronous Receiver/Transmitter): *This interface enables asynchronous serial communication via two lines: TX (Transmit) and RX (Receive). UART is often used for communication between microcontrollers and computers or for connecting modules such as GPS receivers or Bluetooth modules.*

Additional connections: *Pins A3, A6 and D11 have also been led out of the kit so that external sensors or actuators can also be connected. A3 and A6 are analogue pins and can therefore use analogue communication. However, they can also communicate digitally with sensors or actuators. Pin D11 is a digital pin and can therefore only be used for digital communication.*

5 V connections: *These connections provide power for external electronic components. The learning kit has a logic level of 5V. They can be used to supply power to sensors, modules or small actuators, for example.*
The learning kit operates with a logic level of 5 V. This means that the digital inputs and outputs operate with 5 V signals. When an output is activated (high), a 5 V signal is present; when deactivated (low), the signal is 0 V. When connecting external components, it is therefore important to ensure that they are suitable for 5 V logic.

Each of these connections has its specific application and significance in electronics, similar to how different types of buttons in a user interface have different functions. They provide the necessary flexibility and functionality for building and expanding electronic systems.

5. PROJECTS

The various sensors and actuators can be combined with each other. We provide various project ideas with sample code for this purpose. However, there are many other possibilities for using these sensors and actuators together.

HERE you can download all code examples from the projects.

5.01.MAIN PROJECT

This code example combines all of the kit's sensors and actuators in one code. You can use the remote control to select the various projects listed here via the infrared sensor.

You can select the projects using the following buttons.

This code example is significantly more complex than the other projects, as it caused conflicts with the timers available in the kit. In addition, a different import was used for the infrared sensor than in the single code example. This project is well suited for testing the code, but it is significantly more complex to understand.

5.02. SOUND SENSOR &LED

In this project, the state of the LED changes as soon as the sound sensor detects a noise. This can be triggered by clapping, for example. Try programming this project yourself before looking at the example.

5.03. RELAY & BUTTON

In this project, you will switch the relay using the button so that you can execute the relay to write further code. Try programming this project yourself before looking at the example.

In the code example, the button was controlled using the **PinChangeInterrupt** library. Try this in your own code as well.

5.04. MOTION SENSOR & BUZZER

In this project, the buzzer should play a sound as soon as the motion sensor detects movement. This is how an alarm system works, for example. Try programming this project yourself before looking at the example.

5.05. POTENTIOMETER & SERVO MOTOR

In this project, the potentiometer should cause the servo motor to rotate. This means that when the potentiometer is moved all the way to the left, the servo motor is at 0°, and when it is moved all the way to the right, it is at 180°. Try programming this project yourself before looking at the example.

5.06. LIGHT SENSOR &LED

In this project, the light sensor should make the LED shine brighter or dimmer depending on the brightness in the room. It should act as a kind of intelligent light source, recognising that when it gets dark, the light should shine brighter so that you can still see. Try programming this project yourself before looking at the example.

5.07. ULTRASONIC SENSOR & BUZZER

In this project, you will play music using the buzzer. You will control the notes that sound on the buzzer based on the distance measured by the ultrasonic sensor. This is a type of theremin. Try programming this project yourself before looking at the example.

5.08. INFRARED SENSOR & 16X2 LCD SCREEN

In this project, you will display the buttons pressed on the remote control that were received by the infrared sensor on the LCD screen. Try programming this project yourself before looking at the example.

5.09. TEMPERATURE & HUMIDITY SENSOR & 16X2 LCD SCREEN

In this project, you will display the measured values from the temperature and humidity sensor on the 16 × 2 LCD screen. This means you no longer need a computer to view these values. Try programming this project yourself before looking at the example.

5.10. ACCELEROMETER, 16X2 LCD SCREEN & BUTTON

In this project, you will display the measured values of the acceleration sensor on the 16 × 2 LCD screen. This means you no longer need a computer to view these values. Since not all values fit on the screen, you should use the button to switch between the acceleration values, the rotational speed values and the temperature. Try programming this project yourself before looking at the example.

6. INFORMATION AND RETURN OBLIGATIONS

OUR INFORMATION AND TAKE-BACK OBLIGATIONS UNDER THE GERMAN ELECTRICAL AND ELECTRONIC EQUIPMENT ACT (ELEKTROG)

SYMBOL ON ELECTRICAL AND ELECTRONIC EQUIPMENT:
This crossed-out wheelie bin symbol indicates that electrical and electronic equipment should not be disposed of with household waste. You must take old appliances to a collection point. Before disposal, you must remove any old batteries and accumulators that are not enclosed in the old appliance.

RETURN OPTIONS:
As an end user, when purchasing a new device, you can return your old device (which essentially performs the same function as the new one purchased from us) for disposal free of charge. Small appliances with no external dimensions greater than 25 cm can be returned in normal household quantities, regardless of whether a new appliance is purchased.

OPTION TO RETURN ITEMS TO OUR COMPANY LOCATION DURING OPENING HOURS:
SIMAC Electronics GmbH, Pascal Street 8, D-47506 Neukirchen-Vluyn

RETURN OPTION NEAR YOU:
We will send you a parcel label so that you can return the device to us free of charge. To do so, please contact us by email at service@joy-it.net or by telephone.

PACKAGING INFORMATION:
Please pack your old appliance securely for transport. If you do not have suitable packaging material or do not wish to use your own, please contact us and we will send you suitable packaging.

7. SUPPORT

We are also there for you after your purchase. If you still have questions or encounter problems, we are also available to assist you via email, telephone and our ticket support system.

Email: service@joy-it.net
Ticket system: http://support.joy-it.net
Telephone: +49 (0)2845 9360 – 50

For further information, please visit our website:

WWW.JOY-IT.NET