CREATIVITY, COMPUTATIONAL THINKING, INQUIRY-BASED LEARNING FOR INCLUSIVE TECHNOLOGY-ENHANCED ACTIVITIES

LET'S STEAM COURSEBOOK

BEST RECIPES FOR MASTERING YOUR PROGRAMMING PROJECTS WITHIN SECONDARY SCHOOLS

"HELLO, WORLD!"

This coursebook forms part of the Intellectual Outputs and additional work plan deliverables from the "Let's STEAM" project which has received funding from the European Union's Erasmus + programme under grant agreement n°2019-1-FR01-KA201-062946. Specifically, this coursebook is officially referred to as Intellectual Output #2 "*Dedicated raw learning contents for implementation on diverse learning tools & assessment*". Let's STEAM aims at developing training of teachers' programme dedicated to computational thinking and creativity skills using IoT board and digital tools at a larger scale. The project runs from September 2019 to August 2022.

It involves 8 partners and is coordinated by Aix-Marseille Université. More information on the project can be found on the project website: www.lets-steam.eu.

CONTRIBUTORS

Authors
Jonathan Baudin, Toon Callens, Roberto Canonico, Mercè Gisbert Cervera, Carme Grimalt-Álvaro, Georgios Mavromanolakis, Sébastien Nedjar, Maryna Rafalska, Margarida Romero, Despoina Schina, Cindy Smits, Lorena Tovar, Stéphane Vassort, Eleni Vordos

Chief Editor
Manon Ballester

Graphics & Design
Manon Ballester

EU project consortium
The contributors credited in this coursebook form part of the Let's STEAM consortium that you can discover here: www.lets-steam.eu/our-project-heroes

DESIGN & CREDITS

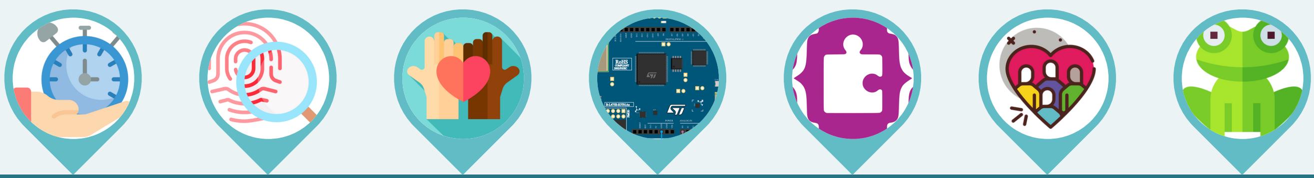
Screenshots
From the authors
makecode.lets-steam.eu
github.com/microsoft

Cover and illustrations
Icon made by Freepik from www.flaticon.com

LICENCE & RIGHTS

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

PARTS & CHAPTERS



ACTIVITY SHEETS & IDEAS

INTRODUCTION

DISCOVER THE LET'S STEAM PATHWAY

The main goal of the training materials you will find in this coursebook and across our e-learning platform is to raise the interest and understanding of all the teachers, independently from their background and technical expertise and appetence, regarding the creation of new content and activities using programming boards and coding practices in a creative way. The "Let's STEAM" coursebook has been created within a European project and this coursebook and activities refer to "Let's STEAM" as the training program and activities that will help teachers to understand and integrate programming boards in a creative learning way.

Algorithmics and coding practices can be highly relevant for approaching the challenges of our current society. Indeed, better knowledge of our surrounding environment is linked to the availability and comparison of datasets that are relevant to physical, environmental, chemical or ecosystemic concepts for instance and constituting them using sensors and programming is a powerful activity for the students, enabling them to concretely understand STEAM topics (*Science, Technology, Engineering, Arts & Mathematics*).

In order to benefit from all the possibilities offered by programmable boards and data collection, Let's STEAM training aims to use the learning of programming as a tool for teaching, in the service of creativity, and curiosity for science beyond being an isolated educational theme. Promoting an active and interdisciplinary pedagogy directed towards the students is therefore one of the fundamentals of our approach. Moreover, in the framework of the promotion of citizen science practices, Let's STEAM training wishes to offer the possibility to use these boards and sensors in order to engage students in a participative scientific approach, being then an effective way to motivate students towards scientific and technical learning.

In parallel, the Let's STEAM training also aims to address one of the major challenges related to the development of technical and technological activities in schools. Indeed, we wanted to offer our readers the opportunity to reflect on the crucial issues of ethics, inclusion and equity through additional concrete and practical resources in this handbook. Although of great importance, these issues are usually not or not adequately addressed in digital literacy training, even though it is a real challenge to stimulate students' motivation, interest and curiosity in science while considering all learners' needs.

These multiple technical and non-technical objectives have been translated by the authors of this manual, members of the Let's STEAM consortium, into a flexible and interdisciplinary methodology that is implemented throughout the content of our course. Understanding the needs of teachers in order to develop hands-on, motivating, inclusive and creative activity is, therefore, an essential aspect of the Let's STEAM approach. In concrete terms, this translates into a general and adaptable framework based on a pedagogical approach through experimentation, data collection and analysis, questioning, illustrated by the authors and supported by practical resources.

Reflecting this approach, the Let's STEAM teacher training programme has been built on both a **theoretical approach** (**PART I**), but also **concrete tools, tutorials and models** (**PART II**) to deepen the knowledge and quickly put into practice the skills acquired here in the classroom. Each part takes up the three essential axes that constitute our approach, namely: ***inquiry-based learning, programming as a tool for STEAM learning, and ethical and inclusion issues in techno creative activities***.

PART I - THEORY - GET FAMILIAR WITH THE LET'S STEAM CONCEPTS AND APPROACHES

The first part of the handbook is to discuss with readers/teachers the triptych of interconnected concepts on which the whole Let's STEAM approach is based. These concepts will be approached in a succinct and factual manner, keeping in mind the pillars of the training and including the following questions:

- How can we create **inquiry-based learning activities** that promote **meaningful and interdisciplinary** content for students in **technology-enhanced education**?
- How can we create **inclusive activities** to ensure the **motivation and interest** of all students and promote content that **goes beyond stereotypes**?
- How to **develop mastery of programming practices** both so that **teachers** feel more comfortable in launching **large-scale interdisciplinary projects** using programming as a tool, but also, in the service of **their students**, to better understand programming as an excellent way to **address societal challenges** in a more advanced way?

This part of the manual is therefore divided into three chapters:

The first chapter "***Deepening your knowledge and use of the inquiry-based approach***" focuses on understanding the steps involved in an inquiry pedagogical methodology in order to replicate it in technology-enhanced activities. This theoretical chapter will be complemented by a set of additional practical resources to develop your own teaching materials related to the implementation of programming-based activities in your classroom and to draw on examples provided in this handbook that address the phases of inquiry-based learning.

The second chapter "***Reflecting on inclusiveness and equity while conceiving a technology-enhanced activity***" discusses the basic concepts and definitions that are essential for developing more inclusive activities that can help to stimulate the interest and curiosity of your students, adapted to the contexts and educational needs of learners and schools. It will be complemented in the second part by concrete activities to stimulate everyone's thinking around this subject which can be complex to tackle.

Finally, the chapter "***Basics of programming - software and hardware***" is intended to introduce teachers/readers to the MakeCode editor and the STM32 board, which are used in the activity sheets found in this manual. It aims to familiarise learners with the programming learning platforms, and with the STM32 Discovery board, which has been chosen for its technical capabilities and its set of integrated sensors, allowing the development of complex experimental projects, stimulating students' interest and creativity. Once the knowledge is acquired, this chapter can be a good introduction for your own students to introduce them to the programming tools and associated features.

PART II - PRACTICAL APPLICATION - ACTIVITY SHEETS AND TEMPLATES

Once you are familiar with the three concepts that form the core of the Let's STEAM approach, it will be time to put all this knowledge into practice with the help of activity sheets on the one hand and outlines and examples on the other.

ACTIVITY SHEET. In this second part, you will find two sets of activity sheets that can be used for training purposes and directly implemented in your classroom:

- The first series "***Programming easily thanks to let's steam activity sheets***" introduces you to the programming and use of sensors and programmable boards. Through 15 different projects, you will approach various functions and components of the electronic board (and in particular of the sensors) in order to discover their potential from concrete and specific practices (such as breadboarding, making an LED blink, creating a readable thermometer with the embedded sensor and a basic screen).

- The second set of activity sheets "***Inclusion and equity: resources for students & trainers***" allows you to work on turning your technology activity into an inclusive project. This is made possible through a number of reflective activities that can be carried out either on your own using the templates provided, or with the help of your Let's STEAM ambassadors (your local contact is given at the end of this manual), or with your colleagues and/or pupils.

TEMPLATE AND EXAMPLES. Eventually, all the knowledge and activity sheets are collected in a reproducible template "***Replicate IBL in your classroom - Guidelines & Template***" allowing you to build your own learning pathway, using Let's STEAM resources. It is strongly recommended that you use and revisit all the resources presented in this manual on a regular basis in order to achieve a good balance between your societal approach and the technical skills you bring to your students on programming.

Feel free to re-use all or part of this manual, whether it is the theoretical concepts or the activity sheets and templates, in your classroom by using the activities as inspiration, copying the activity sheets for direct use by your students and creating your own lesson plan! Our content has been developed entirely under a Creative Commons license. This license gives you the right to use this content for your own materials!

By following the proposed pathway, you will be introduced to programming in a progressive way throughout the Let's STEAM course and will carry out activities of increasing difficulty. You will have the opportunity to apply the technical knowledge acquired through the programming activity sheets to the design of educational material by following the development and content creation steps based on the experimentation phases. This will make your activities more meaningful and inclusive for all your students!

Let's your Let's STEAM adventure begin!

PART I

GET FAMILIAR WITH THE LET'S STEAM CONCEPTS AND APPROACHES

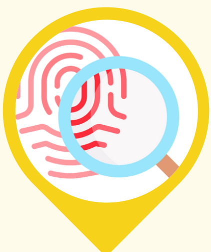
Feel free to reuse the content in this section to introduce these concepts in your classroom! You are free to print, reproduce, modify, reuse and draw inspiration from all the resources in this manual without restriction. Our content has been developed entirely under a Creative Commons license.

CHAPTER 1

DEEPENING YOUR KNOWLEDGE AND USE OF THE INQUIRY-BASED APPROACH

Authors: Georgios Mavromanolakis, Despoina Schina, Stéphane Vassort

To understand and reuse the IBL (Inquiry-Based Learning) approach, the Let's STEAM materials have been designed to approach the training resources without the already-made solution. Our goal is to help you develop your own solutions to problems you would want to solve with your students in the classroom.



Inquiry-based learning (IBL) is an educational flexible strategy with phases that are often organized in a cycle and divided into subphases with logical connections depending on the context under investigation (Pedaste et al., 2015 - Margus Pedaste et al. Phases of inquiry-based learning: Definitions and the inquiry cycle, Educational Research Review, Volume 14, 2015, Pages 47-61, ISSN 1747-938X, https://doi.org/10.1016/j.edurev.2015.02.003). This framework entails five general phases (Orientation, Conceptualization, Investigation, Conclusion and Discussion) and seven sub-phases (Questioning, Hypothesis Generation, Exploration, Experimentation, Data Interpretation, Reflection, and Communication).

FOCUS ON AN INQUIRY-BASED LEARNING APPROACH (OR IBL)

IBL can be used in order to conceptualize a structured way to implement inquiry activities and develop multidisciplinary educational projects in the classrooms. IBL is not a linear procedure and learners should be involved with various forms of inquiry, going through different combinations of the phases, not all of them necessarily. For example, if the data analysis is not satisfactory enough, students can return to the conceptualization phase and reconsider their question and/or their experimental design. When students come to a conclusion, new questions can be generated, and the process starts again in a progressive fashion. A description of the processes of IBL by Pedaste et al. comprises the five phases described below:

- **Orientation**: Orientation is the phase where the identification of the problem occurs. The topic to be investigated is presented and interest in a problematic situation that can be answered with inquiry is stimulated. The topic under investigation must be relevant to students' daily life, interests and prior knowledge. The teacher's role in this phase is to encourage students to express ideas, prior knowledge and questions about the topic while promoting interaction and communication between them. For example, students can create concept maps of what they know, do not know or want to know about the topic under investigation. These kinds of activities can also be useful for the next phases of inquiry.

- **Conceptualization**: Conceptualization refers to the understanding of the concept, which relates to the problematic situation presented in the previous phase. It is divided into two subphases (questioning and hypothesis generation) that lead the learner to the investigation phase. The teacher's role in this phase is to help students understand how they can formulate questions and/or hypotheses that can lead to an investigation. If students are not familiar with the questioning and hypothesis generation subphases, the teacher can choose a structured type of inquiry at first and then progress in more open types of inquiry in order to provide the appropriate guidance.

    - ***Questioning subphase***: Questions are formulated in order to design an investigation that produces answers. As this skill is developed through inquiry, students can gradually understand which question can lead to investigation and which one is more generative and might lead to different or richer processes.

    - ***Hypothesis Generation subphase***: A hypothesis is generated by providing explanations of how the identified variables relate (Pedaste et al., 2015). It explains how and why phenomenon function based on former experiences and prior knowledge.

- **Investigation**: Investigation is the phase where students collect evidence in order to answer their questions and/or test their hypothesis and includes the subphases of exploration, experimentation, and data interpretation. The teacher provides materials that the students might need and keeps them on track so that the process they choose to follow is a process that answers the investigative question. Students should determine what constitutes evidence and collect it. If they are not familiar with this process, a structured type of inquiry can be chosen. The teacher can provide or encourage students to create means (e.g. tables, charts, etc.) that can help them organize, classify and analyze the data.

- ***Exploration subphase***: Exploration is an open process that generates mostly data concerning the identification of a relationship between the variables. It is chosen typically when the question that was formed in the previous phase was generative because students do not have a specific idea of what to explore or how the identified variables relate to each other (Pedaste et al., 2015).

- ***Experimentation subphase***: Experimentation includes the design (e.g. choosing the materials and means to measure) and performing of experiments taking into consideration the variables that need to change, remain constant and be measured. The products of this subphase are data or evidence that can be used later on for analysis and interpretation.

- ***Data Interpretation subphase*:** Depending on the concept under investigation and the inquiry procedures that were chosen, finding relations between the variables is sometimes the key to getting the desired outcome (answering the investigative question). Organizing and classifying the data (with graphs, charts, tables, pictures, etc.) can benefit this process.

- **Conclusion**: In this phase, students draw conclusions based on the investigative question and the interpretation of the data. The teacher's role during this phase, a comparison between the interpreted data and the predictions and initial ideas (that the students expressed during the orientation phase) can be stimulated. This process can also lead to new hypotheses and questions about the topic under investigation.

- **Discussion**: During the discussion phase students articulate their findings through communicating them to others and/or reflecting upon all or some of the stages of inquiry during the processor by the end of it (Pedaste et al., 2015). The teacher's role is to encourage collaboration so that students can present their findings and ideas, provide arguments and give feedback to others. If they are not familiar with these practices, the teacher can provide guidelines that will help them to communicate during all the phases of inquiry.

    - ***Communication subphase***: Communication includes a discussion with others and representation of results in a manner that is understandable to all (National Science Foundation, 2000). It can be applied to a single phase or the whole cycle of inquiry and is usually an external process (Pedaste et al., 2015).

    - ***Reflection subphase***: In this subphase, students reflect on their work, their results and the concept under investigation. Reflection can even give rise to new thoughts regarding the inquiry cycle or a single phase.

TYPES OF INQUIRY

The types of inquiry vary so that students are actively involved in the process to the extent that they are competent and able to do so. The type of inquiry a teacher may choose to follow is highly dependent on the objectives of the lesson, the age of the students, their previous involvement with inquiry and the scientific skills they have already acquired. As shown below, the more responsibility the student has, the less direction is provided and the more open the inquiry becomes.

The variations of inquiry types concern the increasing or decreasing involvement of the teacher and student in the process. Structured inquiry is directed from the teacher so that students reach a specific result, whereas in mixed inquiry students are more involved during an investigation with the teacher guidance still being the most dominant. These forms of inquiry usually are chosen when students are first introduced to inquiry practices and when there is a focus on the development of a specific skill or concept. Open inquiry provides more opportunities for developing scientific skills, given that during the open inquiry the students work directly with the materials and practices in a way that resembles authentic scientific approaches.

For example, if students lack previous experiences with designing investigations and collecting data, a more structured or guided form of inquiry should be chosen. When students acquire the skills needed, they can progress to more open inquiry activities. Students should at some point participate in all the forms of inquiry, while gradually moving from one form of inquiry to another with the simultaneous progression of complexity and self-direction.

GUIDE... WITHOUT LEADING - IBL ADAPTED TO THE LET'S STEAM MATERIALS

To understand and reuse the IBL approach, the Let's STEAM materials have been designed to approach the training resources without the already-made solution. Our goal is to help you develop your own solutions to problems you would want to solve with your students in the classroom. The appropriation of the work will be more important and will facilitate the future transfer to your classes. For inspiring you, several problems can be offered in order to address the different fields of STEAM but also to target the potential interests of your classroom. You will hence find in this coursebook, in addition to our template, a pool of problems.

And keep in mind that by using the template and associated resources you will find in the second part of this coursebook, you are also a great contributor to the Let's STEAM materials! We invite you to share your productions with the Let's STEAM community and beyond!

CHAPTER 2

REFLECTING ON INCLUSIVENESS AND EQUITY WHILE CONCEIVING A TECHNOLOGY-ENHANCED ACTIVITY

Authors: Mercè Gisbert Cervera, Carme Grimalt-Álvaro

Today's technologies, apart from having many educational and learning benefits, present some new security and ethical challenges, which might be necessary to consider. As Let's STEAM activities will be implemented in very different educational contexts, it becomes necessary to appropriately reflect on how these implementations will be carried out. We argue that this reflection should be carried out with the aim of promoting the engagement of all students and, therefore, ensuring inclusive STEAM teaching and learning practices, adapted to the educational contexts and needs of learners.

By ethics, we understand the appropriate and acceptable behaviour in relation to Digital Technology (DT) practices and Internet usage. Computer or digital ethics deal with, for example, the unauthorized use of computer systems, software theft (piracy), information privacy, unauthorized collection, use of information copyright... The responsible and ethical use of DT is an important part of trainees' work and students' learning and, for this reason, it is evidenced in many national curricula.

Safe use of the Internet is one of the main concerns of school curricula in many countries, as adolescents need to be safe when using information technology for learning and in their daily lives. To promote the safe use of information technology, we need to know how our students use the Internet and information technology, and what risks they may encounter online (e.g. harmful content online, online radicalisation and extremism, risks related to personal content sharing and sexting, online harassment, etc.). Although promoting safe use of the internet and digital communication is a broad topic to cover, we want to use this training to bring to the table some of the issues around digital communication that may prevent our students from taking advantage of digital communication for learning in their daily lives.

Some students are disadvantaged and have fewer opportunities than their classmates. Our aim as teachers or educators is to ensure that all students have equal learning opportunities to develop their potential and abilities. In the STEM fields (science, technology, engineering and mathematics), there are particular issues that reduce the learning opportunities of some students, which can sometimes be hindered by our daily teaching practices. By discussing Let's STEAM activities and the ethical and safe use of digital technologies for our students, we would therefore like to contribute to the creation of more equitable and inclusive educational activities.

It is therefore important to recall the definitions of some of the concepts we will use frequently in this manual and in the activity sheets proposed in this training, as terms such as "**equity**" and "**inclusion**" can be confusing and mean different things to different people:

INCLUSION

In the words of UNESCO, inclusion means ensuring that each individual has an equal opportunity for educational progress remains a challenge worldwide. It is increasingly seen as a principle that supports and welcomes diversity amongst all learners (UNESCO 2017). This view presumes that the aim is to eliminate social exclusion resulting from discriminatory attitudes about race, social class, ethnicity, religion, gender, and ability. However, in common language, inclusion is usually used focusing on the integration of students with special needs.

SPECIAL NEEDS

We consider students with special needs those students with learning problems or disabilities that make it more difficult for them to learn than most students their age. These limitations may include (but are not limited to) disadvantages in physical, behavioural, intellectual, emotional and social capacities (UNESCO) (e.g. autism, Asperger, down syndrome, dyslexia, dyscalculia, dyspraxia, dysgraphia, blindness, deafness, ADHD, etc.). They require additional support and adaptive pedagogical methods in order to participate and meet learning objectives in an educational programme.

EQUITY

Equity is an approach that ensures everyone accesses the same opportunities. Equity recognizes that advantages and barriers exist, and that, as a result, we all don't all start from the same place. Equity is a process that begins by acknowledging that unequal starting place and makes a commitment to correct and address the imbalance. Hence, practices promoting equity not only try to engage students with special needs but also many other students who may have fewer learning opportunities. In the STEM field, these students are mostly:

- Girls/female students. As literature has evidenced, STEM fields are socially constructed as masculine items and girls can feel detached from them.
- Students from racial/ethnic minorities. Literature has also evidenced how the social reference of a "STEM person" is a white and brilliant man, which usually serves as a detaching factor for racial minorities.
- Students from low and high socioeconomic backgrounds. Again, the socially accepted image of a STEM person is a middle-class man. Students from low socioeconomic backgrounds might also face economic hardships which may prevent them from developing STEM-related trajectories.

OBJECTIVES OF APPROACHING INCLUSIVENESS AND ETHICS IN THE LET'S STEAM TRAINING

Discussing and reflecting about ethics in the course of the Let's STEAM training will enable you to analyse and transform the designed educational materials and activities to adapt them to students' needs and increase equity and inclusion in learning. This will mainly enable to:

- Create and apply new and different teaching strategies to promote an inclusive and equitable learning environment in the implementation of Let's STEAM activities.
- Apply the knowledge acquired of safety, ethics, and security to identify potential issues in the use of digital technologies by students.

To these ends, the resources you will find in the second part of the manual are structured in two:

- A first set of activity sheets, templates and canvas aim at adapting and improving the design and implementation of your activities so a **more inclusive and equitable learning environment can be promoted**. It is expected that trainees would progressively become aware of their students' needs, adapt the design of the Let's STEAM activities to their educational contexts, reflect on the possible issues in the implementation, and transform their teaching practices to address these issues.

This include R2AS1 "Inclusive Design" and R2AS2 "Inclusive implementation" and their annexes.

- The second part takes a more **general approach to build a comprehensive perspective of ethics and security** with digital technologies in classrooms. Supporting materials are provided, although trainees will be also invited to develop their own resources.

This include R2AS3 "Ethics & Security" and R2AS4 "Promoting & Sharing".

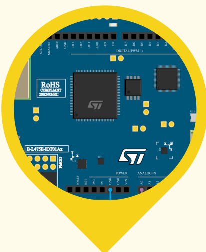
We are aware that Let's STEAM activities will be implemented in very different educational contexts. For this reason, the resources are designed as a flexible proposal. We aim to ensure a proper fit of the resources with the students' needs where Let's STEAM activities will be implemented. They are organised in activity sheets, that can be used for the purpose of this training of trainers programme or directly within the classroom, separating the guidelines for the learners and the ones for the trainers/teachers.

CHAPTER 3

BASICS OF PROGRAMMING - SOFTWARE AND HARDWARE

Authors :Jonathan Baudin, Sébastien Nedjar

As you are now aware from the previous chapters of the pedagogical pillars of the Let's STEAM approach (inclusion, equity, experiential approach), we propose to introduce you to the programming learning tools that are used in our activity proposals: the MakeCode editor and the STM32 board. This presentation will give you the initial information to start your projects with these software and hardware tools.



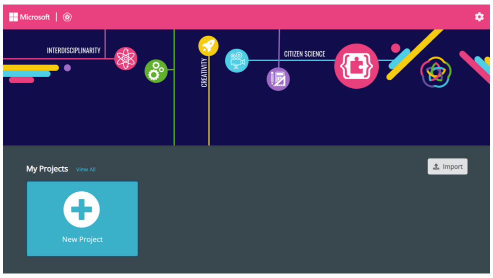
Technological choices made in this coursebook are proposed as they have a real pedagogical interest in the framework of the deployment of large and challenging projects using programming in secondary schools, from lower to higher levels. Specifically, this chapter will approach:

- **The Microsoft MakeCode editor**: a free, open-source platform for creating engaging computer science learning experiences that support a progression path into real-world programming. To access the Let's STEAM MakeCode follow this link: https://makecode.lets-steam.eu/
- The **STM32 IoT Node Board:** a board embedding interesting and relevant sensors and tools, useful for experimenting with challenging projects in the classroom.

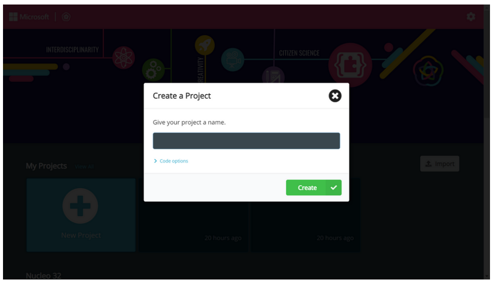
DISCOVER THE MAKECODE SOFTWARE SOLUTION FOR LEARNING PROGRAMMING

TAKE A TOUR OF MAKECODE

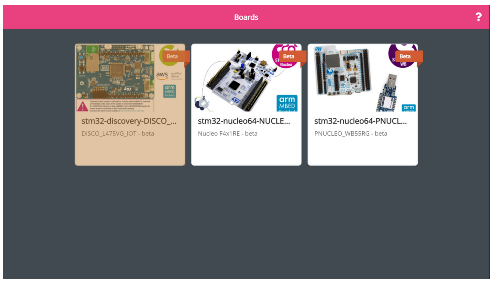
When you enter the MakeCode Let's STEAM website, you will directly land on the homepage. On this page, you can create a new project, open an existing project if you have been working on the editor before, view the supported boards and discover inspiring resources.



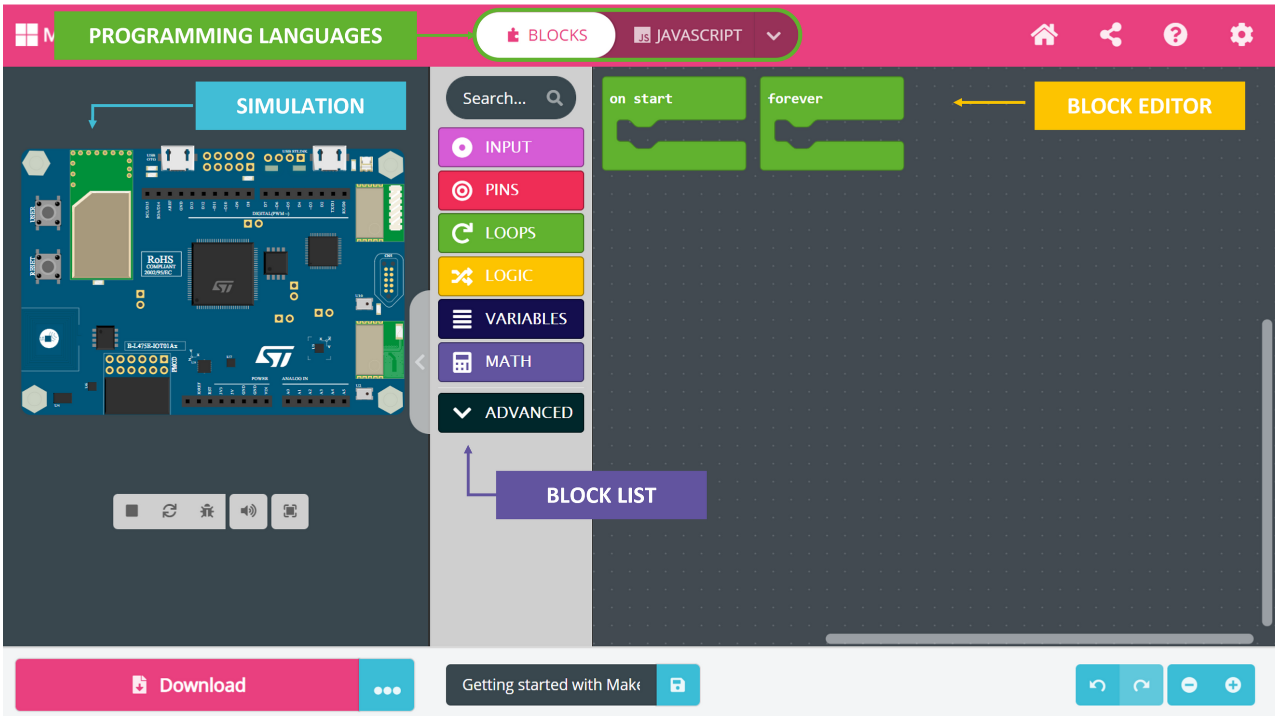
- When you are creating a project, it is important to **name it with a clear and understandable title**, enabling you to express what will be the purpose of the program.



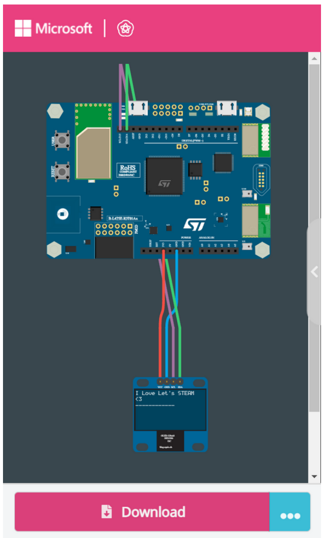
- The next screen will request you to **choose the board on which you will work**. On the Let's STEAM activity sheets, all the examples have been developed using the STM32 IoT Node Board (the board is highlighted in orange in the picture presented here).



Once the board is selected, you will then have access to the editor, with three parts as shown hereunder:

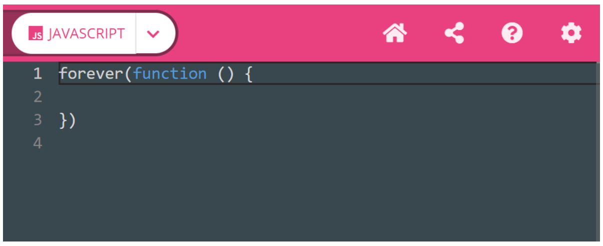


Here are the basic components of your editor:

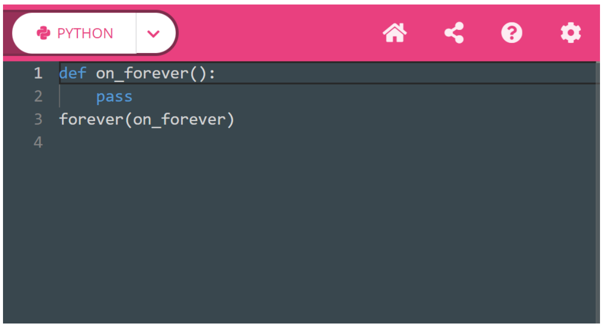


- The **SIMULATOR** (on the left side of the editor): an interactive simulator provides students with immediate feedback on how their program is running and enable them to test and debug their code.
- The **BLOCK LIST** in the middle, that can be used in your program to search for functions.
- The **BLOCK EDITOR** on the right part, that includes already 2 functions common to all activities: on start & forever loop. Students new to coding can start with coloured blocks that they can drag and drop onto their workspace to construct their programs.

In the editor, you will also be able to choose the way of programming i.e.:



- **Through blocks** (see activity sheet R1AS1 - Blink a LED)
- **Through JavaScript editor** (all the activity sheets proposed in this coursebook will include the code in JavaScript that can directly be copy-pasted in this specific editor)



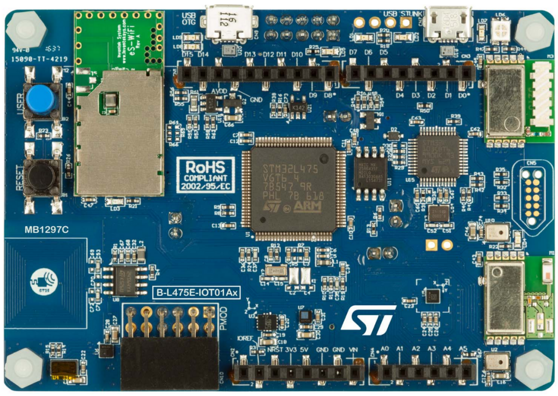
- **Through Python language** for more advanced students.

Even if you will have more precise insights on each block function in the diverse activity sheets proposed in this coursebook, here is the basic blocklist available that can be found on the Let's STEAM MakeCode editor:

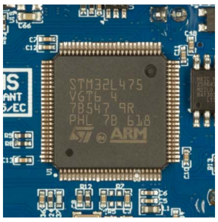
Input	INPUT	Use sensors in your programme (such as buttons, thermometer)
Pins	PINS	Interact directly with the pins and change their status (from low to high, from on to off)
Control	CONTROL	Manage the execution of events
Loops	LOOPS	Implement repetitions
Logic	LOGIC	Perform tests, comparision and boolean logic operations
Variables	VARIABLES	Create variables and counters
Math	MATH	Perform diverse mathematical calculations
Functions	FUNCTIONS	Create subprogrammes
Arrays	ARRAYS	Create a value or text in a table
Text	TEXT	Modify texts
Console	CONSOLE	Display data
Extensions	EXTENSIONS	Access the list of extensions available in the MakeCode version
Datalogger	DATALOGGER	Create a dataset to store the data from the sensors
LCD Screen	LCD	Display text or information on a screen (LCD)
OLED Screen	OLED	Display text or information on a screen (OLED)

DISCOVER THE STM32 IOT NODE BOARD & ITS SET OF SENSORS

The ***"STM32 Iot Node Board"*** is a programming board, which means it allows a user to create a programme and put it inside the board.

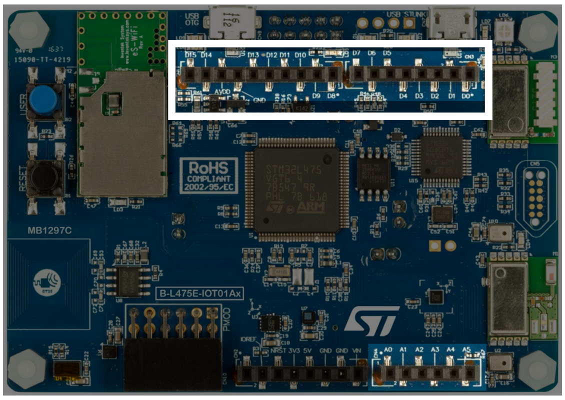


To execute this programme, you need a "microcontroller", i.e. the brain of the board (visible on our board in the middle - the big black square). The name of our microcontroller is: **STM32L475VG.**



THE GPIOs

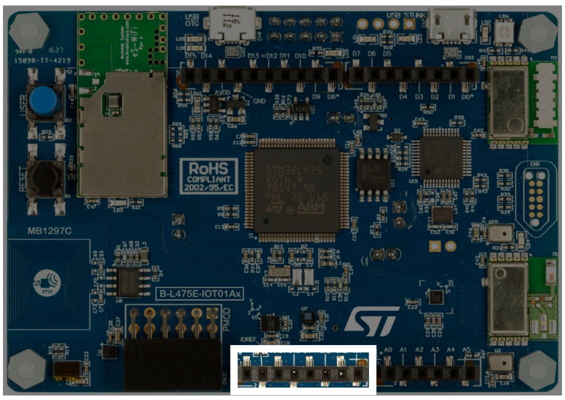
As you can see, there are lots of "legs" or "**pins**" around the microcontroller, called "General Purpose Input / Output" (or GPIO in short). Basically, you can use them to interact with the outer world. Even if there are lots of GPIOs, you cannot use all of them. The usable GPIOs are located on the top and bottom of the board.



There are these black rectangles with holes in them, called "**pinouts blocks**". If you look closely, you can observe some inscriptions around (D0, D1, D2, D3, ..., A0, A1, A2, ...). These inscriptions are the names of the GPIOs.

We will discover the differences between Ax pins (A0, A1, ...) and Dx pins (D0, D1, D2, ...), further in the activities.

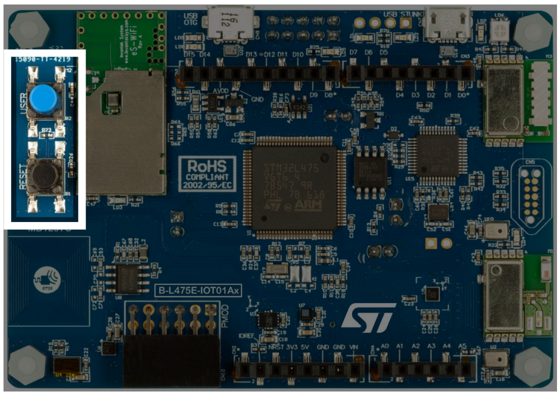
Another pinout block remains which is a "**power pinout block**". You can use these pinouts to power your sensors or actuators (like motor, light, and lots of different things).
The inscription on top of the pinout block, inform us how to use it. The "5V" is like the "+" (positive pole) of a battery and the "**GND**" (short for "Ground") is the "-" (negative pole).



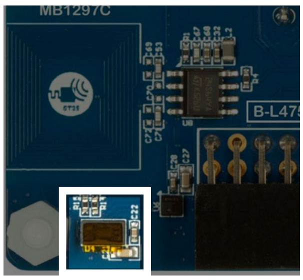
THE PERIPHERALS

The difference between the number of GPIOs available through the pinout block and the number of legs of the microcontroller can be explained by the presence of multiple peripherals already connected to the microcontroller, available on the "STM32 IoT Node Board" itself. The presence of all these peripherals makes this specific board very attractive, as it will enable you to implement a large range of activities, from simple to complex, and from basic to playful. This is a real asset for performing engaging activities in the classroom.

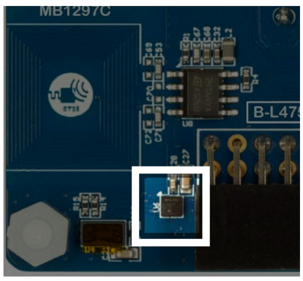
- **BUTTONS:** On the left side of the board, you can find two buttons. The black one is the **RESET** button, enabling the program to restart if you need it. The other one can be used in our program to de**tect when the user push-it** (short push, long push, release it, etc). It can be useful for creating simple user interactions, such as a quiz button for organising competitions using the board.



- **DISTANCE SENSOR**: Let's have a look at the bottom on the lower-left corner of the board. Just on the right of the nylon screw, you can find a sensor to measure distance. It is officially called the "**time of flight**" because it measures the time it takes for a laser beam to travel back and forth (***fly***) from the sensor to an object.



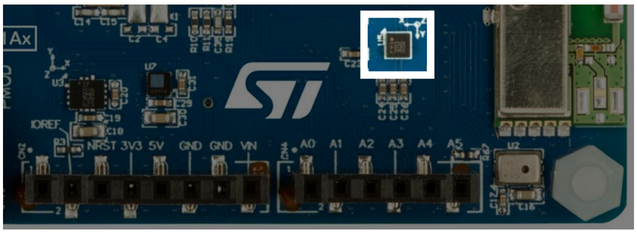
- **TEMPERATURE & HUMIDITY SENSOR**: Next to the "time of flight" sensor on the right, you can find a thermometer and hygrometer sensor ("2 in 1"). This can be useful to implement activities linked to the monitoring of heat or to approaching meteorological concepts.



- **ACCELEROMETER & GYROSCOPE SENSOR:** On the centre of the board, just above the pinout block, there is the accelerometer and gyroscope sensor ("2 in 1"). An accelerometer is used to measure acceleration. You can use it to detect the movements of the board (for instance, if the board is shaken). A gyroscope gives us information about the inclination of the board. This sensor works on 3 axes (X, Y, and Z), which implies you can detect movements in 3D space.

- **ATMOSPHERIC PRESSURE SENSOR**: Next to the Accelerometer/Gyroscope sensor, you can find a little sensor called the barometer. This sensor gives us the value of the atmospheric pressure.

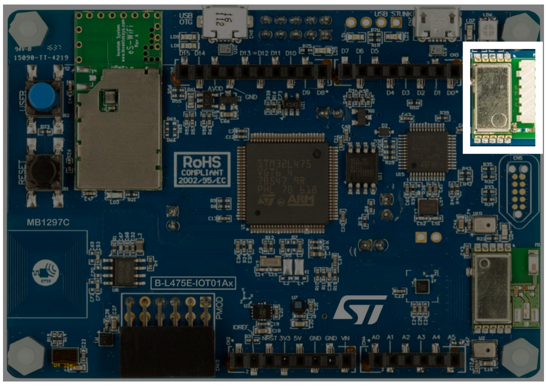
- **MAGNETOMETER SENSOR**: Next to the barometer, you can see the magnetometer. It is used to retrieve the value of a magnetic field. It can also measure values on 3 axes (X, Y, and Z).



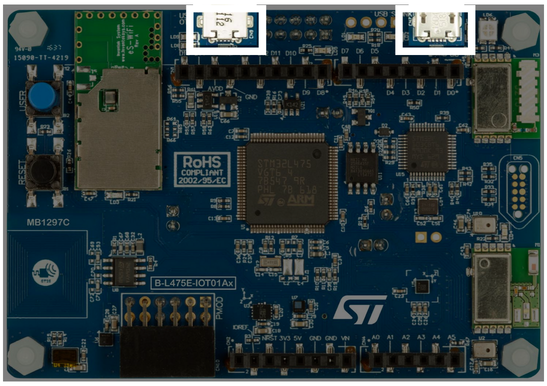
- **MICROPHONE**: On the right corner, you can see the Microphone, useful to capture sounds.

THE MODULES

- **BLUETOOTH MODULE:** On the top left of the board, you can find the Bluetooth module. It can be used to communicate and exchange data with other devices (such as another STM32 IoT Node Board, or your phone).



- **MICRO-USB CONNECTORS**: On the top of the board, you can see two micro-USB connectors. The USB port on the right is the one you will be using most of the time, as it enables to connect the board to your computer and send the program you will have done on MakeCode to the microcontroller. You can also see a second one on the left, call "OTG USB port". This particular one enables you to program the board to act and be recognised as another device such as a keyboard, mouse or gamepad.



PART II

PRACTICAL APPLICATION - ACTIVITY SHEETS & TEMPLATES

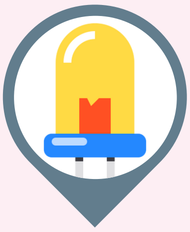
Feel free to reuse the activity sheets and templates presented in this section in your classroom and to share them with your students! You are free to print, reproduce, modify, reuse and draw inspiration from all the resources in this manual without constraint. Our content has been developed entirely under a Creative Commons license.

ACTIVITY SHEETS & TEMPLATES

PROGRAMMING EASILY THANKS TO LET'S STEAM ACTIVITY SHEETS

Authors: Jonathan Baudin, Toon Callens, Roberto Canonico, Georgios Mavromanolakis, Sébastien Nedjar, Cindy Smits

You will find here after a set of 15 activity sheets enabling you to implement concrete coding practices in your classroom lessons. These activity sheets have been developed to ease your understanding of programming and to inspire new projects.


Blink a LED


Breadboarding


Buttons & Display


Light Sensor


Potentiometer


Morse code

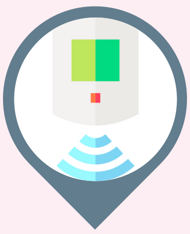

Theremin


Music


Accelerometer

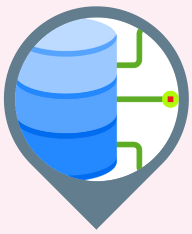

Text display


Thermometer


Motion alarm


Servos


Egg boil timer


Data collection

FULL LIST OF MATERIAL NEEDED

For all activity sheets

- 1 Programming board "STM32 IoT Node Board"
- Micro-B USB Cable

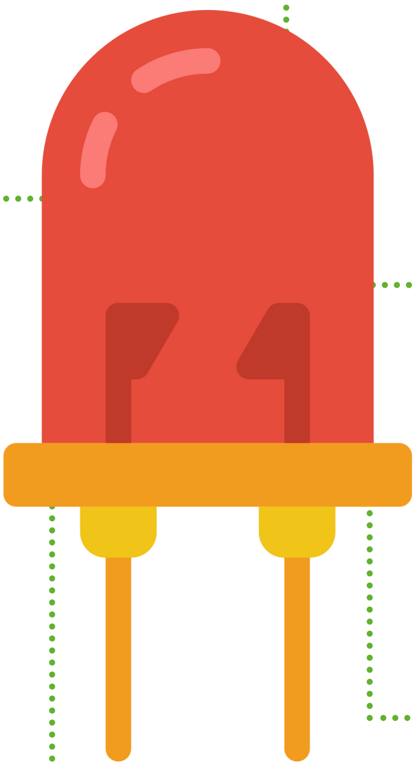
For specific activities sheets

- 1 breadboard : R1AS02, R1AS03, R1AS04, R1AS05, R1AS06, R1AS07, R1AS08, R1AS09, R1AS12
- Jumper wires: R1AS02, R1AS03, R1AS04, R1AS05, R1AS06, R1AS07, R1AS08, R1AS09, R1AS12, R1AS13, R1AS14
- 1 set of resistors: R1AS02, R1AS03, R1AS04, R1AS05, R1AS07, R1AS08, R1AS09
- 1 set of LEDs: R1AS02, R1AS03, R1AS05, R1AS08, R1AS09, R1AS13, R1AS14
- Push-buttons: *R1AS03*, *R1AS06*
- LDR (Light Dependent Resistor): *R1AS04*
- Rotary potentiometer: *R1AS05*
- 1 piezo buzzer or a speaker: *R1AS06*, *R1AS07*, *R1AS12*
- 1 OLED Display Monochrome 1.3" 128x64 OLED: *R1AS10*
- 1 QT Cable: *R1AS10*
- 1 Grove LCD I2C Text Display: *R1AS11*
- 1 Grove jumper cable: *R1AS11*
- 1 small DIY cardboard box (around 15x5 cm): *R1AS12*
- 1 SG-90 Mini Servo(1.6kg): *R1AS13*, *R1AS14*
- 1 small cardboard sheet (20cm*10cm): *R1AS14*
- 1 Sturdy Wood Sticks: *R1AS14*

PROGRAMMING RESOURCES - ACTIVITY SHEET 1

GET STARTED - BLINK A LED

#R1AS01



What is it?

A LED is an electronic component that produces light when a current flows through it. It could be used to light up a room, or to indicate something (a tank almost empty, machine turn on, ...). LEDs exist in various forms and colours.

Available on

Duration

15 minutes

Level of difficulty

Basic

Material

- 1 Programming board "***STM32 IoT Node Board***"
- 1 Micro-B USB Cable

LEARNING OBJECTIVES

- Use block for programming
- Learn the basics of MakeCode
- Use built-in LED

GET STARTED - BLINK A LED

In this getting started activity, you will approach the concept of a **pin**. A pin is a physical wire connected directly to the microcontroller. The state of a pin gives information regarding if the current flows through the pin or not. Specifically:

- LOW means there is no current
- HIGH means there is a current flow.

To make the current flow visible, we are using a component called led (light-emitting diode) already available on the board, that will light when the current flows through the pin.

STEP 1 - MAKE IT

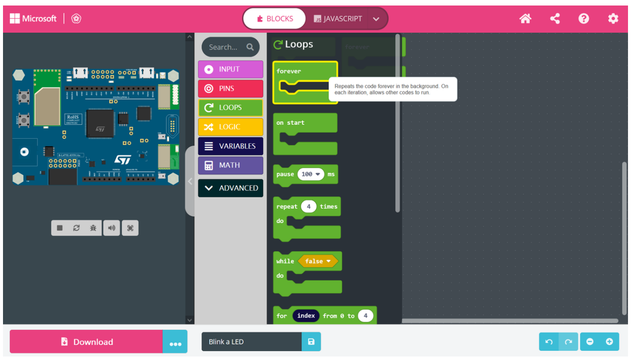
1 Connect the board to the computer

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). You should see a new drive called **DIS_L4IOT** on your computer. This drive is used to program the board just by copying a binary file.

2 Open MakeCode

Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.

Resource: makecode.lets-steam.eu



Add an infinite loop using the FOREVER block

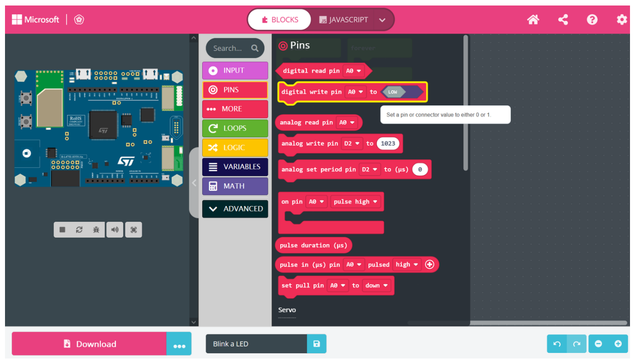
3 Arrange your blocks

From this stage, here are the different steps enabling you to make a led blink using the block editor:

Step 1 - Add an infinite loop

As we want the program to make the led blink indefinitely, the first step consists in adding the forever block. You will find it inside the LOOPS drawer. It might also be already visible in your MakeCode editor.

Step 2 - Light on the LED



Draft the digital write pin block to light on the LED

Controlling a LED is a simple task as it can only be turned on (the current flows through it), or turned off (the current does not flow). To achieve this, we need to set the state of the pin where the led is connected.

In our case, if we want to turn on the LED, we have to set the pin's state to **HIGH.** Pin's state to **LOW** will then turn it off.

On MakeCode, to control a pin's state, select the **PINS drawer**, then drag the **digital write** pin block inside the forever loop.

STEP 1 - MAKE IT

Step 3 - Create the blink

To create the blink, it is necessary for us to be able to see the led on and off for a similar amount of time. To create this blink, we need to follow the following steps:

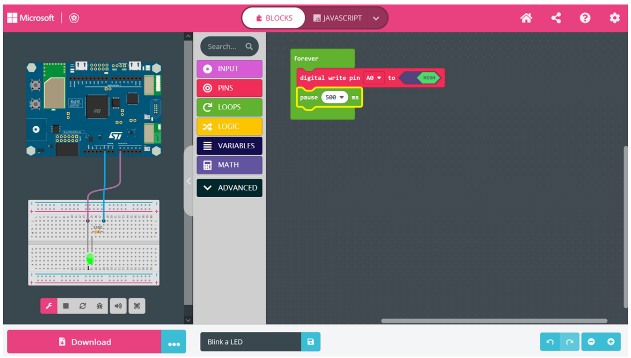
1) **Create a pause when the led is on to see the light**: Before turning the LED off, we have to wait a little amount of time, half a second (500 milliseconds) for example, with the light on. To do so, add the **pause block** (inside the LOOPS drawer), and set the value to 500 (for 500 milliseconds).

You can pick a value inside the list, or directly enter a tailored one by yourself.

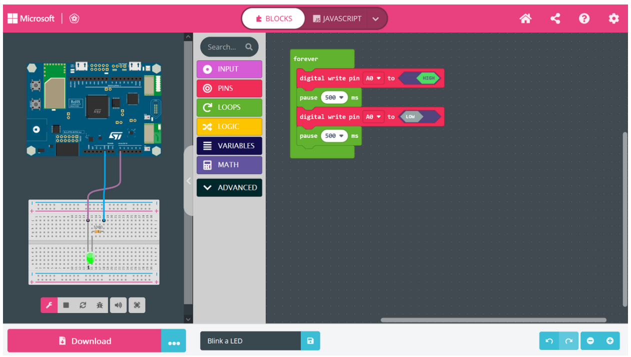
2) **Switch off the light for a similar amount of time to create the blink**: You have done half of the job until now! Add another **digital write** and **pause** block to turn off the LED and wait 500 ms again, enabling to create this effect of blink. Combined with the infinite loop, we can see this blink repeated forever.

Instead of picking blocks inside drawers, you can right-click on a block and "duplicate" it

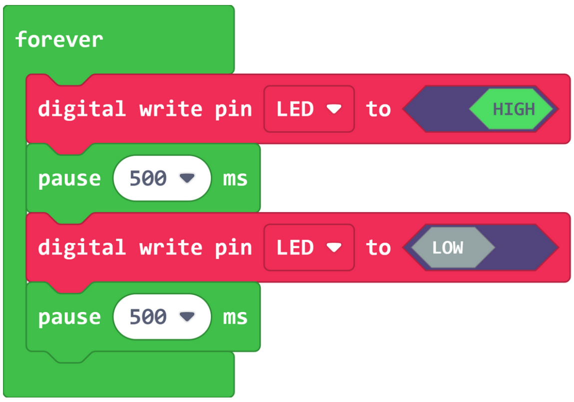
Thanks to this easy activity, you have discovered how to create a piece of code using block programming. You can get a look at the Javascript editor to see this code directly as given in the Code It section below. In the next activity sheets, feel free to directly, copy/paste the code available inside the MakeCode Javascript Editor to see the result in blocks.



Create a pause when the led is on to see the light



Switch off the light for a similar amount of time to create the blink



Full blocks enabling to program the blink a led activity

Program your board

If not already done, think of giving a name to your project and click on the "Download" button. Copy the Binary file on the drive **DIS_L4IOT** and wait until the board finishes blinking. Your first program is now running and the built-in LED should be blinking!

Run, modify, play

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). Try to understand the code and start modifying it by changing the period between two blinks. Feel free to try to blink at several rhythms or make a visual SOS in morse code.
Resource: https://en.wikipedia.org/wiki/SOS

STEP 2 - CODE IT

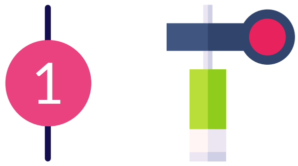
```
forever(function () {
   pins.LED.digitalWrite(true)
   pause(500)
   pins.LED.digitalWrite(false)
   pause(500)
})
```

How does it work?

Here is Javascript's translation of our block program. The keyword is a little bit different, the function digitalWrite takes a *boolean* parameter (true or false). But the translation is easy: true means HIGH & false means LOW

STEP 3 - IMPROVE IT

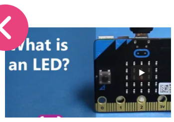
Try to make a **train light signal** by using the other built-in LED named **LED2**



GOING FURTHER

Light-emitting diode - Learn more about LED history, physical principles behind it, typologies and colours. *https://en.wikipedia.org/wiki/Light-emitting_diode*

Behind the MakeCode Hardware - LEDs on micro:bit - How do the lights work on the micro:bit? Learn all about it with **Shawn Hymel**, Technical Content Creator. *https://www.youtube.com/watch?v=qqBmvHD5bCw, https://shawnhymel.com*



Current and Voltage – Basic Electricity - Tutorial for beginners in electronics to explore current, voltage, difference and their working. *https://www.codrey.com/dc-circuits/current-and-voltage/*

Loops - Learn more about the Loops on MakeCode. *https://makecode.st.com/blocks/loops*

Explore other activity sheets

R1AS03 - Button & LED Display

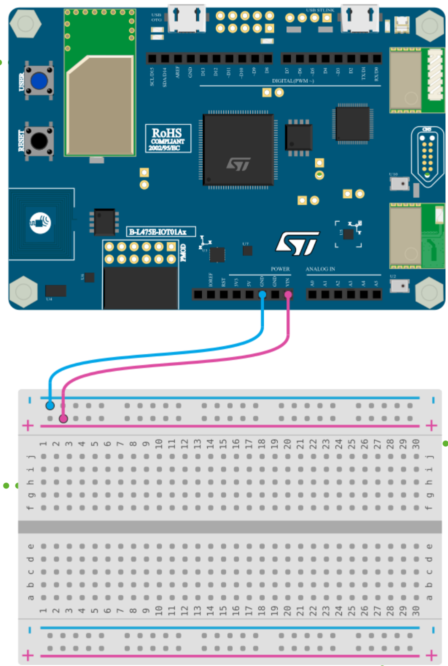
R1AS06 - Morse code

PROGRAMMING RESOURCES - ACTIVITY SHEET 2

BREADBOARDING

MAKE YOUR FIRST CIRCUIT!

#R1AS02



What is it?

A breadboard is basically a rectangular plastic board with a bunch of tiny holes in it to easily insert electronic components to prototype an electronic circuit.

Available on

Duration

15 minutes

Pre-requisites

- R1AS01 - Blink a LED

Level of difficulty

Basic

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 Breadboard
- 1 set of resistors
- 1 set of LEDs
- Jumper wires

LEARNING OBJECTIVES

- Discover breadboards
- Make a simple circuit on a breadboard
- Make a simple electronic circuit with LEDs and resistors

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

BREADBOARDING

When you first lay your hands on a **breadboard**, you will find that there are many **pinholes** and start to wonder how to create a circuit with this small plastic rectangle. Before starting, you need to understand the components of a breadboard. The **pinholes of a breadboard** are made to connect components together. When we want to create an electronic circuit, we need several connections to the same wire.

To do this, the breadboard is organized in **strips**. There are two kinds of strips:

- **Bus strips** are mainly used for power supply connections and are on the two outside columns of a breadboard.
- **Terminal strips** are mainly used for electrical components and are connected line by line. Each strip consists of 5 pinholes, indicating that you can only connect up to 5 components in one particular section.

Resources: https://en.wikipedia.org/wiki/Breadboard#Bus_and_terminal_strips

As long as an electronic component has leads (long metal legs protruding out the component) or pins (shorter metal legs), it can be used with a breadboard. To connect some strips together, we generally use **Jumper Wires**.

Resources: https://en.wikipedia.org/wiki/Jump_wire

STEP 1 - MAKE IT

Wire the power supply

1

Before connecting the components, we generally add some wires to the bus strips to distribute the power supply (**+5V** and pin **GND**). Take two wires and make the following connections.

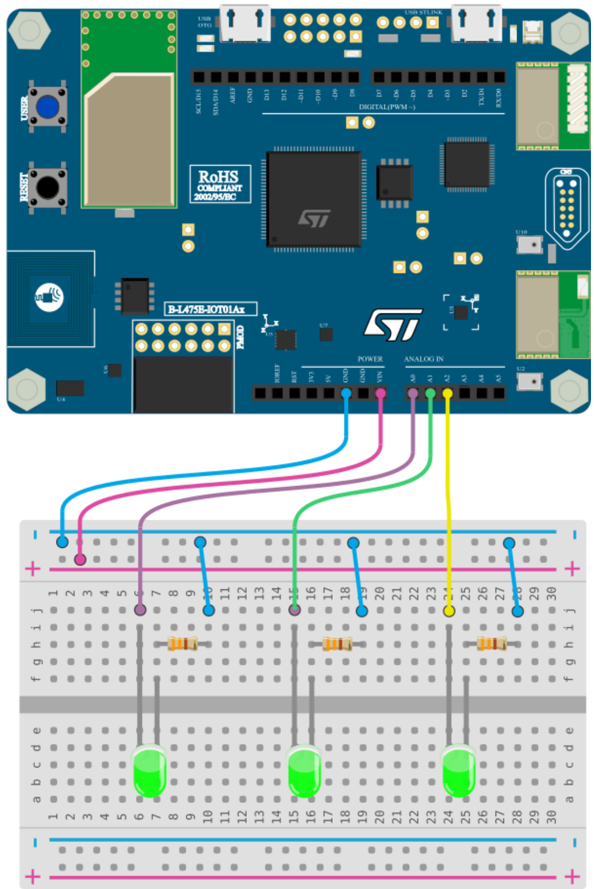
Wire the first LED

2

Our circuit is just a simple LED connected to one pin of the board. Connect the anode of the LED on the pin labelled **A0** (for Analog 0). Then connect the cathode to a resistor (330 ohms) and plug the unconnected resistor's lead into the pin labelled **GND**.

LED has an orientation. To designate the correct orientation, each leg has a name. This is how to find the difference between anode and cathode:

- **Anode: This is the '+' of the LED. The anode leg is longer than the cathode lead.**
- **Cathode: This is the '-' of the LED. The cathode leg is shorter than the anode lead.**



Wiring LEDs

STEP 1 - MAKE IT

Wire other LEDs

3

We will duplicate the previous circuit with two additional LEDs. The anode of these new LEDs will be connected in pin **A1** and pin **A2**.

Connect the board to the computer

4

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

Open MakeCode

5

Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.

Resource: makecode.lets-steam.eu

Program your board

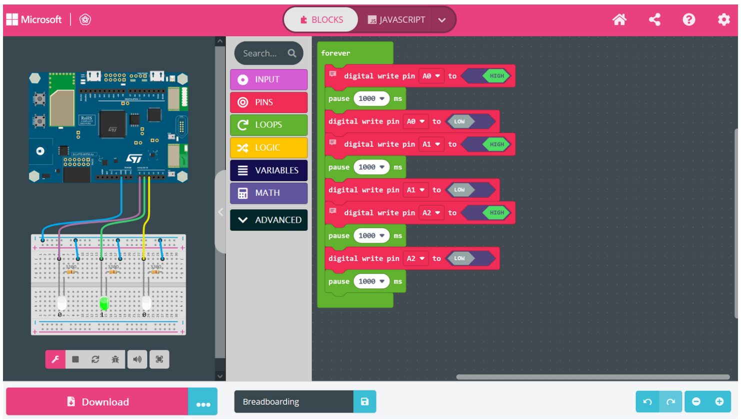
6

Inside the MakeCode Javascript Editor, Copy/Paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "**Download**" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finishes blinking and your first program is ready!

Run, modify, play

7

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). Use the knowledge acquired on this activity sheet to make more or less complex projects and explore the next activity sheets.



MakeCode editor in blocks

Full blocks enabling the program to run

STEP 2 - CODE IT

```
forever(function () {
    // Blink the first LED
    pins.A0.digitalWrite(true)
    pause(1000)
    pins.A0.digitalWrite(false)

    // Blink the second LED
    pins.A1.digitalWrite(true)
    pause(1000)
    pins.A1.digitalWrite(false)

    // Blink the third LED
    pins.A2.digitalWrite(true)
    pause(1000)
    pins.A2.digitalWrite(false)
    pause(1000)
})
```

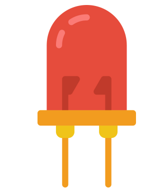
How does it work?

This program is an extended version of the "Blink a led" program adapted with three LEDs. For each LED :

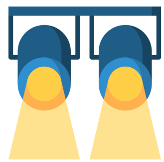
- the digitalWrite block lights off or lights on a specific LED
- the pause block waits a small amount of time.

STEP 3 - IMPROVE IT

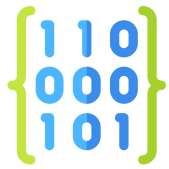
By **changing the on/off lighting order**, make a simple animation where the LEDs are turned on and off, one after the other.



Plug LEDs of different colours - **red, green and yellow** - and try to simulate a stoplight.

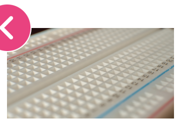


You can use LEDs to approach binary counting! When we count in binary, we represent numbers with arrangments of 1's and 0's. Discover more information on binary counting on the **CS Unplugged resource centre**. Once the basics of binary counting are acquired, **transform this program to show numbers from 0 to 7 in binary with the three LEDs.**
Resource: https://csunplugged.org/en/topics/binary-numbers/unit-plan/

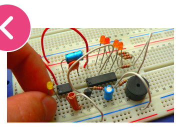


GOING FURTHER

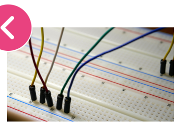
How to Use a Breadboard - Tutorial video giving a basic introduction to breadboards and explaining how to use them in beginner electronics projects. *https://www.sciencebuddies.org/science-fair-projects/references/how-to-use-a-breadboard*



Use a Real Bread-Board for Prototyping Your Circuit - Step by step prototyping with a breadboard. *https://www.instructables.com/Use-a-real-Bread-Board-for-prototyping-your-circui/*



Basic LED Animations for Beginners (Arduino) - Tutorial to revisit some concepts about using LEDs and make some fun effects using the RedBoard Qwiic to control the individual LEDs. *https://learn.sparkfun.com/tutorials/basic-led-animations-for-beginners-arduino/all*



Electronics Basics 10 - An insight into how breadboards work. *https://www.youtube.com/watch?v=fq6U5Y14oM4*

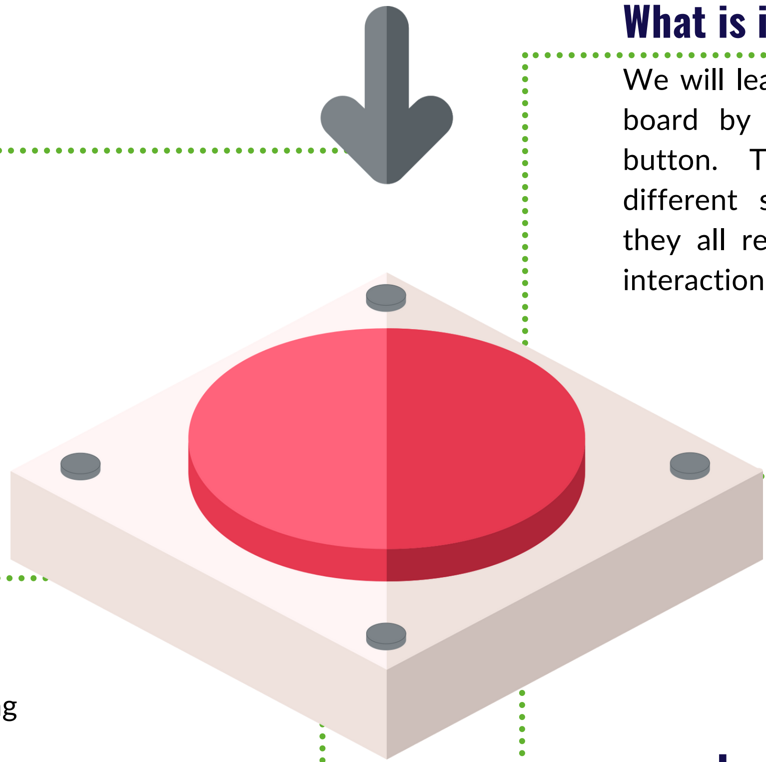
Explore other activity sheets

R1AS03 - Buttons and LED Display

PROGRAMMING RESOURCES - ACTIVITY SHEET 3

BUTTONS AND LED DISPLAY

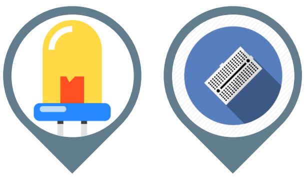
#R1AS03



What is it?

We will learn to interact with the board by using a simple push-button. They come in many different shapes and sizes, but they all require the most simple interaction: pushing it!

Available on



Duration

25 minutes

Pre-requisites

- R1AS01 - Blink a LED
- R1AS02 - Breadboarding

Level of difficulty

Intermediate

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 2 Push-buttons
- 1 set of LEDs
- 1 set of resistors
- 1 breadboard
- Jumper wires

LEARNING OBJECTIVES

- Add interactivity
- React to an event on a physical button
- Use a variable to store the current state of the program
- Wire simple circuit on a breadboard with buttons and LEDs
- Use the MakeCode simulator

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

BUTTONS AND LED DISPLAY

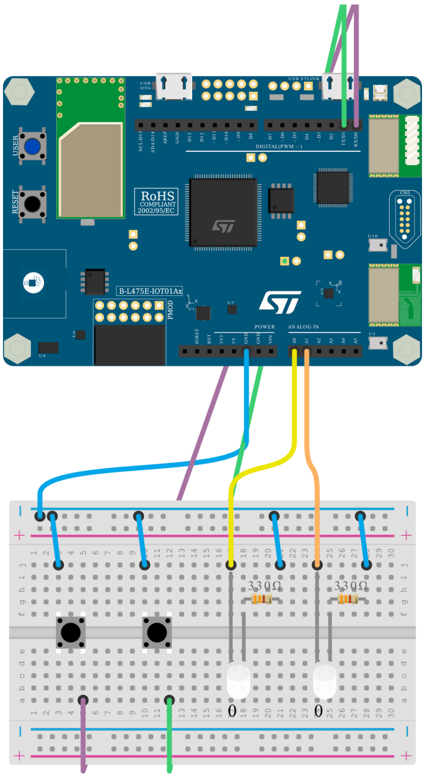
To learn how to use a button, **let's play a quiz game**!
The idea is pretty simple: **2 players, one button, and one LED for each**. When the animator asks a question, the player has to push her/his button first to give the correct answer. LEDs indicate which player pushes the button first and can talk.

STEP 1 - MAKE IT

1

Wire buttons and LEDs

Connect one side of each button to the **pin GND** on the board. Then connect the other side on **pin D0** for player 1, and on **pin D1** for player 2. Connect the anode of player 1's LED on **pin A0** and the one of player 2 on **pin A1**. Connect each LED's **cathode** to a resistor (330 ohms). Then plug the unconnected resistors' legs into **pin GND.**



Wiring buttons and LEDs

LED has an orientation. To designate the correct orientation, each leg has a name. This is how to find the difference between anode and cathode:

- **Anode: This is the '+' of the LED. The anode leg is longer than the cathode lead.**
- **Cathode: This is the '-' of the LED. The cathode leg is shorter than the anode lead.**

2

Connect the board to the computer

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

3

Open MakeCode

Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.
Resource: makecode.lets-steam.eu

4

Program your board

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "**Download**". Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finishes blinking, your quiz buzzer is ready!

5

Run, modify, play

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). Try to understand the example and start modifying it by changing the period between two game sessions. Feel free to use this activity sheet to make an evaluation quiz inside the classroom more attractive and engaging for your learners!

STEP 2 - CODE IT

```
//Initialization
let weCanPushIt = true
pins.A0.digitalWrite(false)
pins.A1.digitalWrite(false)
```

Initialization

As a first step, we need to declare a variable named weCanPushIt, of boolean type - a form of data with only two possible values, usually "true" and "false". This variable will be useful to know if we can push the button, or if the other player is already doing it. The 2 last lines give the information that all the LEDs are turned off.

A variable is a way of naming and storing a value for later use by the program, such as data from a sensor or an intermediate value used in a calculation. Variable has a name and a type. The type allows specifying which kind of data the variable can hold.

```
input.buttonD0.onEvent(ButtonEvent.Down, function () {
    if (weCanPushIt) {
      weCanPushIt = false
      pins.A0.digitalWrite(true)
      pause(3000)
      pins.A0.digitalWrite(false)
      weCanPushIt = true
    }
})

input.buttonD1.onEvent(ButtonEvent.Down, function () {
    if (weCanPushIt) {
      weCanPushIt = false
      pins.A1.digitalWrite(true)
      pause(3000)
      pins.A1.digitalWrite(false)
      weCanPushIt = true
    }
})
```

Interactions

The main code is about the buttons' interactions made with the **input.buttonXX.onEvent** functions.

A Function is a block of code that executes a specific task. It is really useful to simplify the code and make a block of code more expressive.

The most important line here is the condition if (weCanPushIt) { ... } which tests if players have already or not pushed their button yet. If this is the case (weCanPushIt is equal to true), we:

1. Set weCanPushIt to false, to disallow the opponent to push his button.
2. Turn on the LED of the player to show who is the winner
3. Wait 3 seconds (3,000 milliseconds)
4. Turn off the winner's LED
5. Set weCanPushIt to true, to allow players to push their buttons.

BUTTONS AND LED DISPLAY

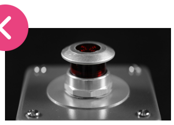
STEP 3 - IMPROVE IT

1. **Add other buttons and LEDs** and modify your program for playing with more players!

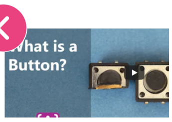
2. Modify your program to **make the winner's LED blinking** using the blink a led activity sheet.

GOING FURTHER

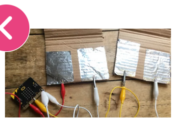
Push-button - Learn more about push-button uses. *https://en.wikipedia.org/wiki/Push-button*



Behind the MakeCode Hardware - Buttons on micro:bit - All about buttons and their use in MakeCode with Shawn Hymel, Technical Content Creator. *https://www.youtube.com/watch?v=t_Qujjd_38o*, *https://shawnhymel.com*



Reaction game - Make a reaction game with real physical switches you can bash as hard as you like! *https://microbit.org/projects/make-it-code-it/reaction-game/*



Discover what is a **Variable** - Learn more about the variables and **What Is a Function in Programming?** - Learn more about function. *https://www.computerhope.com/jargon/v/variable.htm*, *https://www.makeuseof.com/what-is-a-function-programming/*

Explore other activity sheets

R1AS04 - Basic Light Sensor

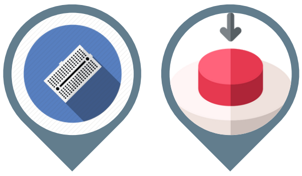
BASIC LIGHT SENSOR

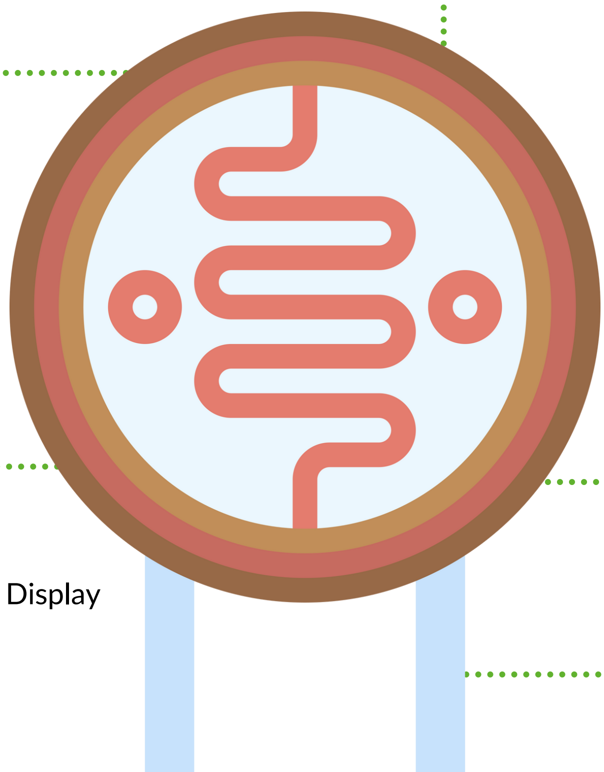
#R1AS04

What is it?

This activity sheet will approach resistors. A light-dependent resistor (LDR) is a component used to measure light levels.

Available on





Duration

25 minutes

Pre-requisites

- R1AS02 - Breadboarding
- R1AS03 - Buttons and LED Display

Level of difficulty

Intermediate

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 set of resistors
- 1 LDR
- 1 Breadboard
- Jumper wires

LEARNING OBJECTIVES

- Create a simple light sensor with a few electronic components on a breadboard and connect it to the board
- Create a program in MakeCode that is able to measure an analogue physical quantity by means of a sensor
- Produce a plot that shows how a measured value varies over time

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

BASIC LIGHT SENSOR

This activity illustrates a key feature of physical computing: the ability to measure a physical quantity using a sensor and graphically represent how this quantity varies over time. We will connect a light-dependent resistor (LDR) to the board to measure light levels. This kind of sensor is called an **analogue sensor** because we need to get an analogue characteristic of the circuit (the voltage) to get the value of the sensor.
Resource: https://www.watelectrical.com/what-are-analog-sensors-types-and-their-characteristics/

STEP 1 - MAKE IT

1

Wire the photocell

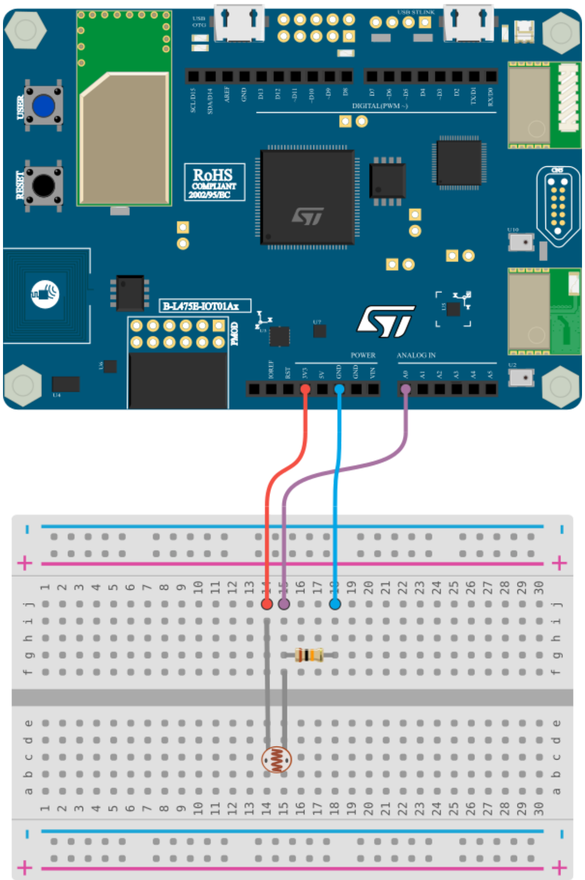
The circuit we need to assemble consists of two components: a **4.7 kΩ resistor** and a **photocell**.

The colour of the first three stripes indicates the resistance value of the component, according to a code that is known as "resistors colour code". The fourth stripe indicates that the resistance value is subject to uncertainty (tolerance) that may be either 5% (if the stripe is gold) or 10% (if the stripe is silver) of the nominal resistance value.

Light Dependent Resistors (aka. LDR, Photocell, Photoresistor and CdS Cell) is a component whose electrical resistance varies according to the intensity of the light to which the component is exposed.

The easiest way to measure a resistive sensor is to connect one end to Power and the other to a pull-down resistor to the ground. Then, the point between the fixed pulldown resistor and the variable photocell resistor is connected to the analogue input of a microcontroller. Such an arrangement forms what we call an analogue sensor. This term means that **this circuit is able to sense a physical quantity** (namely, light intensity) and transform it into a **proportional electrical quantity** (specifically, a voltage whose value is between 0 V and 3.3 V).

These two components need to be assembled on a small breadboard, as depicted in the picture aside.



Assembly of the 4.7 kΩ resistor and the photocell on the breadboard

2

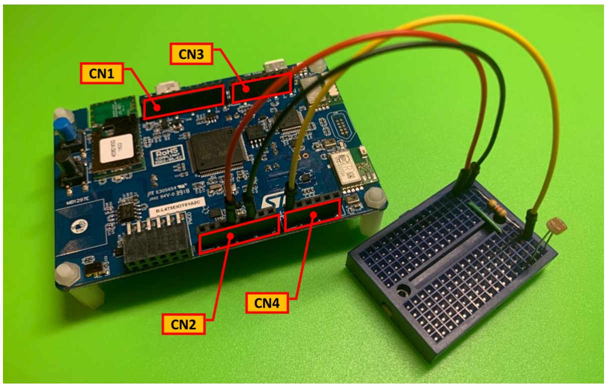
Wire the breadboard to the STM board

Once the breadboard has been assembled, it needs to be connected to the board. The picture shows that the board has four connectors, named **CN1**, **CN2**, **CN3** and **CN4**, respectively. Since the four connectors have different purposes, use the blue buttons located at one of the four corners of the board to properly identify the four connectors.

BASIC LIGHT SENSOR

STEP 1 - MAKE IT

The red wire must be connected to **pin 4** of connector **CN2**, which is internally connected to a 3.3 V potential. The black wire must be connected to **pin 6** of the connector **CN2**, which is internally connected to the ground potential (**GND**). Finally, the yellow wire must be connected to **pin 1** of connector **CN4**. This pin is internally connected to the analogue input pin named **A0**.



Wiring the breadboard to the STM board

3

Connect the board to the computer

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

4

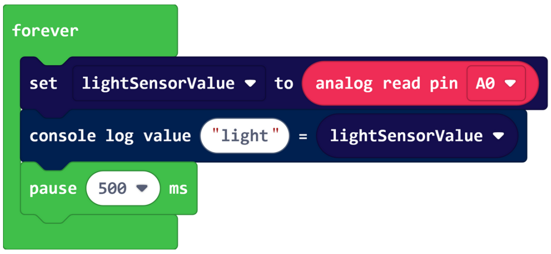
Open MakeCode

Go to the **Let's STEAM MakeCode editor.** On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.
Resource: makecode.lets-steam.eu

5

Program your board

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "**Download**" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finishes blinking and your program is ready!



Full blocks enabling the program to run

6

Connect to the board console

In your MakeCode editor, click on the button "Show console Simulator" below, on the left side, the board simulation. The terminal shows then the periodic light values read by the program. This value can be exported as a CSV file by clicking on the button "export data" in the top right corner of the console.


Console on the MakeCode editor

7

Run, modify, play

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). Try to understand the example and start modifying it by changing the period between two measurement sessions. You can hide the photocell with your hand to directly observe the value changing.

STEP 2 - CODE IT

```
let lightSensorValue = 0
forever(function () {
   lightSensorValue = pins.A0.analogRead()
   console.logValue("light", lightSensorValue)
   pause(500)
})
```

How does it work?

The code consists of:

- a forever block;
- a console log block;
- a pause block.

The forever block implements "a loop", which keeps executing three basic instructions until the board is turned off.

The first block reads the value of the analogue input pin **A0** and stores it in a variable named lightSensorValue. This value is an integer number between 0 and 1023.

An analogue input pin may be used to read a value between 0 and 1023. This value is proportional to the voltage applied to the pin, which MUST be comprised between 0 V and 3.3 V (relative to GND).

The second block writes to the console terminal of the board what is obtained by reading the value of the sensor.

As soon as this instruction has been carried out, the board suspends its activity (pause) for 500 milliseconds, i.e. half a second.

Now a question naturally arises: what is the board console? How is it possible for us to read what is written to the console? The board console allows the board to simply interact with the PC connected to it through the USB cable.

STEP 3 - IMPROVE IT

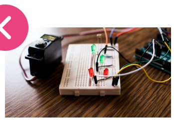
1. Use your sensor in **many light conditions** (ambient light, moonlit night, ….). How can we calibrate our sensor to be well adapted to the sensing condition? **Try several values of the pull-down resistor to see the impact.**

2. Add an LED and transform this circuit into a hand **controllable light dimmer.**

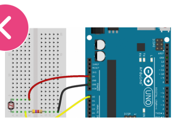
3. The actual value of the sensor is a value between 0 and 1023. **Read the value of the darkest light and the value of the brightest light** and transform the original value into a more explicit percentage value.

GOING FURTHER

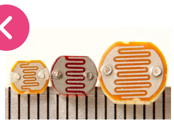
Light-dependent resistor - Learn more about photoresistors, their applications and design. *https://en.wikipedia.org/wiki/Photoresistor*



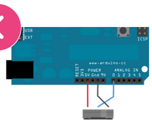
Photocell Hookup Guide - Quick primer on resistive photocells', and demonstrates how to hook them up and use them. *https://learn.sparkfun.com/tutorials/photocell-hookup-guide/all*



Photocells - Discover photocells, a resistor that changes its resistive value depending on how much light is shining onto the squiggly face. *https://learn.adafruit.com/photocells*



Analog Read Pin - Choose a pin and read an analogue signal (0 through 1023) from it. *https://makecode.microbit.org/reference/pins/analog-read-pin*



Explore other activity sheets

R1AS11 - Make a very readable thermometer

R1AS15 - Collecting data

POTENTIOMETER

#R1AS05

What is it?

In this activity sheet, we will learn about potentiometer by programming the board to adjust the brightness of a LED by turning a knob.

Available on

Pre-requisites

- R1AS01 - Blink a LED
- R1AS02 - Breadboarding
- R1AS04 - Basic Light Sensor

Duration

20 minutes

Level of difficulty

Intermediate

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 Breadboard
- 1 Potentiometer
- 1 set of LEDs
- 1 set of resistors
- Jumper wires

LEARNING OBJECTIVES

- Wire external components to the board
- Read an analog input using a potentiometer
- Use an analog input to write an analog output

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

POTENTIOMETER

A potentiometer is a **three-terminal resistor** with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat. The measuring instrument called a potentiometer is essentially a **voltage divider** used for measuring electric potential (voltage); the component is an implementation of the same principle, hence its name.

Potentiometers are commonly used to **control electrical devices** such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position transducers, for example, in a joystick. Potentiometers are rarely used to directly control significant power (more than a watt) since the power dissipated in the potentiometer would be comparable to the power in the controlled load.

Resource: https://en.wikipedia.org/wiki/Potentiometer

STEP 1 - MAKE IT

1 Wire the potentiometer

Connect the left prong of the potentiometer to **GND**. The right prong should be connected to **3.3V**. Wire the middle one to **A0**.

2 Wire the LED

Connect the **anode** (**+**) of the LED on **D9**. Connect the LED's **cathode** (**-**) to a resistor (330 ohms). Then, connect the unconnected side of the resistor to **GND**.

3 Connect the board to the computer

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

4 Open MakeCode

Go to the **Let's STEAM MakeCode editor.** On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.

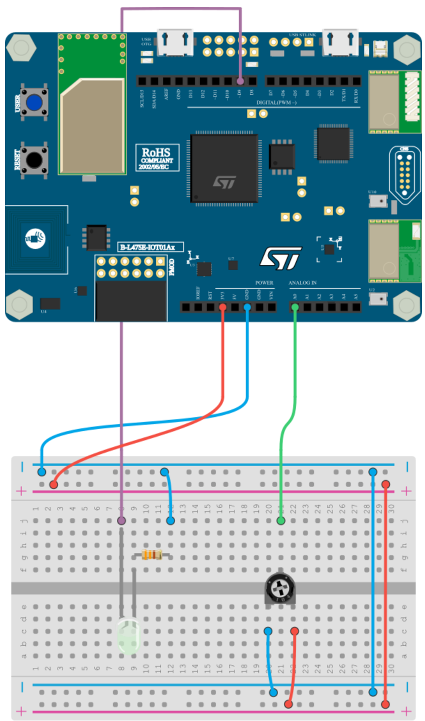
Resource: makecode.lets-steam.eu

5 Program your board

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "**Download**" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finishes blinking and your board is ready!

6 Run, modify, play

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). Try to understand the example and start modifying it.



Wiring the potentiometer and LED

STEP 2 - CODE IT

```
forever(function () {
    pins.D9.analogWrite(pins.A0.analogRead())
})
```

How does it work?

The code consists of three elements:

- a forever block;
- an analogRead block;
- an analogWrite block.

The forever block implements “a loop” which keeps executing the instructions until the board is turned off.

The analogRead block is used to get the value of the potentiometer on pin A0. This value is an integer number between 0 and 1023. Turning the knob changes the value.

The potentiometer acts as an adjustable voltage divider. By changing the position of the knob, you change the voltage applied on A0. The more you turn it to the left, the more the voltage will be close to 0V. The more you turn it to the right, the more the voltage will be close to 3.3V.

An analogue input pin may be used to read a value between 0 and 1023. This value is proportional to the voltage applied to the pin, which must be comprised between 0V and 3.3V.

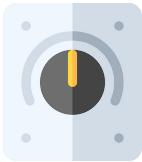
The analogWrite block is used to light up the LED on D9. By using analogWrite, the board is able to limit the voltage to a certain value to make the LED shine dimmer or brighter. The brightness is set by the value of analogRead on pin A0: the higher the value, the brighter the LED.

By using pin D9, we are able to write an analogue value through a digital pin to the board. Pin D9, like a few other pins on the board, supports Pulse Width Modulation or PWM. This technique uses on-off patterns to simulate different voltages and thus different analogue signals. The value passed to analogWrite should be between 0 and 255. 0 stand for a 0V voltage and 255 for 3.3V.

As you will see by using this program, you will not use the full range of the potentiometer. You can transform the range of value of the potentiometer(0...1023) into the range of the PWM (0...255) with the map function.

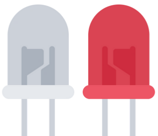
STEP 3 - IMPROVE IT

1



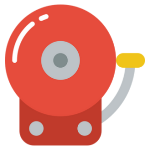
By using the map function, try to use **the full range of the potentiometer**. You can define two variables to be more expressive and separate reading, transforming and writing on a specific statement.

2



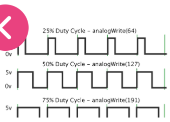
Add another LED and inverse the value of the potentiometer, so that the second LED dims as the first one lights up.

3

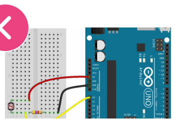


Use the potentiometer to **control the pitch of a buzzer**. Use a potentiometer to **control the position of a servo.**

GOING FURTHER



Pulse Width Modulation - Arduino tutorial on the use of analogue output (PWM) to fade an LED. *https://www.arduino.cc/en/Tutorial/Foundations/PWM*



Voltage Dividers - Discover how voltage dividers behave in the real world. *https://learn.sparkfun.com/tutorials/voltage-dividers*

Arduino pong game on 24x16 matrix with MAX7219 - Build a small pong console. *https://www.youtube.com/watch?v=dK9F5AJM2XI*

Potentiometer Game - Control a game's avatar using a potentiometer. *https://www.hackster.io/matejadjukic03/potentiometer-game-05ee93?f=1#*

Explore other activity sheets

R1AS11 - Make a very readable thermometer

R1AS15 - Collecting data

PROGRAMMING RESOURCES - ACTIVITY SHEET 6

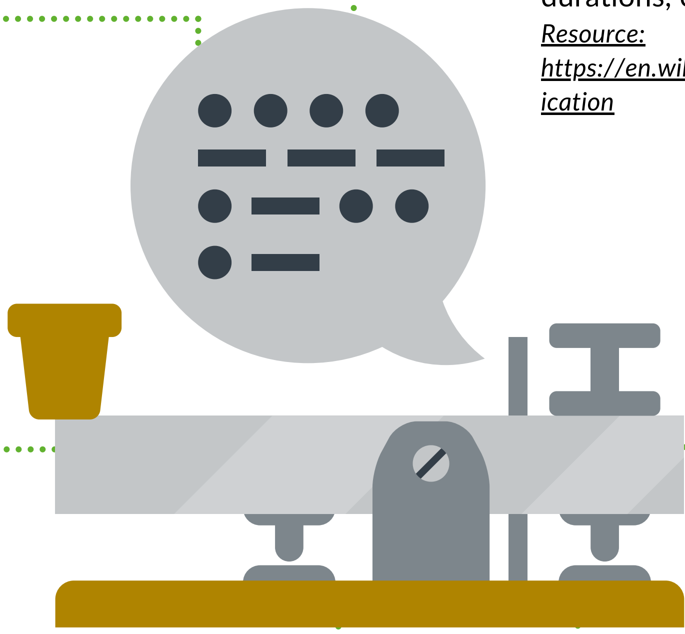
MORSE CODE

#R1AS06

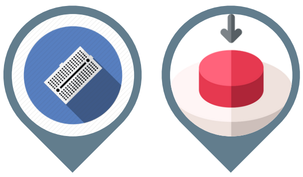
What is it?

Morse code is a method used in telecommunication to encode text characters as standardized sequences of two different signal durations, called dots and dashes.

Resource:
https://en.wikipedia.org/wiki/Telecommunication



Available on



Pre-requisites

- R1AS02 - Breadboarding
- R1AS03 - Buttons and LED Display

Duration

30 minutes

Level of difficulty

Advanced

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 Breadboard
- 1 piezo buzzer or a speaker
- 2 buttons
- Jumper wires

LEARNING OBJECTIVES

- Wire and use the passive buzzer
- Communicate with Morse code

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

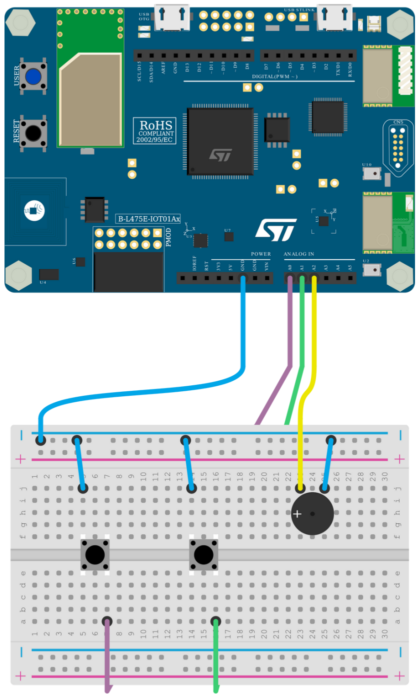
MORSE CODE

From microwaves to game shows, buzzers are all around us and can help to signal something with a beeping noise. To emit sound (or noise), the buzzer contains a thin membrane (made of quartz), which vibrates to a given frequency (between 20Hz and 20,000Hz, which are the listenable frequencies).
Resource: https://en.wikipedia.org/wiki/Buzzer

In this activity sheet, you will attach some buttons and a buzzer to the board and learn to communicate with **morse code**!
Resource: https://en.wikipedia.org/wiki/Morse_code

STEP 1 - MAKE IT

1 Wire buzzer
In theory, a buzzer is not polarized (it means that there is no "+" nor "-"), but you often have a pair of wires black/red or signs ("+" and/or "-") on the device. If you are in this configuration, attach the lead on the "+" side of the buzzer to pin **A2** and the other one to pin **GND**. If there is no colour or indication, just plug one wire on pin **A2** and the other one on pin **GND.**



Wiring the buzzer and buttons

2 Wire buttons
Connect one side of each button to the pin **GND** on the board. Then attach the other sides, on pin **A0** (button 1), and pin **A1** (button 2).

3 Connect the board to the computer
With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well, you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

4 Open MakeCode
Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.
Resource: makecode.lets-steam.eu

5 Program your board
Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "**Download**" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finishes blinking and your work is ready!

6 Run, modify, play
Your program will automatically run each time you save it or reset your board (push the button labelled RESET). Try to understand the example and start modifying it.

STEP 2 - CODE IT

```
// Send short signal
input.buttonA0.onEvent(ButtonEvent.Click, function () {
   music.playTone(440, 100)
})

// Send long signal
input.buttonA1.onEvent(ButtonEvent.Click, function () {
   music.playTone(440, 300)
   })
```

How does it work?

The code is really simple! You can see the two functions onEvent to detect when a button is pressed.

Then we simply used the music.playTone function, with 2 parameters:

- **440**: the frequency we want to play
- **100** or **300**: the duration of the tone in milliseconds (1 second = 1,000 milliseconds)

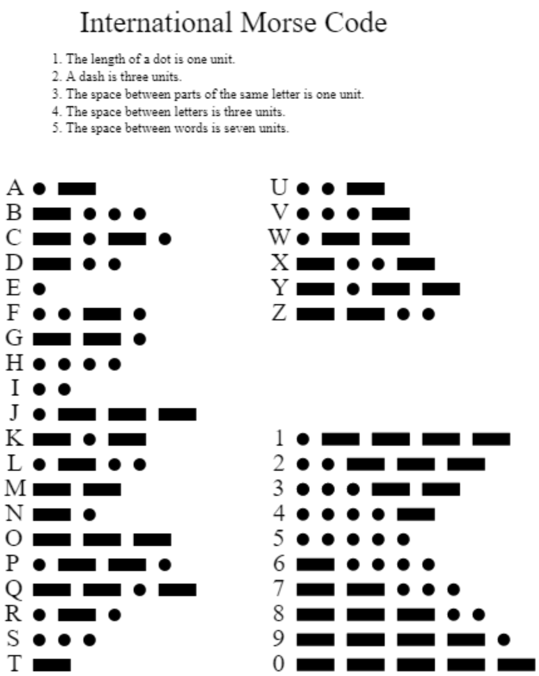
Now you have understood the basics, we will send a morse message!

Signalling morse code

Morse code is a method of communication that encodes characters as a sequence of **2 different signal durations** known as **dots** and **dashes**.

A **dot** is a **short signal** while a **dash** is a **longer signal**. By combining multiple sequences, you can convey a message consisting of several words. Morse code can be signalled in various ways: by using a (flash)light, a radio, or with a board like the one you have!

The figure on the right gives you an overview of how to signal each letter in morse. Try to send "**SOS**" to someone!



STEP 3 - IMPROVE IT

1. To help hearing-impaired people, **add a LED to indicate when the buzzer is going on.**

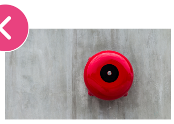
2. You can try to make your **preferred music by playing several tones** when you click on the button.

3. Add more buttons and try to play a **simple melody**.

GOING FURTHER

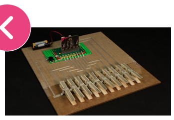
Morse Code - Learn more about morse code history, representations, timing, speeds and learning methods. *https://en.wikipedia.org/wiki/Morse_code*

Buzzer Basics - Technologies, Tones, and Drive Circuits. *https://www.cuidevices.com/blog/buzzer-basics-technologies-tones-and-driving-circuits*



Sound - Discover acoustics basis, physics and perception of sounds. *https://en.wikipedia.org/wiki/Sound*

Clothespin Piano with micro:bit - Read an analogue signal (0 through 1023) from the pin. *https://browndoggadgets.dozuki.com/Guide/Clothespin+Piano+with+micro:bit/302*



Explore other activity sheets

R1AS07 - Make a theremin with the distance sensor

PROGRAMMING RESOURCES - ACTIVITY SHEET 7

MAKE A THEREMIN

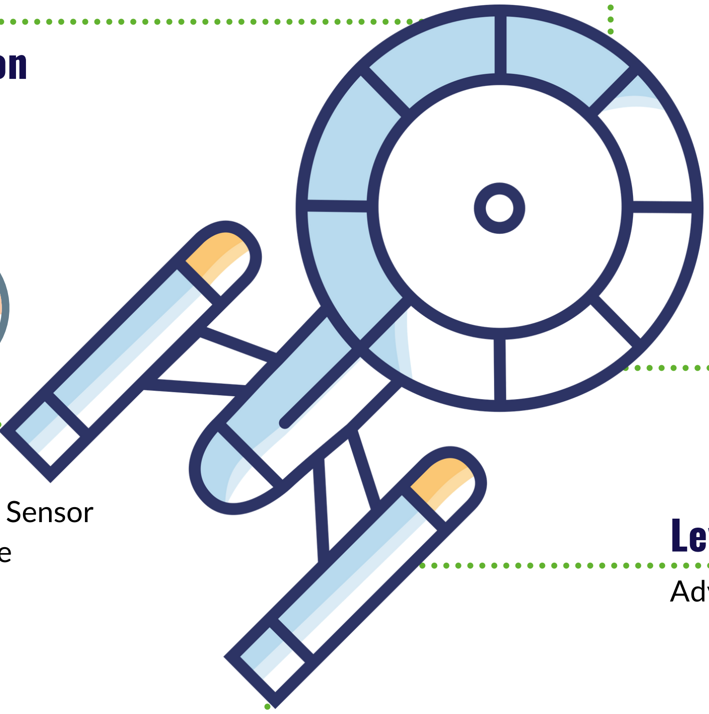
WITH THE DISTANCE SENSOR

#R1AS07

What is it?

A theremin is an electronic musical instrument, that can be played without touching it. The original concept is based on using two antennas to detect the hands' position. One antenna is used for the volume, and the other for the pitch of the tone.

Available on



Duration

30 minutes

Pre-requisites

- R1AS04 - Basic Light Sensor
- R1AS06 - Morse code

Level of difficulty

Advanced

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 piezo buzzer or a speaker
- 1 Breadboard
- Jumper wires

LEARNING OBJECTIVES

- Use a distance sensor and understand how it works
- Make music with a really strange instrument
- Use the map function to transform a number from one range to another

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

MAKE A THEREMIN WITH THE DISTANCE SENSOR

The theremin is an electronic musical instrument controlled without physical contact by the thereminist (performer). It is named after its inventor, Leon Theremin, who patented the device in 1928. The instrument's controlling section usually consists of two metal antennas that sense the relative position of the thereminist's hands and control oscillators for frequency with one hand and amplitude (volume) with the other. The electric signals from the theremin are amplified and sent to a loudspeaker.
Our version will be more simple, we will only control the pitch of the tone, with the distance sensor, the volume will be predetermined. **Let's make music!**
Resources: https://en.wikipedia.org/wiki/Theremin, https://youtu.be/x0NVb25p1oU

STEP 1 - MAKE IT

1 Wire buzzer/speaker

In theory, a buzzer or a speaker is not polarized (it means that there is no "+" nor "-"), but you often have a pair of wires black/red or signs ("+" and/or "-") on the device. If you are in this configuration, attach the lead on the "+" side of the buzzer to **A0** and the other one to **GND**.

If there is no colour or indication, just plug one wire on **A0** and the other one on **GND**.

2 Connect the board to the computer

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

3 Open MakeCode

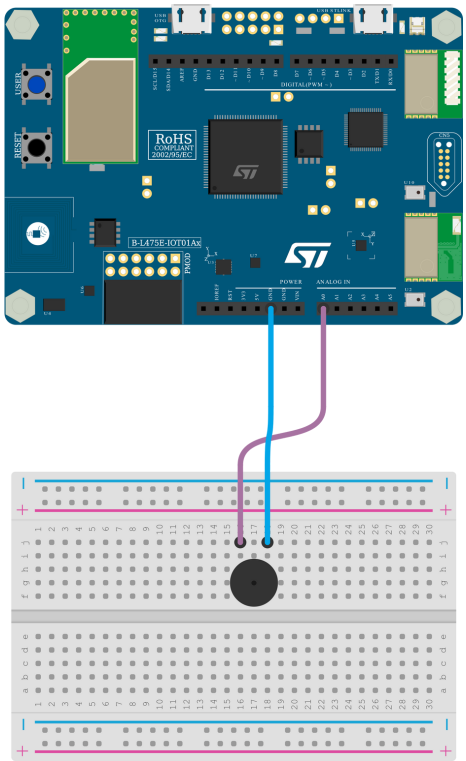
Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.
Resource: makecode.lets-steam.eu

4 Program your board

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "**Download**" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finishes blinking and your program is ready!

5 Run, modify, play

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). Try to understand the example and start modifying it.



Wiring the buzzer/speaker

MAKE A THEREMIN WITH THE DISTANCE SENSOR

STEP 2 - CODE IT

```
let distance = 0
forever(function () {
   // Get distance
   distance = input.distance(DistanceUnit.Millimeter)

   if (distance > 500) {
      // Convert the distance into frequency
      let note = Math.map(distance, 0, 500, 440, 830)
      music.ringTone(note)
   } else {
      music.stopAllSounds()
   }
})
```

Variables

In this program, there are 2 variables. The first, distance, is used to keep the same distance across the condition and for the tone to play. Then, you will find note, which is not technically necessary/mandatory but helps to introduce a greater understanding of each step of the program. It contains the transformation of the distance into tone frequency.

Getting distance

Using a variable to keep the distance is great, but knowing how to get the distance is better! Once again, there is no difficulty. We need to call the input.distance(DistanceUnit.Millimeter) function. The parameter DistanceUnit.Millimeter specifies to the function that we want the result in millimetres (1 meter = 1,000 millimetres).

Condition

The condition, if (distance > 500) { ... }, gives the information that we only play a sound if the measured distance is lower or equal to 500 millimetres.

Convert the distance into frequency

The most important part is the **conversion**. To make it, we use a mathematical function named map. This function remaps a value from a range to another. In this case, the value is remapped from **distance range** to **frequency range**. As you can see in the code above, this function takes five parameters, namely: map(value, in_min, in_max, out_min, out_max). Let's get a closer look at each of them:

- **value**: the value to re-map
- **in_min**: The minimum value of input range (distance)
- **in_max** : the maximum value of input range (distance)
- **out_min** : the minimum value of output range (frequency)
- **out_max**: the maximum value of output range (frequency)

So, we can understand, what this line does i.e., remapping the distance (with a range of 0 mm to 500 mm) to frequency (with a range from 440 Hz to 830 Hz).

The chosen frequencies are not random, the range of frequency from 440Hz to 830Hz represents an octave. It means you can find all the notes: LA, SI, DO, RE, MI, FA, SOL

Now we have a frequency. It's time to play it, simply using the music.ringTone(note).

MAKE A THEREMIN WITH THE DISTANCE SENSOR

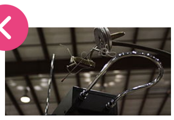
STEP 3 - IMPROVE IT

1. Change the map value to add octaves and/or distance to enhance your song.

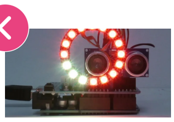
2. Try to add a potentiometer to control the volume.

GOING FURTHER

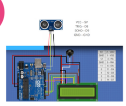
Theremin - Learn more about the history, operating principles and uses of the theremin. *https://en.wikipedia.org/wiki/Theremin*



LED Ring Distance Sensor - Discover a fun project, which will end up in an alternative parking sensor. *https://www.instructables.com/LED-Ring-Distance-Sensor/*



Water Level Detector - Discover ultrasonic sensors converting electrical energy into acoustic waves. *https://www.instructables.com/Water-Level-Detector-2/*



Cat Feeder - Use an ultrasonic sensor to build an automatic cat feeder. *https://www.instructables.com/Cat-Feeder/*

Explore other activity sheets

R1AS05 - Potentiometer

PROGRAMMING RESOURCES - ACTIVITY SHEET 8

MUSIC

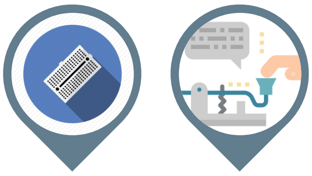
LET'S CREATE A MELODY

#R1AS08

What is it?

Let's create a melody pleasant to our ears inspired by 8-bit consoles.

Available on



Duration

30 minutes

Pre-requisites

- R1AS02 - Breadboarding: Make your first circuit!
- R1AS06 - Morse code

Level of difficulty

Advanced

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 set of LEDs
- 1 set of resistors
- 1 Breadboard
- Jumper wires

LEARNING OBJECTIVES

- Play music with a programming board

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

MUSIC - LET'S CREATE A MELODY

While we are making a lot of noises using buzzers and speakers in diverse activity sheets such as making a theremin with the distance sensor or the buttons and LEDs buzzer quiz game, let's see what can be done for creating a more pleasant for the ears melody. We will learn how to play some notes and tones using a program to play a well-known melody. To stay in the electronic sound atmosphere, we will start with music inspired by 8-bit consoles.

Chiptune, also known as chip music or 8-bit music, is a style of synthesized electronic music made using the programmable sound generator (PSG) sound chips or synthesizers in vintage arcade machines, computers and video game consoles.
Resource: https://en.wikipedia.org/wiki/Chiptune

STEP 1 - MAKE IT

1 Wire buzzer/speaker

In theory, a speaker, or a buzzer, is not polarized (it means that there no "+" nor "-"), but often you've got a pair of wires **black/red** or **signs** ("+" and/or "-") on the device.

If you are in this configuration plus the **red** (or "**+**" wire side) on **A0**, and the **black** (or "**-**" wire side) on **GND**.

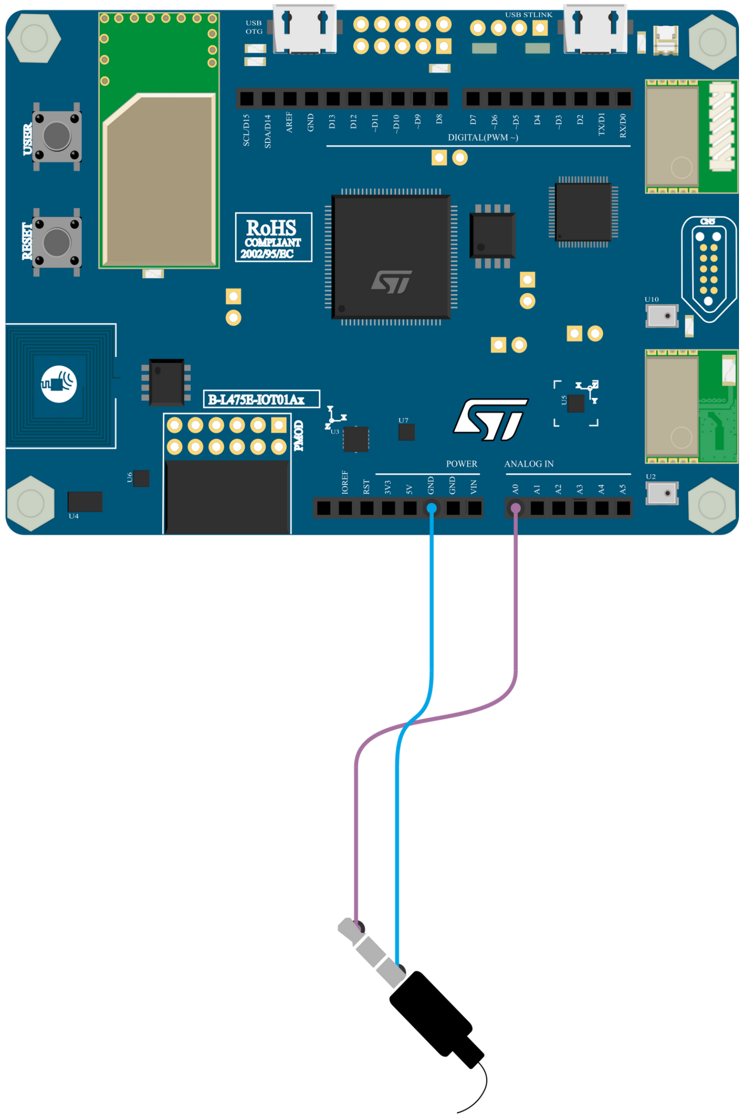
If there is no either colour or indication, just plug one wire on **A0** and the other one on **GND**.

2 Connect the board to the computer

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

3 Open MakeCode

Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.
Resource: makecode.lets-steam.eu



MakeCode simulator

MUSIC - LET'S CREATE A MELODY

STEP 1 - MAKE IT

4

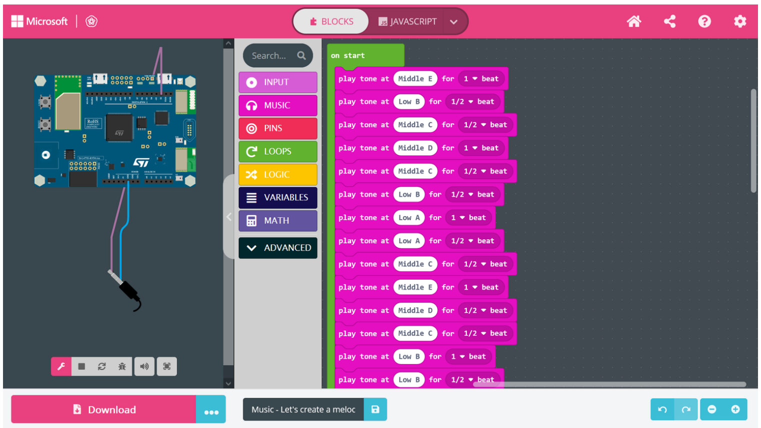
Install extension

After creating your new project, you will get the default "ready to go" screen shown here and will need to install an extension.

Extensions in MakeCode are groups of code blocks that are not directly included in the basic code blocks found in MakeCode. Extensions, like the name implies, add blocks for specific functionalities. There are extensions for a wide array of very useful features, adding gamepad, keyboard, mouse, servo and robotics capabilities and much more.

See the black **ADVANCED** button at the bottom of the column of different block groups. Clicking **ADVANCED** will show additional block groups. At the bottom is a grey box named **EXTENSIONS**.

Click on that button. In the list of extensions available, you can easily find the **Music extension** that will be used for this activity. If not directly available on your screen, you can search it using the searching tool. Click on the extension you want to use and a new block group will appear on the main screen.



MakeCode editor with the Music Extension

5

Program your board

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "**Download**" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finishes blinking and your program is ready!

Full blocks enabling the program to run

6

Run, modify, play

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). Try to understand the example and start modifying it by changing the period between two notes.

MUSIC - LET'S CREATE A MELODY

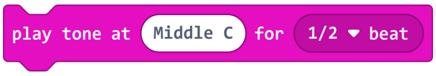
STEP 2 - CODE IT

```
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(247, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(247, music.beat(BeatFraction.Half))
music.playTone(220, music.beat(BeatFraction.Whole))
music.playTone(220, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(247, music.beat(BeatFraction.Whole))
music.playTone(247, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(262, music.beat(BeatFraction.Whole))
music.playTone(220, music.beat(BeatFraction.Whole))
music.playTone(220, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(349, music.beat(BeatFraction.Half))
music.playTone(440, music.beat(BeatFraction.Half))
music.playTone(440, music.beat(BeatFraction.Half))
music.playTone(392, music.beat(BeatFraction.Half))
music.playTone(349, music.beat(BeatFraction.Half))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(262, music.beat(BeatFraction.Whole))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(247, music.beat(BeatFraction.Whole))
music.playTone(247, music.beat(BeatFraction.Half))
```

STEP 2 – CODE IT

How does it work?

This program represents a sequence of notes with timing. The understanding of this activity is more related to music than programming.

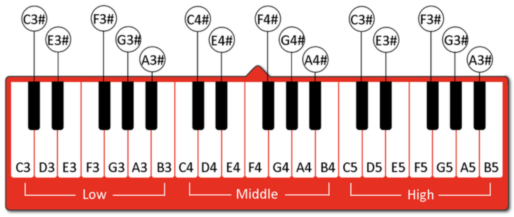
The built-in music library in MakeCode allows us to play music on our board. To play a note we use the following command: 

Where Middle C = note and 1 beat = duration.

Transcribe songs from sheet music

If we want to re-create our favourite songs, we first need a basic understanding of sheet music. Here's a reminder of the most common notes used in a musical score:

To choose the right note on MakeCode, you can click on the name of the note and make appear the virtual piano. Each key is a specific note:



Duration of the note

If we look again at the notes in a musical score, you will notice that they have different shapes and colours. These different shapes and colours indicate different durations called note values and expressed in number of beats.

Notes	Name	Value	Code
	Semibreve Whole note	4 beat	4 beat
	Minim Half note	2 beat	2 beat
	Crotchet Quarter note	1 beat	1 beat
	Quaver Eighth note	1/2 beat	1/2 beat
	Semiquaver Sixteenth note	1/4 beat	1/4 beat

MUSIC - LET'S CREATE A MELODY

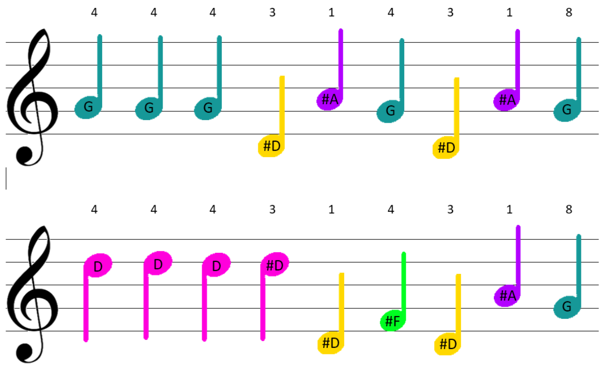
STEP 3 - IMPROVE IT

1

Write a program that **plays the following sound**:

2

Try to make the **Darth Vader's theme** with this partition:



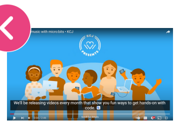
3

By using the **distance sensor** as a presence detector, make a program that **plays the music of your choice each time it detect something**.

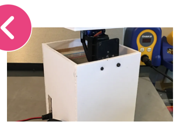
GOING FURTHER

233 music projects using Arduino. *https://create.arduino.cc/projecthub/projects/tags/music*

How to make music with micro:bits - Using alligator clips, you can connect all sorts of things to your micro:bit, including a speaker. *https://www.youtube.com/watch?v=bm7MGKspk0o*



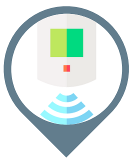
Coding with micro:bit - Part 4 - Making Music - Look at sound and audio from the micro:bit and test out a variety of different buzzers and speakers. *https://www.youtube.com/watch?v=6hxvLZSM_pM*



Making music with micro:bit - Using the built-in music library in Make Code to play music on our micro:bit. *https://www.teachwithict.com/microbit-music.html*

Explore other activity sheets

R1AS12 - Motion Detection Alarm



R1AS07 - Make a theremin with the distance sensor

PROGRAMMING RESOURCES - ACTIVITY SHEET 9

MAKE A TILT SENSOR

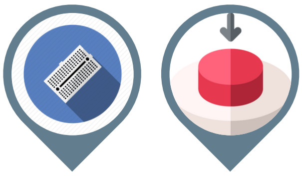
WITH THE ACCELEROMETER

#R1AS09

What is it?

Accelerometers are small sensors that can detect the force of acceleration and are great for detecting motion and orientation.

Available on



Pre-requisites

- R1AS02 - Breadboarding
- R1AS03 - Buttons and LED Display

Duration

30 minutes

Level of difficulty

Advanced

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 set of LEDs
- 1 set of resistors
- 1 Breadboard
- Jumper wires

LEARNING OBJECTIVES

- Use an accelerometer by reading the value of the acceleration on each axis
- React to shaking with events
- Detect free-fall situation

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

MAKE A TILT SENSOR WITH THE ACCELEROMETER

Acceleration makes the world go round - literally! It is the force that causes movement like a car accelerating away from a traffic light or an object falling on the ground from gravity when dropped.

To discover the potential of this motion sensor, we will write a tilt sensor that lights on a led when the acceleration is too strong. This kind of device is useful if you want to avoid cheating on classical old **pinball**.
Resource: https://en.wikipedia.org/wiki/Pinball

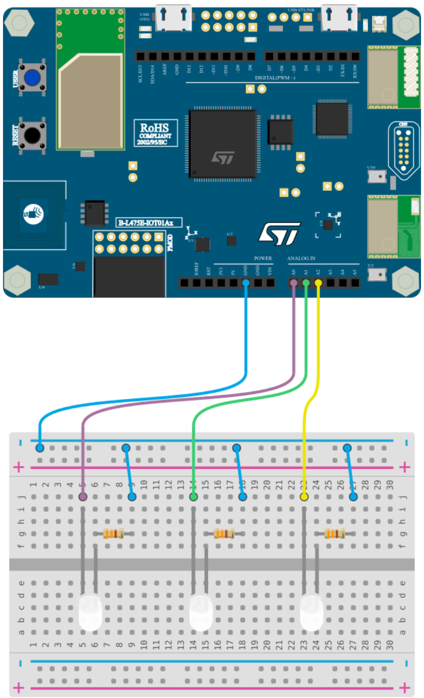
The 3-axis accelerometer is already embedded on the board so you do not need to connect anything to use it!

STEP 1 - MAKE IT

1 Wire three LEDs to the board

By using a breadboard, connect three simple LEDs to the pins of the board:

- **Green** LED to the pin **A0**
- **Blue** LED to the pin **A1**
- **Red** LED to the pin **A2**



Wiring three LEDs to the board

2 Connect the board to the computer

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

3 Open MakeCode

Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.
Resource: makecode.lets-steam.eu

4 Program your board

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "**Download**" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finishes blinking!

5 Run, modify, play

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). Try to understand the example and start to modify it by changing the thresholds to test how sensible you need to calibrate your tilt sensor. To test your tilt sensor, put the board on a table and give a little kick to the table. If your led light is on, the acceleration of your kick is strong enough!

STEP 2 - CODE IT

```
function turnOffLEDs() {
   pins.A0.digitalWrite(false) // Green
   pins.A1.digitalWrite(false) // Blue
   pins.A2.digitalWrite(false) // Red
}

forever(function () {
   turnOffLEDs()
   // X axis: green L
   if (Math.abs(input.acceleration(Dimension.X)) > 700)
      pins.A0.digitalWrite(true)
   // Y axis: blue LED
   if (Math.abs(input.acceleration(Dimension.Y)) > 700)
      pins.A1.digitalWrite(true)
   // Z axis: red LED
           if (Math.abs(input.acceleration(Dimension.Z)) > 700)
      pins.A2.digitalWrite(true)
   pause(500)
})
```

How does it work?

The program consists of lighting a LED along the axis on which the acceleration (-1g) due to gravity is detected.

The g-force of an object is its acceleration relative to free fall. On earth, this is 1g, or 9.8 meters per second squared (m/s^2). Astronauts experience unusually high and low g-forces. G-force can also be seen on rollercoasters. When the coaster goes down the drop, you are pushed back into your seat because of g-force.

Here is the configuration of the acceleration axes / LED colors:

- X axis: green LED
- Y axis: blue LED
- Z axis: red LED

Read the value acceleration

To read the value of the acceleration, MakeCode provides the function acceleration(). The value is by default in mg. We use the absolute value function abs() to ignore the direction of the acceleration. To detect the "tilt" condition, we use a threshold of 700mg. To turn off all three LEDs at the same time and improve the expressiveness of our code, we define a function turnOffLEDs().

A Function is a block of code that executes a specific task. Like a variable, it has a name to use in many places in your program. It's really useful to simplify the code and make a block of code more expressive by giving a name that explains your intent.

MAKE A TILT SENSOR WITH THE ACCELEROMETER

STEP 3 - IMPROVE IT

1. What happens if you **increase the time of pause()** inside your loop? How do you improve the **responsiveness of your tilt** sensor?

2. By using the value of the acceleration of the gravity (1g acceleration Z-Axis oriented), can you **determine the orientation of your board** (on the left side, on the bottom side, on the top side, on the bottom side)?

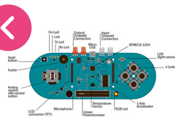
3. By using the knowledge that when a solid is in free fall, the value of the acceleration becomes close to zero very quickly, can you **modify the program to detect this situation**?

4. How can you detect if the **board is shaked** ?

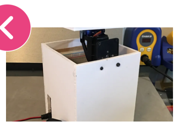
GOING FURTHER

Accelerometer - Learn more about physics principles and applications of the accelerometer. https://en.wikipedia.org/wiki/Accelerometer

Free Fall Detection Using 3-Axis Accelerometer - The easy method to determine free-fall detection with the help of a simple 3-axis accelerometer. https://www.hackster.io/RVLAD/free-fall-detection-using-3-axis-accelerometer-06383e



Level Platform Using Accelerometer - Uses an accelerometer to level a platform. https://www.hackster.io/mtashiro/level-platform-using-accelerometer-80a343



Explore other activity sheets

R1AS12 - Motion Detection Alarm

PROGRAMMING RESOURCES - ACTIVITY SHEET 10

TEXT DISPLAY

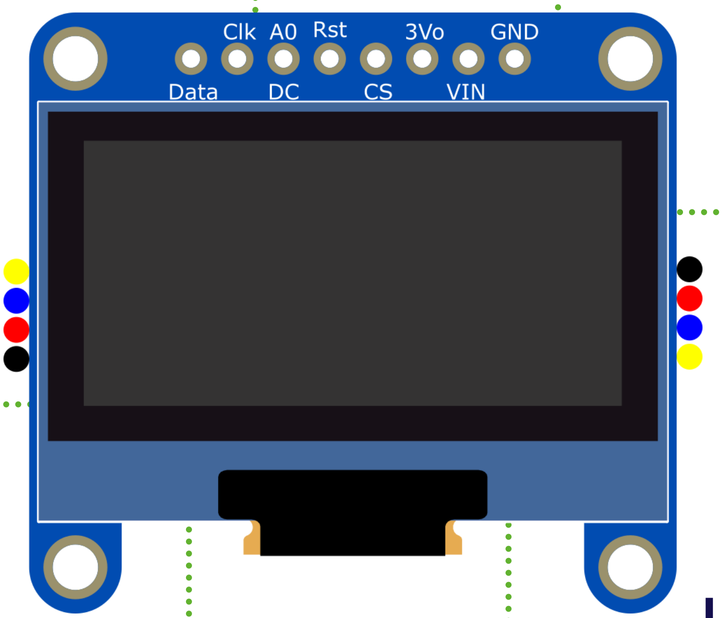
WITH AN OLED SCREEN

#R1AS10

What is it?

A screen helping you to display some pieces of information hidden inside your electronic components.

Available on



Duration

30 minutes

Pre-requisites

- R1AS03 - Buttons and LED Display

Level of difficulty

Advanced

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 OLED Display Monochrome 1.3" 128x64 OLED from Adafruit
- 1 QT Cable to connect the display to the board

LEARNING OBJECTIVES

- Connect an LCD Screen to your board
- Display text on your LCD screen
- Place text on a screen
- Display the current state of your program

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

TEXT DISPLAY WITH AN OLED SCREEN

Programming an electronic board is sometimes a very confusing activity. A microcontroller is a black box where we cannot see how it works and what happens inside. To light up your code, you can use a screen that helps you to display some pieces of information hidden inside your electronic components. This activity sheet explores how to use **SSD1306-based monochrome OLED displays** with MakeCode.
Resource: https://www.electronicwings.com/sensors-modules/ssd1306-oled-display

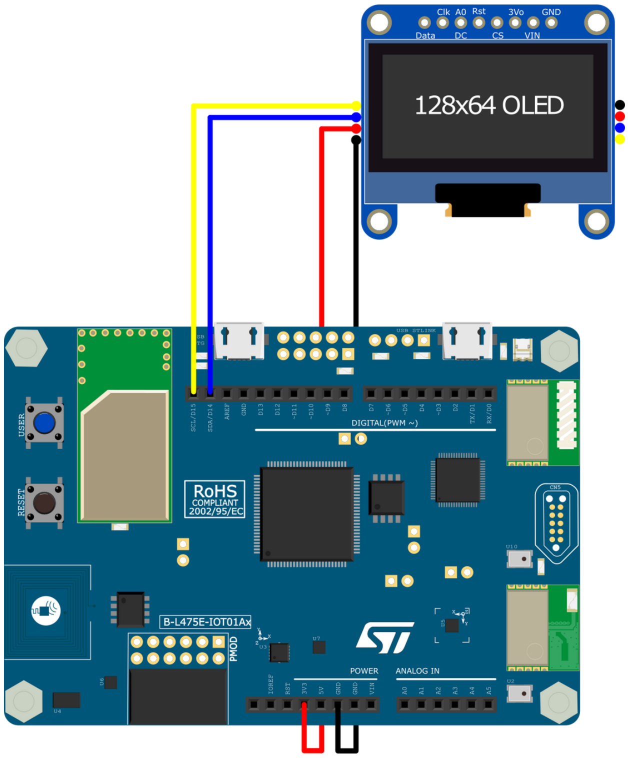
STEP 1 - MAKE IT

1 Connect the board to the display

There are two ways to wire the SSD1306 OLED to a board, either with an **I2C** or **SPI** connection. For our screen, we use the **I2C connection** through the **QWIIC**/**STEMMA** cable with the following convention :

- Black for **GND**
- Red for **V+** (**3V3**)
- Blue for **SDA** (**D14**)
- Yellow for **SCL** (**D15**)

Resources: https://en.wikipedia.org/wiki/I2C, https://en.wikipedia.org/wiki/Serial_Peripheral_Interface, https://www.sparkfun.com/qwiic, https://learn.adafruit.com/introducing-adafruit-stemma-qt/what-is-stemma-qt



Connect the board to the display

2 Connect the board to your computer

With your USB Cable, connect the board to your computer by using the **USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

3 Open MakeCode

Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.
Resource: makecode.lets-steam.eu

4 Install Extension

After creating your new project, you will get the default "ready to go" screen shown here.

TEXT DISPLAY WITH AN OLED SCREEN

STEP 1 - MAKE IT

What is an Extension? Extensions in MakeCode are groups of code blocks that are not directly included in the basic code blocks found in MakeCode. Extensions, like the name implies, add blocks for specific functionalities. There are extensions for a wide array of very useful features, adding gamepad, keyboard, mouse, servo and robotics capabilities and much more.

See the black **ADVANCED** button at the bottom of the column of different block groups. When you click on it, you'll find out additional block groups. At the bottom, there is a grey box named **EXTENSIONS**. Click on that button.

Choose the extension "**SSD1306 Display**".

5

Program your board

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "**Download**" button. Copy the binary file on the drive **DIS_L4IOT**, wait until the board finishes blinking and your program display some text!

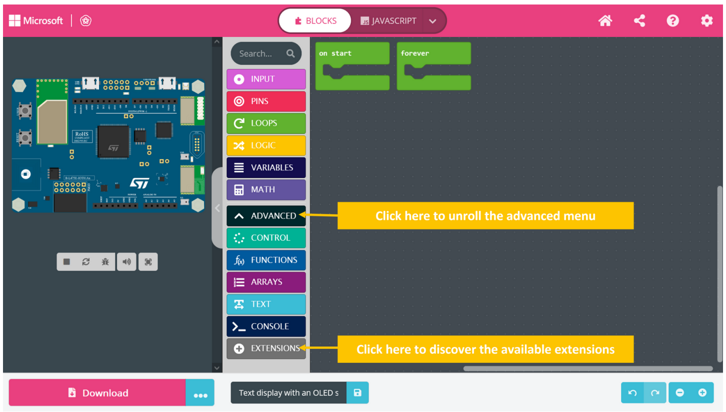
6

Run, modify, play

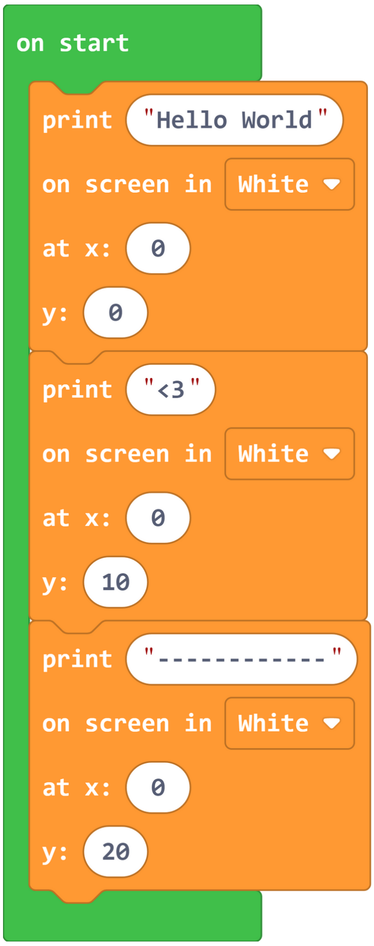
Your program will automatically run each time you save it or reset your board (push the button labelled RESET).

If everything is fine, your board will give you some friendly salutations. Try to understand the example and start to modify it by changing the text, adding as many symbols as you can or just fill the screen slowly one letter by second.

Feel free to try to display any piece of information on your program to see the current state of your board.



Advanced menu with extensions



Full blocks enabling the program to run

STEP 2 - CODE IT

```
oled.printString("Hello World", PixelColor.White, 0, 0)
oled.printString("<3", PixelColor.White, 0, 10)
oled.printString("------------", PixelColor.White, 0, 20)
```

How does it work?

You can write a line of text with the printString() function. This function takes the following parameters:

- String of text
- Text color (PixelColor.Black or PixelColor.White)
- Text X position
- Text Y position

On the SSD1306 screen, the origin (the position x=0 and Y=0) is on the top left corner.

TEXT DISPLAY WITH AN OLED SCREEN

STEP 3 - IMPROVE IT

1. Try to **center the heart** of the second line by modifying the X position of the text.

2. By **adding a loop**, create a simple text animation in the spirit of **La Linea** by using the symbols | and _. To slow down your animation, use the pause() function.
Resource: https://en.wikipedia.org/wiki/La_Linea_(TV_series)

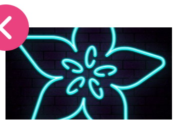
3. **Show the current state of the USER button** at each moment. What happens if you add a long sleep() inside your main loop? How to improve the responsiveness of your display?

4. **Display** the **value of all the inboard sensors**. Try to position each value at a strategic place to improve as much as possible the readability.

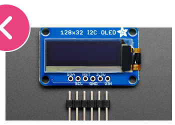
GOING FURTHER

I2C - Tutorial to learn all about the I2C communication protocol, why and how to use and implement it. https://learn.sparkfun.com/tutorials/i2c/all

QWIIC/STEMMA - Keep the level shifting/regulator, to use it with Grove/Gravity/STEMMA/Qwiic controllers. https://learn.adafruit.com/introducing-adafruit-stemma-qt/what-is-stemma-qt



OLED Display - Organic light-emitting diode (OLED or organic LED), known as organic electroluminescent (organic EL) diode. https://en.wikipedia.org/wiki/OLED



Explore other activity sheets

R1AS09 - Make a tilt sensor with the accelerometer

R1AS11 - Make a very readable thermometer

R1AS15 - Collecting data

MAKE A VERY READABLE THERMOMETER

#R1AS11

What is it?

In this activity, we will learn how easy it is to read the temperature sensor of the board and display its value.

Available on

Duration

20 minutes

Pre-requisites

- R1AS04 - Basic Light Sensor

Level of difficulty

Intermediate

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 Grove LCD I2C Text Display
- 1 Grove jumper cable

LEARNING OBJECTIVES

- Read the temperature sensor
- Use an LCD text display

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

MAKE A VERY READABLE THERMOMETER

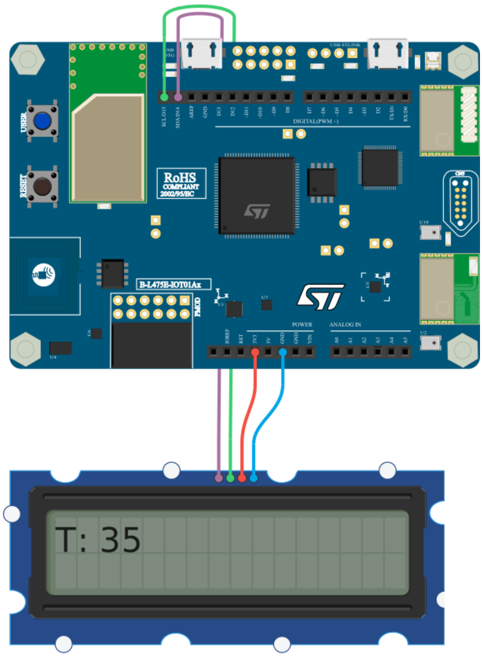
Temperature is a physical quantity that expresses hot and cold. It is the manifestation of thermal energy, present in all matter, which is the source of the occurrence of heat, a flow of energy when a body is in contact with another that is colder or hotter. In this activity, you will be able to discover the usage of the temperature sensor, integrated into the board. A temperature sensor is an electronic device that measures the temperature of its environment and converts the input data into electronic data to record, monitor, or signal temperature changes.

STEP 1 - MAKE IT

Connect the display to the board

To connect the Grove LCD screen, we will use the **I2C** Bus. For our screen, we use the **I2C** connection through the Grove cable with the following convention :

- Red for **V+** (**3V3**)
- Purple for **SDA** (**D14**)
- Green for **SCL** (**D15**)



Connect the display to the board

Connect the board to the computer

With your USB Cable, connect the board to your computer by using the **USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

Open MakeCode

Go to the Let's STEAM MakeCode editor. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.

Resource: makecode.lets-steam.eu

Program your board

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "**Download**" button. Copy the Binary file on the drive **DIS_L4IOT,** wait until the board finish blinking and your datalogger is ready!

Run, modify, play

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). If everything is working well, your board will update the LEDs' status to show that the data collection is running. Try to understand the example and start modifying it by changing the period between two measurements, adding other data from other sensors of the board. Try to display as much data as you want in many locations to understand how the temperature and evolve.

STEP 2 - CODE IT

```
lcd.clear()
forever(function () {
   lcd.setCursor(0, 0)
   lcd.ShowValue("T", input.temperature(TemperatureUnit.Celsius))
   pause(500)
})
```

How does it work?

The code consists of:

- a clear screen block
- a forever block
- a set cursor position block
- a show value block

The LCD Screen keeps a cursor to the next insert location. When we want to write somewhere on the screen, we always need to set the position of the cursor first.

Before writing on the display, we erase the screen by calling the function LCD.clear().

On each iteration of the loop, before writing something, we set the cursor to the origin of the screen (at the first character of the first line).

input.temperature(TemperatureUnit.Celsius) returns the integer value of the temperature in degrees Celsius. The value is shown on the screen with the function LCD.ShowValue(). The first parameter of this function gives the label of the value and the second, the value to be shown.

Simulation of the temperature sensor

You can play with the simulated sensor by touching the little thermometer icon shown on the board simulator. You can change the sensed value (e.g. just like touching the actual sensor on board with our finger) which accordingly changes the one on the LCD display.

MAKE A VERY READABLE THERMOMETER

STEP 3 - IMPROVE IT

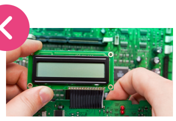
1. Try to modify the program of this activity to **read each sensor one by one and display its value on the LCD screen**. Get familiarized with the various available sensors. Try also to **utilize additional blocks from LOGIC or LCD to display text or values.**

2. **Add a condition that changes the backlight** according to the temperature value. For example, you can set the backlight in blue when the temperature is below 10° and red when the temperature is above 20°.

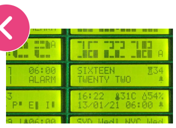
3. Put your board at different places in your classroom to **create a comparable set of data**. If you wish, you can also get in contact with other schools in your country or abroad to enlarge your dataset and work on meteorological topics.

GOING FURTHER

Liquid-crystal display - Learn more about the history and characteristics of LCF. _https://en.wikipedia.org/wiki/Liquid-crystal_display_



LCD Alarm Clock with many faces - including many of the other LCD1602 clocks found on maker sites. _https://www.hackster.io/john-bradnam/lcd-alarm-clock-with-many-faces-new-version-9352a2_



The Chrome Dino Game on an LCD Shield. _https://create.arduino.cc/projecthub/Unsigned_Arduino/the-chrome-dino-game-on-an-lcd-shield-883afb_

Light Meter - Measure and display light levels. _https://learn.adafruit.com/light-meter_

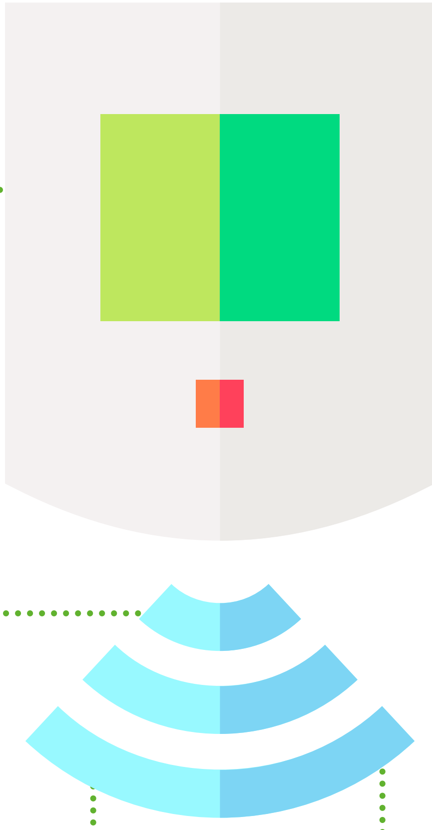
Explore other activity sheets

R1AS10 - Text Display

R1AS15 - Collecting data

MOTION DETECTION ALARM

#R1AS12



What is it?

Alarm with 2 kinds of protections: Prevent opening by force and Opening protection.

Available on

Duration

30 minutes

Pre-requisites

- R1AS09 - Make a tilt sensor with the accelerometer
- R1AS07 - Make a theremin with the distance sensor

Level of difficulty

Advanced

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 Breadboard
- 1 piezo buzzer or a speaker
- 1 small DIY cardboard box (around 15x5 cm)

LEARNING OBJECTIVES

- Use distance event block
- Use shake event block

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

MOTION DETECTION ALARM

In this activity sheet, we will work on a motion detection alarm, enabling you to keep secure all your precious and important items. For the purpose of the activity sheet, your most valuable object will be contained in a box. We will create an alarm with 2 features:

- Trigger the alarm when the box is shaken,
- Trigger the alarm when someone or something is entering the box.

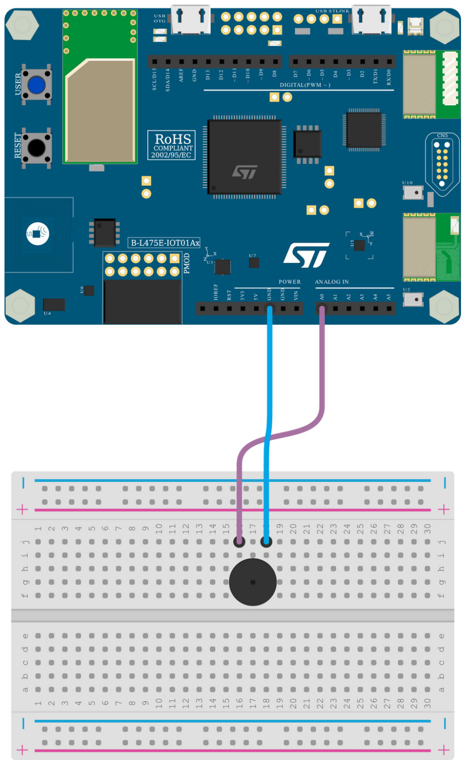
This will also enable discovering the integrated motion detector and its usages. A motion detector is an electrical device that uses a sensor to detect nearby motion. Such a device is often integrated as a component of a system that automatically performs a task or alerts a user of motion in an area. They form a vital component of security, automated lighting control, home control, energy efficiency, and other useful systems.

Resource: https://en.wikipedia.org/wiki/Motion_detector

STEP 1 - MAKE IT

1 Wire the buzzer/speaker

In theory, a buzzer is not polarized (it means that there is no "+" nor "-"), but you often have a pair of wires black/red or signs ("+" and/or "-") on the device. If you are in this configuration, attach the lead on the "+" side of the buzzer to pin **A0** and the other one to pin **GND**. If there is no colour or indication, just plug one wire on pin **A0** and the other one on pin **GND**.



Wiring the buzzer/speaker

2 Connect the board to the computer

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

3 Open MakeCode

Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.

Resource: makecode.lets-steam.eu

4 Program your board

Inside the MakeCode Javascript Editor, Copy/Paste the code available in the **Code It Section** below. Give a name to your project (more expressive than "Untitled") and click on the "**Download**" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until your alarm is ready.

5 Run, modify, play

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). Put your programmed board in your box or in a closet and see the reaction when shaking or opening it. Try to understand the example and start modifying it by changing the distance for the opening detection.

STEP 2 - CODE IT

```
let isAlarmEnable = false

// Turn on/off the alarm when "User" built-in button is pressed
input.buttonUser.onEvent(ButtonEvent.Click, function () {
    isAlarmEnable = !(isAlarmEnable)
    pins.LED.digitalWrite(isAlarmEnable)
})

// When the board is shaking
input.onGesture(Gesture.Shake, function () {
    if (isAlarmEnable) {
        music.playTone(880, 3000)
    }
})

// When the distance is over 1,000 millimeters (1 meter)
input.onDistanceConditionChanged(DistanceCondition.Far, 1000, DistanceUnit.Millimeter,
    function () {
       if (isAlarmEnable) {
            music.playTone(880, 3000)
    }
})
```

How does it work?

This program is a simple aggregation of what has already been learned in the previous activity sheets. As you can see, there are 3 parts in addition to a variable enabling to know the alarm state. Let's detail them hereunder:

Turn on/off the alarm

The first block aims to detect when the built-in button is pressed. When this event occurred, we invert the alarm state: isAlarmEnable = !(isAlarmEnable).

Shake detection

When the board is shaken, then if the alarm is turned on (if (isAlarmEnable) {...}), it means that someone tries to force our box, so we have to ring the alarm (startAlarm)!

Opening detection

Consider your box is closed. The distance between the object inside the box and the cover is nearly 0. When someone is opening your box, then your object is not anymore in direct contact with your cover. In this case, the distance between your precious treasure and the closer item will be higher than previously. You can then detect the opening of your box by approaching the variable of distance change (onDistanceConditionChanged). This will enable when we detect a distance greater than 1,000 millimetres (this distance can be adapted) with your alarm turned on, to identify that someone has opened the container and the alarm should ring (startAlarm)!

STEP 3 - IMPROVE IT

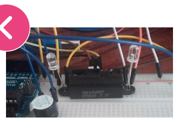
1. By **adding a second variable**, you can make the alarm tone repeating forever until the alarm is turned off.

2. By **adding a two-tone alarm noise**, you can change the melody of your alarm.

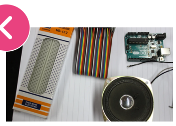
3. You can give the user a **little delay to desactivate the alarm** before ringing.

GOING FURTHER

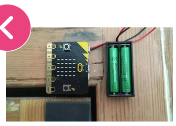
Arduino IR Alarm - Tutorial for building your own infrared alarm using an Infrared Proximity Sensor. *https://www.instructables.com/Arduino-IR-Alarm/*



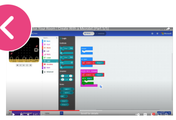
Arduino Door Alarm - Apply what you have learned to build a DIY door alarm. *https://www.instructables.com/Arduino-Door-Alarm-1/*



Radio door alarm - Tutorial to create a wireless alarm to warn you when someone opens a door. *https://microbit.org/projects/make-it-code-it/door-alarm/*



Make an Alarm for Your Room - Program an alarm for your room with a Micro:bit. *https://www.youtube.com/watch?v=aqRh9Phjcwc*



Explore other activity sheets

R1AS14 - Create an egg timer

R1AS15 - Collecting data

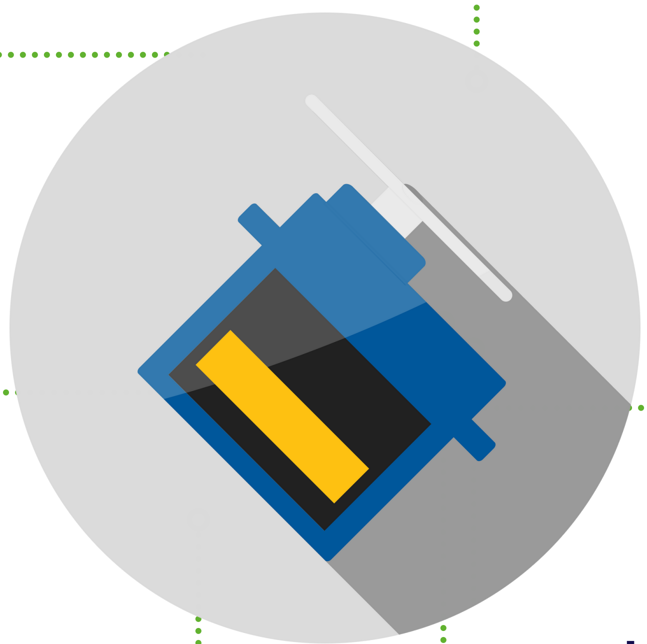
SERVOS MAKE THINGS MOVE!

#R1AS13

What is it?

Servo is a driver for keeping the position. It is suitable to control a system with constant angle change and can remain its status.

Available on



Duration

25 minutes

Pre-requisites

- R1AS03 - Buttons and LED Display

Level of difficulty

Intermediate

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 SG-90 Mini Servo(1.6kg)
- Jumper Wires

LEARNING OBJECTIVES

- Set an object in motion

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

SERVOS MAKE THINGS MOVE!

A Servo is a motor with a set of automatic control systems, which consists of an ordinary **DC motor** (rotary electrical motors that converts direct current electrical energy into mechanical energy), a reduction gear unit, a **potentiometer** (voltage divider used for measuring electric potential or voltage) and a control circuit. It can define the rotation angle of the output shaft by sending signals. Usually, a servo has a maximum rotation angle (e.g. 180 degrees).
Resources: https://en.wikipedia.org/wiki/DC_motor, https://en.wikipedia.org/wiki/Potentiometer

The servo system can be controlled by impulse, which can change its width. We use a control cable to transmit the impulse. The cycle of a servo reference signal is 20ms and the width is 1.5ms. The position defined by the servo reference signal is the middle position. Since servo has a maximum rotation angle, the definition of middle position is from this position where the maximum value and the minimum value are the same.

STEP 1 - MAKE IT

Connect the servo to the board

1

There are many ways to wire a servo to your board. You can use any analog output pin (PWM pins) to connect the control pin. In our example, we will use the **D2** pin. The servo will be connectect as follows:

- Black for **GND**
- Red for **V+** (**3V3**)
- Orange for **SIG** (**D2**)

Connect the board to the computer

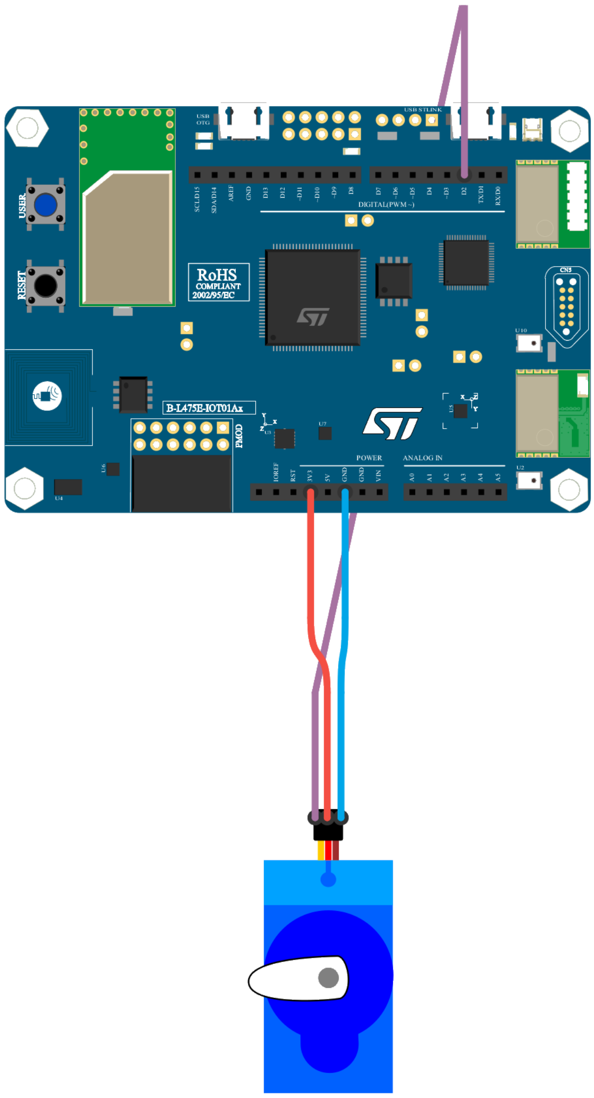
2

With your USB Cable, connect the board to your computer by using the **USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

Open MakeCode and create a new blank project

3

Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.
Resource: makecode.lets-steam.eu



Connect the servo to the board

SERVOS MAKE THINGS MOVE!

STEP 1 - MAKE IT

After creating your new project, you will get the default "ready to go" screen shown here.

4

Program your board

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below.

Before trying this program on the board, you can try it directly inside the simulator. If you change the values 0 and 180, you will see the result directly.

If not already done, think of giving a name to your project and click on the "**Download**" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finish blinking and your servo will start to move!

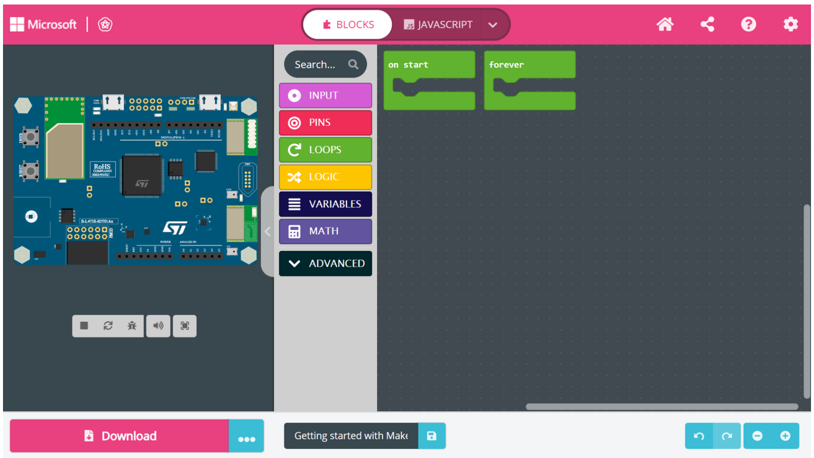
5

Run, modify, play

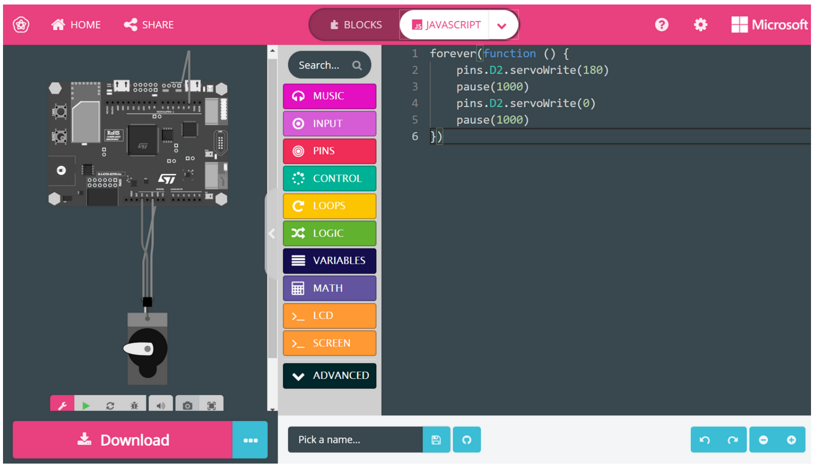
Your program will automatically run each time you save it or reset your board (push the button labelled RESET).

If everything is working well, your servo will start to move.

Try to understand the example and start modifying it by changing the period between the two moves.



Default "ready to go" screen



Makecode Javascript Editor

Your servo starts to move

STEP 2 - CODE IT

```
// goes from 0 degrees to 180 degrees
forever(function () {
   // tell servo to go to position 180 degree
   pins.D2.servoWrite(180)
   // wait for the servo to reach the position
   pause(1000)
   // tell servo to go to position 0 degree
   pins.D2.servoWrite(0)
   // wait for the servo to reach the position
   pause(1000)
})
```

How does it work?

This sample is pretty straightforward as it is the classical "blinky" adapted to a servo.

The main instruction is pins.D2.servoWrite(XXX). This instruction asks the servo to rotate at an angle of XXX degrees (as set by your specific needs depending on the project you are developing).

To move between two positions, the servo takes some time so we always need to add a delay before starting another move.

This program just sweeps left and right forever!

Compared with an ordinary DC motor, a servo rotates within a certain angle range only, while an ordinary DC motor rotates in a circle.

A servo cannot rotate in a circle. An ordinary DC motor cannot give us feedback about rotation angle but a servo can do it. Their usages are hence different.

Ordinary DC motors use a whole circle rotation as power while servo uses a certain angle of an object it controlled such as a robot joint.

SERVOS MAKE THINGS MOVE!

STEP 3 - IMPROVE IT

1. Try to **reduce as much as possible the value of the pause** to remove any motion stop.

2. Add instructions to **make a short stop in the middle position**. Adapt the delay of the pause to be sure that the stop was very short.

STOP

3. Transform this program to **make a timer with a servo**. At each step, move the servo of 3 degree. Adapt the delay such as each step take approximatly 1s.

4. Start the **sweep move** only when the USER button was clicked.

GOING FURTHER

Servomotor - Learn more about the mechanism and control operation of servomotor. *https://en.wikipedia.org/wiki/Servomotor*

Servo Motors with micro:bit - All about buttons and their use in MakeCode with Shawn Hymel, Technical Content Creator. *https://www.youtube.com/watch?v=okxooamdAP4&t=200s, https://shawnhymel.com*

DIY Color Sorting Robotic Arm - Learn how to make your own DIY color sorting robotic arm using ultrasonic and IR sensors. *https://thestempedia.com/project/diy-color-sorting-robotic-arm/*

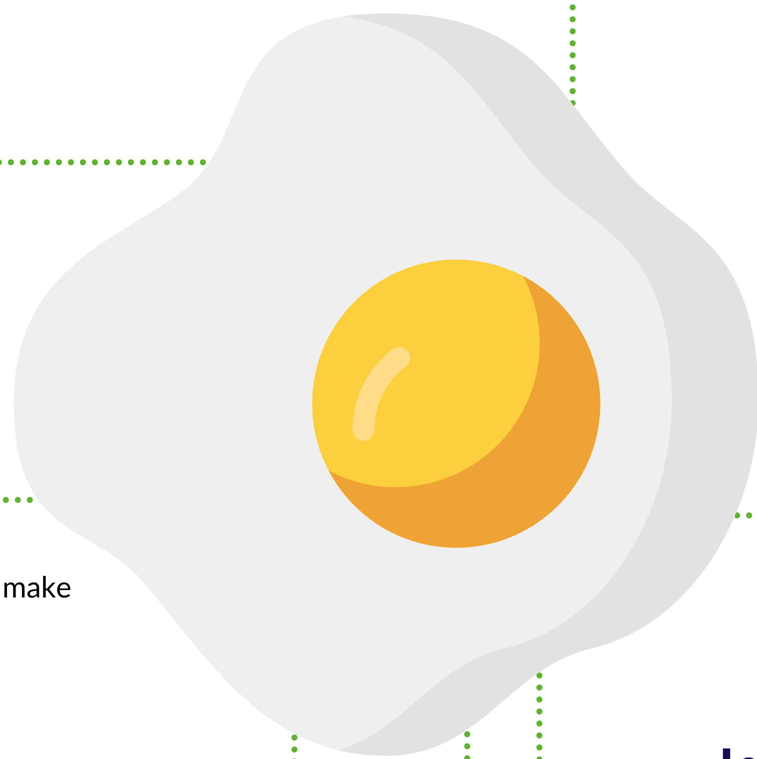
Explore other activity sheets

R1AS14 - Create an egg timer

PROGRAMMING RESOURCES - ACTIVITY SHEET 14

CREATE AN EGG TIMER

#R1AS14



What is it?

Let's create a simple but useful object, an egg timer! This activity will enable to applicate knowledge acquired on servos, as a system control solution.

Available on

Duration

35 minutes

Pre-requisites

- R1AS13 - Servos make things move!

Level of difficulty

Advanced

Material

- 1 Programming board "***STM32 IoT Node Board***"
- 1 Micro-B USB Cable
- 1 SG-90 Mini Servo (1.6kg)
- Jumper Wires
- 1 small cardboard sheet (20cm*10cm)
- 1 Sturdy Wood Sticks (less than 10cm)

Extended activity

LEARNING OBJECTIVES

- Create a physical timer
- Use a servo to display data
- Make a calibration process to improve the precision of the timer

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

CREATE AN EGG TIMER

In this activity, we will create a simple but useful object, an egg timer using programming and DIY practices! After performing it, you will be a real French cook! To boil correctly an egg, French people use the rule called **3,6,9**! This rule gives the exact time in minutes for correctly bake an egg depending on your cooking objectives:

- 3 minutes for soft-boiled eggs - *Œufs à la coque*
- 6 minutes for boiled eggs - *Œufs mollets*
- 9 minutes for hard-boiled eggs - *Œufs durs*

STEP 1 - MAKE IT

1 Prepare your electronic hardware

Wire correctly your board and your servo using the activity sheet #R1AS13 - Servos make things move!

2 Create the clock hand and attach it to the servo horn

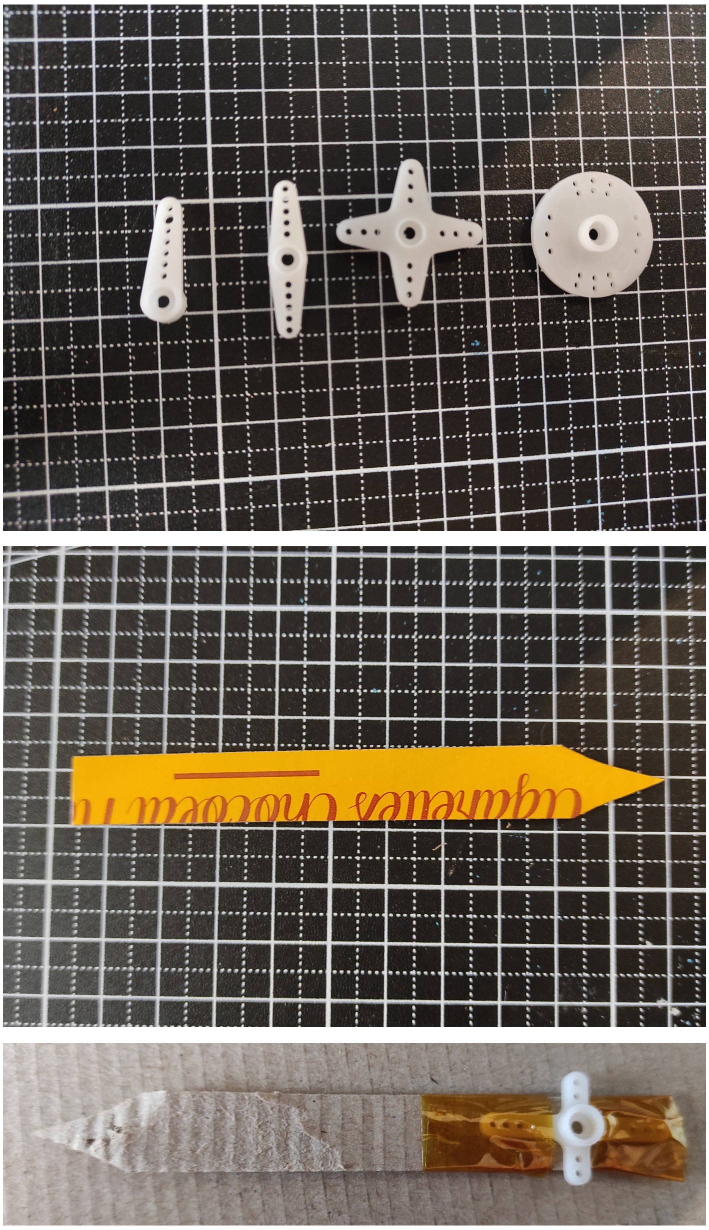
Take the sturdy Wood Sticks and attach it to the servo horn.

Servo horns are attachments that fit over the output shaft and allow you to mechanically link the servo output to the rest of your mechanism. Servos are usually supplied with an assortment of servo horns.

Unfortunately, the exact horns included are usually not specified and can vary.

And, since servo output shafts and their splines vary, horns are often incompatible between brands and models of the servo.

The easiest way to attach your clock hand is to use an elastic band but you can also use hot glue or scotch.



Create the clock hand and attach it to the servo horn

STEP 1 - MAKE IT

3 Create the timer front panel

On the cardboard, make a small hole of the size of your servo shaft. The hole should be on the middle of the longer side of your cardboard.

Put the servo behind and attach the clock hand on the shaft of the servo.

Turn the horn in the minimum position (angle 0°) and fix the servo so that the clock hand be horizontal. With a pen, make a small mark to indicate the 0s. Turn the horn in the maximum position (angle 180°) and make a small mark to indicate the 180s.

Create the timer front panel

4 Connect the board to the computer

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

5 Open MakeCode and create a new blank project

Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.
Resource: makecode.lets-steam.eu

6 Program your board

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. Before trying this program on the board, you can try it directly inside the simulator. If you click on the USER button, you will see your timer start. . If not already done, think of giving a name to your project and click on the "Download" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finish blinking and your servo will start to move!

7 Run, modify, play

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). If everything is working well, your servo will start to move.

STEP 2 - CODE IT

```
input.buttonUser.onEvent(ButtonEvent.Click, function () {
    for (let pos = 0; pos <= 179; pos++) {
         pins.D2.servoWrite(pos)
         pause(1000)
    }
    for (let i = 0; i < 5; i++) {
         pins.D2.servoWrite(0)
         pause(1000)
         pins.D2.servoWrite(180)
         pause(1000)
    }
})
```

How does it work?

The main part of the code is about the buttons interactions. These interactions are made with the input.buttonUSER.onEvent function.

When you click on the button USER, you will start the timer by changing the position of the servo to one degree each second.

When you have finished counting from 179 to 0, you start to move quickly your servo to signal the end of the timer.

STEP 3 - IMPROVE IT

1

By **adding a servo**, make a second indicator enabling to know the status of your egg cooking (*raw*, *soft boiled*, *boiled*, *hard-boiled*).

2

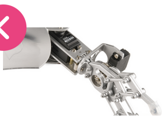
Change the final animation of the timer **by adding a buzzer** to make more sound.

3

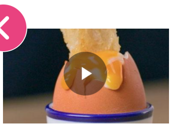
The current version of the program is not calibrated, your timer will give you an approximate value. If you want to be a more scientific egg cooker, you need to **follow a calibration process.** To calibrate a timer, **use a reference clock**. You can easily use the clock of your smartphone for instance to measure the duration of the timer. To reduce the uncertainty, you will **repeat the measurement many times** (e.g. ten times is enough) to be able to calculate the average value and use a **cross product to find the correct delay value.**

GOING FURTHER

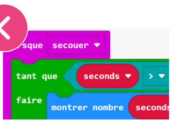
Pulse Width Modulation - Learn more about the pulse width modulation digital signal https://learn.sparkfun.com/tutorials/pulse-width-modulation/all



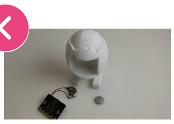
How to boil an egg perfectly - Learn how long to boil an egg to achieve the perfect consistency. https://www.bbcgoodfood.com/howto/guide/how-boil-egg-perfectly



Countdown Timer - Make a countdown timer and see the seconds tick by on micro:bit watch. https://makecode.microbit.org/projects/watch/timer



Micro:bit Egg Timer - Make a fun timer to guarantee the perfect cooking time of eggs using 3D printing and micro:bit. https://www.myminifactory.com/object/3d-print-micro-bit-egg-timer-18361



Explore other activity sheets

R1AS15 - Collecting data

PROGRAMMING RESOURCES - ACTIVITY SHEET 15

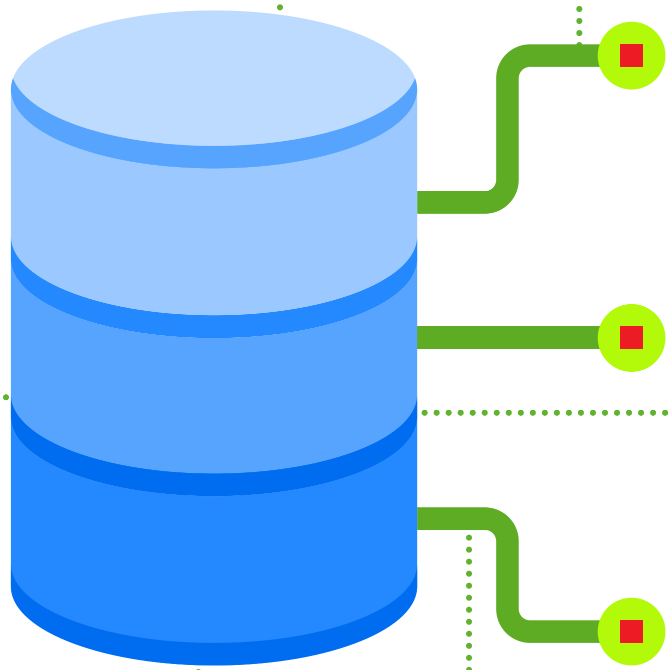
COLLECTING DATA

#R1AS15

What is it?

This activity sheet will focus on how collecting data from an environmental sensor and to export them to a computer enabling to perform a simple analysis with a spreadsheet.

Available on



Duration

50 minutes

Pre-requisites

- R1AS04 - Basic Light Sensor

Level of difficulty

Advanced

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable

LEARNING OBJECTIVES

- Read a sensor value
- Store the sensor value on the flash memory of the board
- Export all the collected values in a Comma Separated Values (CSV) file
- Add an extension to MakeCode

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

COLLECTING DATA

A sensor measures a physical quantity and converts it into a signal which can be transformed into a numerical value by a microcontroller. On your program, you can use this value to adapt the behaviour of your algorithm (for example close the door of the house when the value of the light sensor is becoming low).

When you want to conduct a scientific experiment, just one value does not give you enough information to make assumptions. You need to observe how the value of your sensor will evolve over a long period of time.

This activity sheet explores how to collect data from an environmental sensor and how to export them to a computer enabling to perform a simple analysis with a spreadsheet.

STEP 1 - MAKE IT

1 Connect the board to the computer

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

2 Open MakeCode and create a new blank project

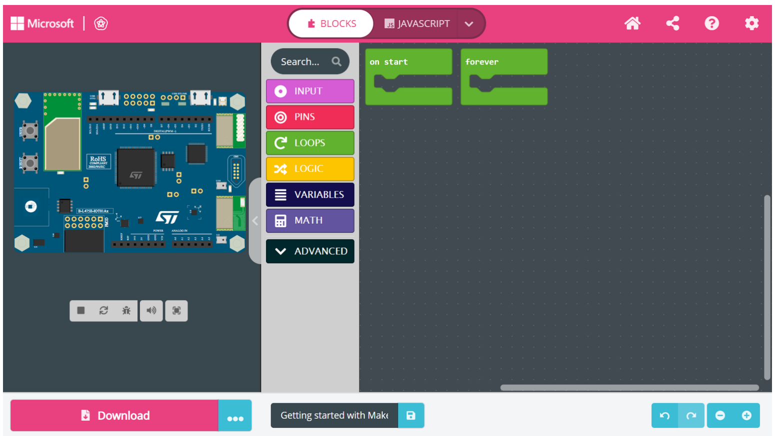
Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.

Resource: makecode.lets-steam.eu

3 Install Extension

After creating your new project, you will get the default "ready to go" screen shown here and will need to install an extension.

What is an Extension?
Extensions in MakeCode are groups of code blocks that are not directly included in the basic code blocks found in MakeCode. Extensions, like the name implies, add blocks for specific functionalities. There are extensions for a wide array of very useful features, adding gamepad, keyboard, mouse, servo and robotics capabilities and much more.



Ready to go MakeCode screen

STEP 1 - MAKE IT

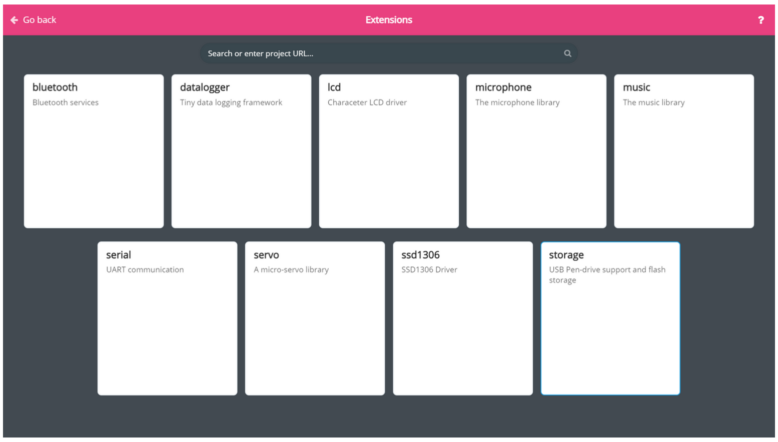
See the black **ADVANCED** button at the bottom of the column of different block groups. Clicking **ADVANCED** will show additional block groups. At the bottom is a grey box named **EXTENSIONS**. Click on that button. In the list of extensions available, you can easily find the **Datalogger extension** that will be used for this activity. If not directly available on your screen, you can search it using the searching tool. Click on the extension you want to use and a new block group will appear on the main screen.

Advanced functionalities appeared

4

Program your board

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "Download" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finish blinking and your datalogger is ready!



List of extensions and searching tool

5

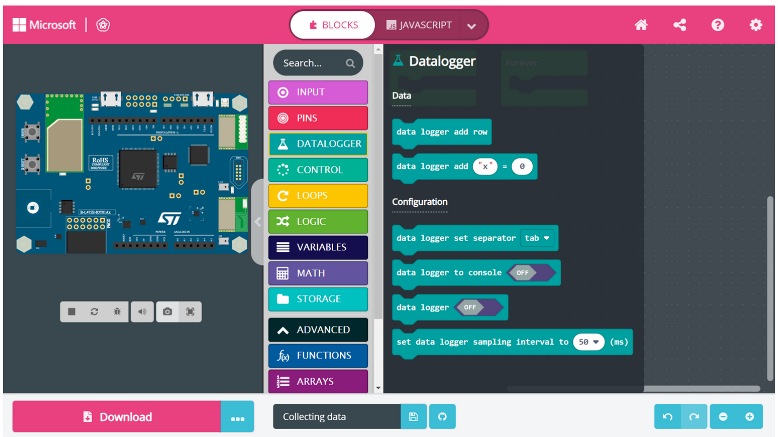
Use your datalogger

The program logs the data to the flash memory (LED 1 is on) until you press the USER button, at which time the LED2 is on. This is your indication the data logging is stopped and you can copy the data to your computer.

6

Get your data

With your USB Cable, connect the board to your computer by using the USB OTG connector (the left one when you look at the board from the upside). When your project is logging, a new flash drive should appear named **MAKECODE**.
The **SPIFLASH** directory contains program data. Logging data is written to a file named log.csv.
Resource: wikipedia.org/wiki/Serial_Peripheral_Interface



Datalogger and associated blocks

Be sure you have stopped logging your data prior to accessing log.csv with any program. Pressing Reset or unplugging the board without pausing the data logging with the USER button will corrupt the log.csv file! Press the USER button to stop logging which will properly close the file and allow copying the data.

STEP 1 - MAKE IT

Copy the **log.csv** file to your hard drive to save it and view it later.

7

View your data

Open a **spreadsheet program** such as *Google Sheets, Microsoft Excel, macOS Numbers* etc. Open the log.csv file. The spreadsheet should recognize the **CSV** (if your program does not, you might have to specify you are trying to open a **CSV** file or use an import feature). In Google Sheets, the file just opens correctly.
Resource: https://en.wikipedia.org/wiki/Comma-separated_values

The sep= and NAN lines may be ignored if they appear.

Line 2 has the headings for the data you read. Time first, then for the example: temp, light, and soil moisture readings in each column.

The data can go quite a way as the example logs data every 10 s. You can log data slower, 60 seconds (1 minute), 300 seconds (5 minutes), etc. The data may be used for analysis or to graph values over the time period. Using the Google Sheets graph function, push the graph button on the toolbar and without any formatting, you have a great graph!

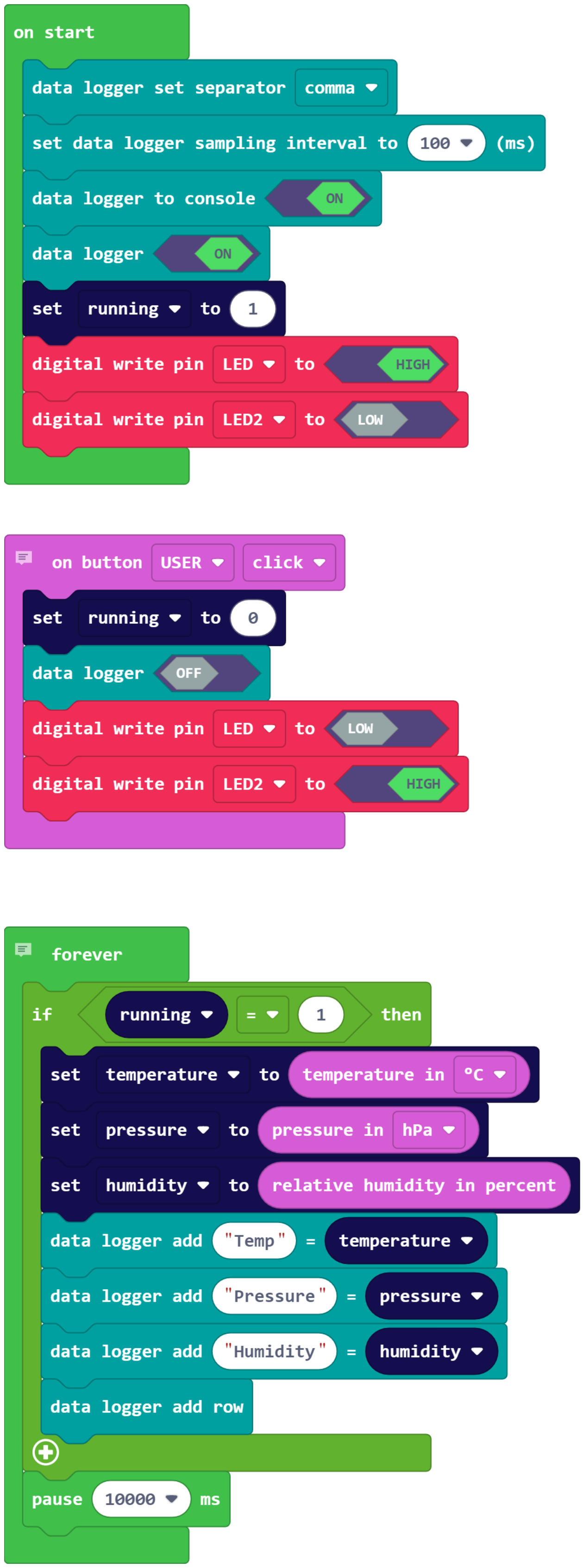
8

Run, modify, play

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). If everything is working well, your board will update the status LEDs to show that the data collection is running.

Try to understand the example and start modifying it by changing the period between two measurements, adding other data from other sensors of the board.

Feel free to try to log as much data as you want in so many locations to understand how the temperature, humidity and pressure evolve.



Full blocks enabling the program to run

STEP 2 - CODE IT

```
//Init the data collection
let running = 0
datalogger.setSeparator(LogSeparator.Comma)
datalogger.setSampleInterval(100)
datalogger.sendToConsole(true)
datalogger.setEnabled(true)
running = 1
pins.LED.digitalWrite(true)
pins.LED2.digitalWrite(false)

//Stop the data collection after the USER button is clicked
input.buttonUser.onEvent(ButtonEvent.Click, function () {
    running = 0
    datalogger.setEnabled(false)
    pins.LED.digitalWrite(false)
    pins.LED2.digitalWrite(true)
})

//Collect the sensors data every 10s
forever(function () {
    if (running == 1) {
        let temperature = input.temperature(TemperatureUnit.Celsius)
        let pressure = input.pressure(PressureUnit.HectoPascal)
        let humidity = input.humidity()

        datalogger.addValue("Temp", temperature)
        datalogger.addValue("Pressure", pressure)
        datalogger.addValue("Humidity", humidity)
        datalogger.addRow()
    }
    pause(10000)
})
```

STEP 2 - CODE IT

How does it work? Initialize the data collection:

To download the file on a computer, we need to stop data collection when we want. The variable running allows knowing the current state of the data collection process. When the value is 0, the data collection is off and when it is 1, the data collection is running.

The three following instructions configure the data logger with the following parameters :

- A comma is used as a field separator in the CSV File
- The minimal interval between two rows is set to 100 ms
- All the data are sent to the MakeCodeconsole to show the current data directly inside MakeCode

After the configuration, the data collection process is activated and the status led is used to show the current state of the process.

Stop the data collection after the USER button is clicked

To stop the data collection process, we use the USER button. When the button is clicked, the data logger is disabled, the status LEDs updated and running is set to 0.

To handle the asynchronicity of the button click (a button click can happen at any step of our program), we use the Event mechanism of MakeCode. This mechanism allows running a specific set of instructions when a specific condition appears. In our case, the event is "the USER button is clicked".

When the data logger is disabled, there is no more writing on the log file, so we have no risk to corrupt it.

Collect the sensors data every 10s

In the main loop, just read the data and send it to the data logger if the variable running, is set to 1. The pause at the end of the loop allows fixing the period between two measurements. If we want to observe a longer experiment, we will probably increase this value.

STEP 3 - IMPROVE IT

1. **Add a battery pack** to your board to make experiments on environmental sensors in many contexts.

2. Allow to restart the data collection process by **clicking again on the USER button**.

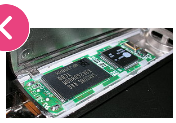
3. **Produce some graphic** that compare multiple data collection sessions.

4. **Record sensors remotely** by using one board for data logging and another board to gather sensor values in several places.

5. **Conduct a physics experiment** into the forces acting on a board as it **spins in a salad spinner (centrifuge)**. Do you guess what will happen? (Bear in mind that the accelerometer on the board can only read forces up to 2g, twice the force of the Earth's gravity – if you spin it fast it may experience forces that are too large for it to register.)

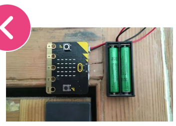
GOING FURTHER

Flash memory - Learn more about flash memory, an electronic non-volatile computer memory storage medium. *https://en.wikipedia.org/wiki/Flash_memory*

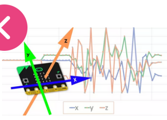


Event handlers - Discover event handlers i.e. code that is associated with a particular event, such as "button A pressed". *https://makecode.microbit.org/reference/event-handler*

Make It Log - Log your Circuit Playground Express data directly into a spreadsheet. *https://learn.adafruit.com/make-it-data-log-spreadsheet-circuit-playground/logging-via-android-phone*



MakeCode data logger - Use micro:bit as a wireless data logger recording readings from its sensors. https://microbit.org/projects/make-it-code-it/makecode-wireless-data-logger/



Explore other activity sheets

R1AS07 - Make a theremin with the distance sensor

R1AS11 - Make a very readable thermometer

R1AS12 - Motion Detection Alarm

ACTIVITY SHEETS & TEMPLATES

INCLUSION AND EQUITY: RESOURCES FOR STUDENTS & TRAINERS

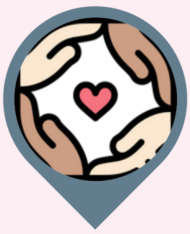
Authors: Mercè Gisbert Cervera, Carme Grimalt-Álvaro

The activity sheets available in this chapter aim to provide a space for Let's STEAM learners to reflect on how to adapt their activities to the needs of all the students in their classes. These activities can be directly promoted to students to jointly reflect on the ethical and safety issues that may arise when creating and sharing resources and data from and to the digital ecosystem.

In this chapter you will find two sets of activity sheets:

- **Activities for learners** - *You are participating in Let's STEAM as a trainee*: gaining knowledge about inclusiveness and equity
- **Hints for trainers/teachers** - *You are a Let's STEAM trainer and/or you have completed the Let's STEAM programme and want to train your students on inclusion and equity issues*: provide the necessary information and content if you are ready to initiate discussions on this topic in the classroom with your students

Inclusive design



Inclusive implementation

Data privacy & security

Sharing and promotinG.

ACTIVITIES FOR THE LEARNERS

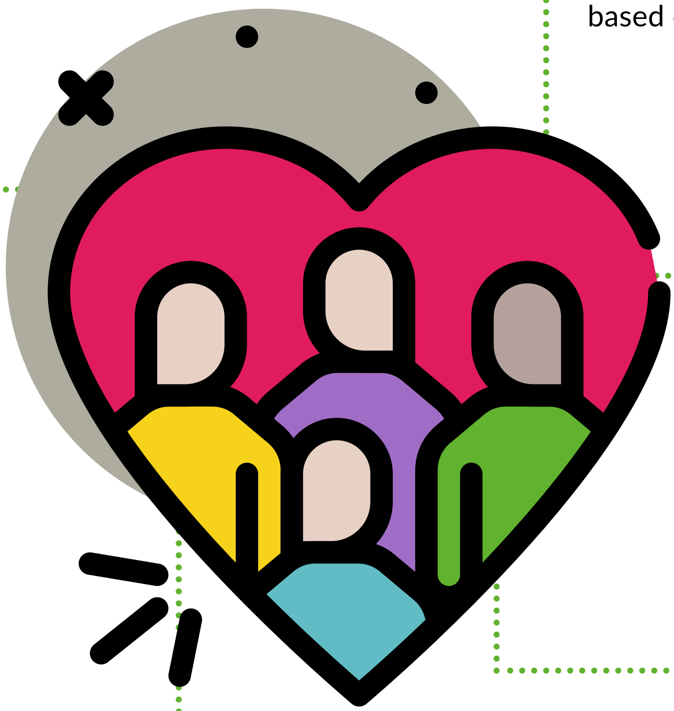
You are participating in Let's STEAM as a trainee

Reminder: Feel free to reuse the activity sheets and templates presented in this section in your classroom and share them with your students! You are free to print, reproduce, modify, reuse and draw inspiration from all the resources in this manual without constraint. Our content has been developed entirely under a Creative Commons license.

RESOURCES ON INCLUSIVE EDUCATION - ACTIVITY SHEET 1

INCLUSIVE DESIGN

#R2AS01



What is it?

This activity is willing to analyse and transform the Let's STEAM activities adapting them to students based on their needs.

Modalities

Teachers will work in groups of 4-5 people. They are expected to work autonomously, following the guidelines.

Duration

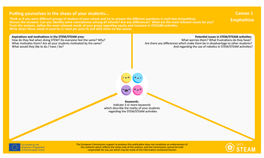
1h45

Level of difficulty

Basic

Material

- Specific canvas on empathising the different activities



LEARNING OBJECTIVES

- Identify the needs regarding inclusive design and suggest transformations to increase its inclusiveness
- Analyse and transform designed STEM educational materials and activities to adapt and increase the inclusiveness especially regarding potential groups of students at a disadvantage which are students with special needs, and women, racial minorities, and low socioeconomic students

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

STEP 1 - INSPIRE

15 min.

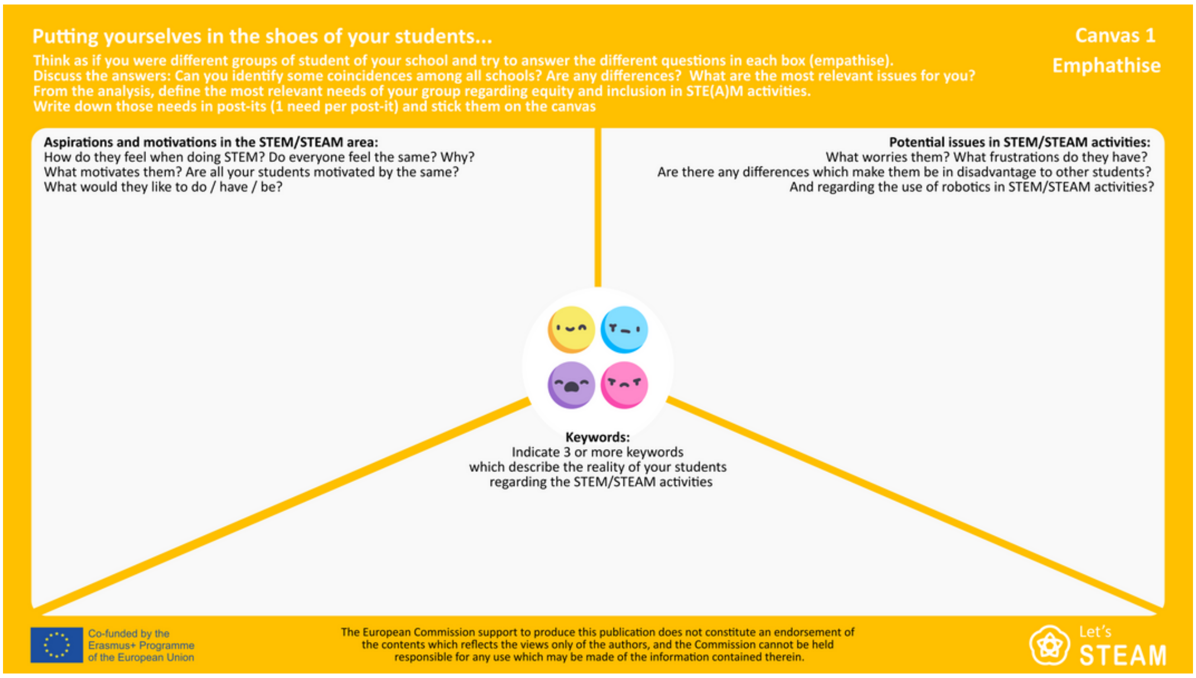
Meet as a team and introduce each other. Introduce yourselves and briefly explain where do you come from (the type of school, your role... etc.). Reflect on your students' traits, background and relationship with STEM (students with special needs, girls, racial minorities, and low socioeconomic background) and present if they are already special policies or practices in your schools to promote equity and inclusion.

STEP 2 - CONTEXTUALISE & EMPHATHISE

20 min.

1

Answer **individually** the questions that are suggested in the **Canvas #1 - Emphatise** available **to all in this coursebook on page 102** and in this activity sheets. It will enable putting yourself in the shoes of your students.



2

Analyse your answers: what **coincidences** can you identify from the contributions of all members? Is there a difference? What are the most **relevant issues** as a group?

Based on the analysis of the answers given, which most relevant issues can be identified regarding your students' relationship with STEM activities in terms of:

- **Aspirations** and **motivations** regarding STEM activities and fields
- **Previous experiences** in STEM activities with trainees
- Previous experiences in activities **promoting computational thinking skills**

Use Canvas #2 - Checlist page 103 for stimulating questionning

3

Write these issues on different post-it notes and paste them on the canvas (one need per post-it).

STEP 3 - ANALYSE

15 min.

1 **Individually**, read again the designed Let's STEAM activities. Imagine yourself implementing some of the different suggested activities with your students.

2 Think individually and try to consider **which potential issues will appear when these activities are implemented with your students** based on the needs you have identified in the previous part of the activity (contextualize). You can read again the Canvas 1 of your group if you need it to refresh what you discussed in your group.

3 Among the potential issues brainstormed, try to focus on the **ones more closely related to equity and inclusion issues**.

4 Write down in a document those potential issues, being as specific as possible. You will be asked to explain those issues to your peers, so maybe a little bit of context can be useful to understand its potential impact on the activity. Feel free to share your conclusions with the Let's STEAM community!

STEP 4 - DESIGN & IDEATE

55 min.

Return to your group and remember that the success of co-creation is based on the free association of ideas, postponing trials, building on the ideas of others and enjoying teamwork!

SHARE YOUR THOUGHTS

1 Share your ideas and listen to the rest of the members of the group. Try to **identify common issues** that might appear when trying to implement and engage students in the Let's STEAM activities. If difficult to reach an agreement, prioritize the main issues and select the top 3.

REVISE AND REDESIGN

2 Based on these selected or prioritized issues, try to revise and redesign one Let's STEAM activity so it can be **more inclusive and equitable** for your students. Try to concrete:

- Which inquiry question would best engage/be more relevant for your students?
- Which plan for collecting and showing data would best engage your students? (you can think both about the experiment design and the robotics elements needed)
- Which solution would best engage or be more relevant for your students?
- Which practices or additional resources/activities can contribute to having a more positive impact on the engagement of all students in the Let's STEAM activities?

Use Canvas #2 - Checlist page 103 for stimulating questioning

CONCLUDE

3 Share your proposal of the revised Let's STEAM activity to other members of other groups participating in the training or to the community. Try to explain to others which modifications you introduced and why you introduced them, specifically relating to the equity and inclusion issues identified in your group. You are invited to provide feedback and suggestions to help other groups improve their designs.

EMPHATHISE

#R2AS01

Putting yourselves in the shoes of your students...

Canvas 1
Emphathise

**Think as if you were different groups of student of your school and try to answer the different questions in each box (empathise).
Discuss the answers: Can you identify some coincidences among all schools? Are any differences? What are the most relevant issues for you?
From the analysis, define the most relevant needs of your group regarding equity and inclusion in STE(A)M activities.
Write down those needs in post-its (1 need per post-it) and stick them on the canvas**

Aspirations and motivations in the STEM/STEAM area:
How do they feel when doing STEM? Do everyone feel the same? Why?
What motivates them? Are all your students motivated by the same?
What would they like to do / have / be?

Potential issues in STEM/STEAM activities:
What worries them? What frustrations do they have?
Are there any differences which make them be in disadvantage to other students?
And regarding the use of robotics in STEM/STEAM activities?

Keywords:
Indicate 3 or more keywords
which describe the reality of your students
regarding the STEM/STEAM activities

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

R2AS01 - INCLUSIVE DESIGN - CANVAS #2

CHECKLIST - WHERE TO START?

#R2AS01

Here is a list of basic questions to consider while addressing inclusive design! There are no right or wrong answers, just different experiences that are important to be shared! Comment on your own feedback and experience below each topic!

- [ ] 1 **Have you considered how students with special needs might face difficulties in the accessibility to STEM activities/ activities using digital technology in your lessons? What is your experience with it?**

- [ ] 2 **Have you considered how students with special needs might have difficulties in understanding the purpose and what they are expected to do in educational activities?**

- [ ] 3 **Have you considered how women, racial minorities, and students from low socioeconomic backgrounds might feel that STE(A)M activities are "not for them"?**

- [ ] 4 **Have you considered how students from diverse cultural backgrounds may have issues understanding the main language of the lesson?**

- [ ] 5 **Have you considered how students from low socioeconomic backgrounds will have difficulties accessing the resources?**

- [ ] 6 **Have you considered how to improve the design of your STE(A)M activities so they can be more aligned with the universal design for everyone?**

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

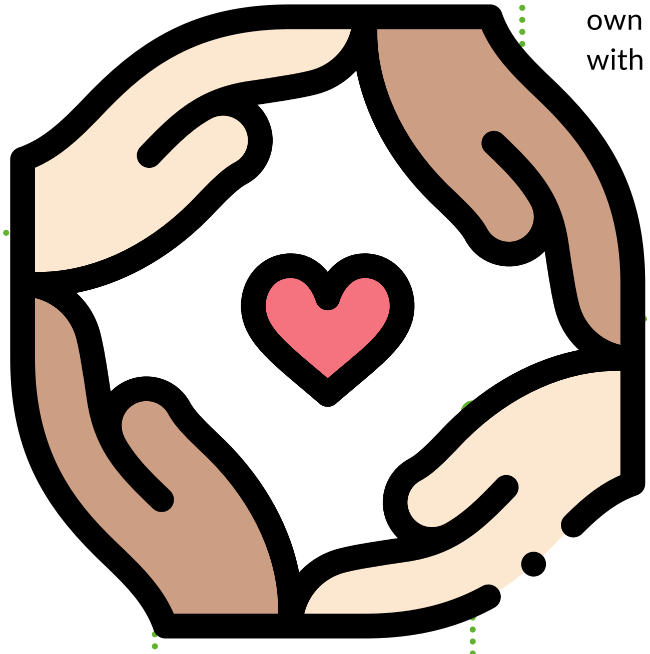
RESOURCES ON INCLUSIVE EDUCATION - ACTIVITY SHEET 2

INCLUSIVE IMPLEMENTATION

#R2AS02

What is it?

Based on their reflections in R2AS01, you will develop their own resources to be implemented with inclusive design.



Modalities

Work autonomously, following the guidelines.

Duration

1h30

Material

- Table of initial analysis
- Table of final analysis
- Additional materials for enlarging the context

Level of difficulty

Basic

LEARNING OBJECTIVES

- Analyse and transform designed STEM educational materials and activities to adapt and increase the inclusiveness especially regarding potential groups of students at a disadvantage which are students with special needs, and women, racial minorities, and low socioeconomic students
- Identify successful strategies that could be implemented in different educational contexts

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

STEP 1 - ORIENTATE

30 min.

ASSESSING THE IMPACT OF THE IMPLEMENTATION

Try to remember from the first activity sheet M3AS1 "Inclusive Design", what other trainees suggested to you based on their previous experiences. Introduce the modifications you consider appropriate to **improve the design of the activity**. Discuss in groups how will you know if the aims of the activity have been reached and possible pieces of evidence that you may collect.

Now it is time for you to try out this magnificent activity you have designed in your group!

STEP 2 - INVESTIGATE

40 min.

TO WHICH EXTENT THE AIMS HAVE BEEN REACHED?

1

In this part, you are invited to **implement the activity designed and assess to which extent the inclusion and equity aims are achieved.** As well, you can also consider the extent of achievement of **educational aims**.

<u>During a training session</u>: The trainer will provide you with the details for the implementation with your group peers depending on the training. As a trainee, you will conduct the activity as if you were in your regular classroom and the other trainees in your group will act as your students. Note that is important to define with your group, your type of students and try to reproduce potential issues that can occur in a regular lesson. **The implementation would not be longer than 7-12 minutes.**

<u>Within your classroom</u>: You will implement your designed and improved activity with your students and reflect on this activity using the analysis tables and guidelines given in this activity sheet, if possible reflecting about it during a training session with your group, or within your school, with colleagues. You can also interact with your local trainer dedicated to the implementation of the Let's STEAM project.

2

In both cases, you will be invited to collect the evidence agreed upon and fill in the table of initial analysis given in this coursebook page 106 after your implementation with your students or with your peers' (depending on the modality of this activity).

STEP 3 - CONCLUDE

20 min.

LEARNING FROM OUR EXPERIENCE

At the end of all implementations, read again what you have written and try to identify the **main issues regarding inclusion and equity** in the implementation of Let's STEAM activities in all cases. If it helps you, you can use the **table of final analysis (page 107 in this coursebook)** to structure the reflection.

TABLE FOR INITIAL ANALYSIS

#R2AS02

Identification of the episode/classroom: _____________________________________

FIRST IMPRESSION (INDIVIDUAL)

What catches your attention, in general? Identify what you think is the most relevant part in the episode/classroom and that needs to be highlighted (write down at least 3 ideas)

1

1.

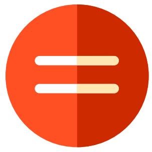
2.

3.

INITIAL INTERPRETATION (INDIVIDUALLY)

How do you interpret the actions that students show in the episode/classroom? Are all children equally participating?

2



How can you interpret the role of the teacher in the episode, responding to the actions of the students?

3

Given what happens in the episode (the ideas/doubts expressed by the children, the interactions that take place, possible difficulties...) what do you think you could do as a teacher to improve the inclusive participation of all students?

4

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

R2AS02 - INCLUSIVE IMPLEMENTATION - CANVAS #4

TABLE FOR FINAL ANALYSIS

#R2AS02

FINAL REFLECTION (INDIVIDUAL)

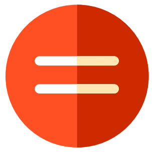
What do you consider the most relevant topics or ideas discussed in the previous analysis? What do you think you should need to learn to better promote inclusion in the classroom? Write the most relevant items.

1

FINAL INTERPRETATION (INDIVIDUAL)

Which are the main issues regarding the promotion of inclusion in STE(A)M activities which have been highlighted from the interpretation of the students' actions? (possible issues in students regarding inclusion)

2



Which main aspects can be highlighted regarding the role of the teacher responding to the actions of the students? (in terms of possible issues in teaching practices regarding the promotion of inclusion)

3

Given what has been discussed previously (the ideas/doubts expressed by the children, the interactions that take place, possible difficulties...) what do you think you could do as a teacher to improve the inclusive participation of all students? (in terms of teaching practices that can be developed for the promotion of inclusion)

4

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

DATA PRIVACY, ETHICS & SECURITY

#R2AS03

What is it?

The activity sheets #R2AS03 and #R2AS04 aim to address general themes in order to engage teachers and students in a discussion on the different ethical and security issues related to data sharing and promotion.

Modalities

Going further activity.

Duration

1h30

Level of difficulty

Basic

Material

- No specific material needed
- Access to the internet will be a plus to check the resources proposed in the activity sheet

LEARNING OBJECTIVES

- Basics of data privacy and exposure of private data when acting on the digital ecosystem

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

STEP 1 - ORIENTATE

20 min.

It is possible that you have heard that Google spy on us. Have you ever searched how can it be? What have you found? Some years ago, research revealed that Google was spying through non-encrypted routers. You can find some more information here and here!

DISCUSS WITH YOUR PEERS WHAT YOU THINK

Do you know what is a router for?
Where have you seen a router?
Have you ever heard what is encryption?
Can you imagine what is it for?
Can you imagine what kind of information can be accessible in WiFis detected?

You can perform a small brainstorming exercise with your peers.

STEP 2 - CONCEPTUALISE

15 min.

As you can imagine, all information that is shared through the router can be in the form of an email, social networks, or many other things you do when you use the Internet.

THINK ABOUT IT A BIT

What do you use the internet for? What searches do you do? What pages do you visit?
Have you noticed that on many pages it appears a request for you to accept cookies? Do you usually accept cookies? Why?
Can you imagine what type of information is shared (and can be stored about you), when surfing the internet?
Do you "like" that all this information about you can be stored? Can you imagine the potential risks of storing it?
Have you read in detail the privacy policy of data protection of some of the websites you usually use?
Which other Internet resources have you used in the previous Let's STEAM modules? Are there some resources which you would like to review its privacy policy?

Discuss with your peers and try to sot the type of information that can be stored about you when you use the Internet, based on the potential risk of it.

STEP 3 - INVESTIGATE

How do you know which type of information is shared about you and what risk might lead to? To have a first approach, try to search your full name on the Internet and see what results come up (include search in online gaming and social media accounts).

Do you think that the results reflect who you are and/or what you do? How? Try to search for one or two close friends.
Do you think that the results reflect who they are and/or what they do? How?
Have you contributed to providing more information about them on the Internet? How?
Which information do you think your friends have shared about you?

Discuss these issues with your peers. You can update the list of information and the risks you previously identified with new topics if is necessary. With a **group of 3-4 peers**, try to identify **10 best practices or actions to reduce the risks of sharing different types of information** and keep privacy in personal data. *For example: Which actions could we undertake to keep our information private? (Is it better to have a public profile on social media or a private profile? // download any app from the AppStore // navigate on the internet logged in on your Google account...).* You can relate to the list you previously did and define different practices according to the level of sensitivity of the information.

Afterwards, your group will merge with another small group. **Read the best practices designed by the other members of the group**. Try to merge and make a common list of 10 best practices by:

Identifying which practices/actions are similar between groups and merge them.
Discussing the relevance of the different practices/actions trying to order them from more relevant to less relevant.

Additionally, you can repeat the same merging with another group work so you can finally have a common list for the big group of trainees.

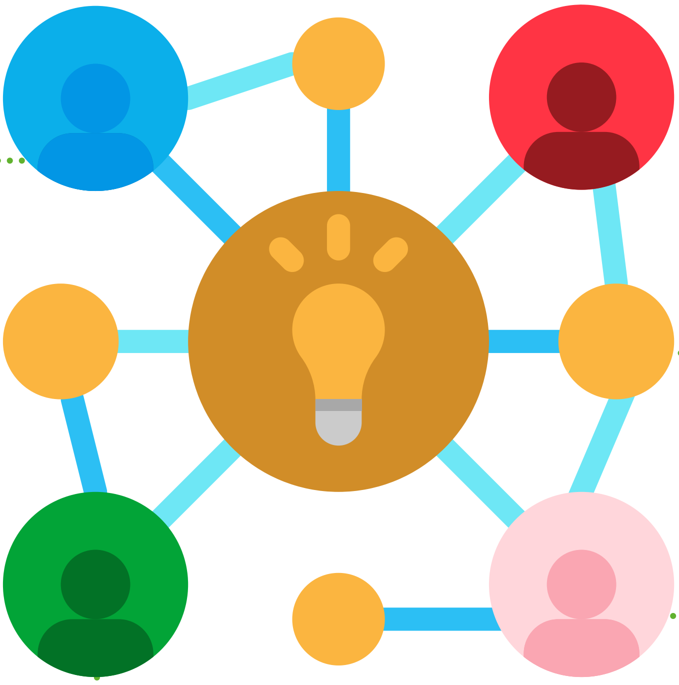
STEP 4 - CONCLUDE

Discuss with the whole group the most important actions/best practices to reduce the exposure of private data.

WILL YOU TRY TO IMPLEMENT THEM FROM NOW ON?

PROMOTING AND SHARING

#R2AS04



What is it?

The activity sheets #R2AS03 and #R2AS04 aim to address general themes in order to engage teachers and students in a discussion on the different ethical and security issues related to data sharing and promotion.

Modalities

Going further activity.

Duration

1h

Level of difficulty

Basic

Material

- No specific material needed
- Access to the internet will be a plus to check the resources proposed in the activity sheet

LEARNING OBJECTIVES

- Imagine the positive implications of sharing information on the Internet
- Consider to which extent they would share activities or products (such as pictures, videos, or images) that they have made and what prevents them from doing it
- Get familiarized with the framework of Creative Commons

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

PROMOTING AND SHARING

STEP 1 - ORIENTATE

10 min.

Following the discussion about sharing different personal information about you and your friends and peers on the Internet, **try to think now about the positive uses** of sharing information can have.

STEP 2 - CONCEPTUALISE

15 min.

1

So sharing information on the Internet can indeed have some positive aspects. Based on what you have previously discussed, **will you be keen on simply sharing activities** or other products (such as pictures, videos, or images) that you have made? **Will you share the Let's STEAM activities you adapted to promote inclusion and equity? What can prevent you/others from doing it?**

2

Discuss with your peers how you would feel if these pictures/videos/images are shared again on the Internet from people whom you do not know without your permission? Have you used images, videos, music, or other resources from other people in previous works? Did you know if you could use those resources? How?

3

Consider different situations:

- **A well-known T-shirt manufactured logo is used on T-shirts produced in another country. Who should get the profits for the sales of the T-shirts?**
- **Some software is loaded on a computer at a large company. Employees are downloading the software for use on their home computers. Should someone pay? If so, who? How much? Why?**
- **A student in the class copies this handout and uses it in her business class at another class. Is that a violation of the copyright of these materials?**
- **A television program uses the same plot and characters as another show. Should the program obtain permission to use the copyrighted elements of the original show? Why/why not?**
- **A company makes copies of a famous painting. The company sells copies. Who should pay for the right to copy these paintings? Why?**
- **A teacher uses an article from the newspaper in her class. She copies the article and gives it to her students. Have intellectual property rights been violated? If so, whose? If not, why not?**
- **An architect copies the design of a building and sells it to a client. Whose intellectual property rights have been violated? What should be done? Who should pay?**

STEP 3 - INVESTIGATE

25 min.

If you like that, when you share documents, images, videos, or other resources created by you on the Internet people give you credit as the author, **there are some tools you can benefit from**.
One is the use of **Creative Commons licenses**, which are tools that offer the possibility of a simple and standardized way to grant copyright permissions on their creative works. In this part, we ask you to **have a look at the types of Creative Commons licenses** and think about which license would you use if you were to share documents, images, videos, or other resources.

Check here!
https://creativecommons.org/licenses/

If there is too much information on this page, try with the **simpler version** to choose which type of license would be more useful for you. Try to put the type of license you have chosen on the document, image, video, or other resources you would like to share.
Resource: https://chooser-beta.creativecommons.org

STEP 4 - CONCLUDE

10 min.

Share with your peers **which type of license you have chosen**. Explain to them why you have selected this license and listen to their choices. Refine the type of license you would use in case you need it. To end with, discuss with your peers and consider other possible best practices that can be implemented to **give credit and respect the ownership of the shared materials**, as:

- **Share a link to a mentioned work instead of making copies of it (e.g., via open libraries, website, or by linking to another legitimate depository or website).**
- **Use caution in downloading digital material from the Internet. Some copyrighted works may have been posted to the Internet without the authorization of the copyright holder (creator).**
- **In your creations, take precautions to protect the copyrighted work from broader distribution (e.g., by streaming rather than posting a video; by posting on a password-protected site).**
- **When you do a project, credit all sources, display the copyright notice, and indicate which materials have been used with permission. Sometimes citing the material does not grant permission to use a copyrighted work.**

Consider the use of creative commons repositories, as:

HINTS FOR THE TRAINERS/TEACHERS

You are a Let's STEAM trainer and/or you have completed the Let's STEAM programme and want to train your students on inclusion and equity issues

Reminder: Feel free to reuse the activity sheets and templates presented in this section in your classroom and share them with your students! You are free to print, reproduce, modify, reuse and draw inspiration from all the resources in this manual without constraint. Our content has been developed entirely under a Creative Commons license.

HINTS FOR TRAINERS/TEACHERS

STEP 1 - INSPIRE

Hints for the trainers/teachers: Ensure that everyone presents themselves and identify if some information is missing. Invite trainees to talk about their students' traits, background and relationship with STEM (students with special needs, girls, racial minorities, and low socioeconomic background). Also, ask trainees if there are special policies or practices in their schools to promote equity and inclusion.

STEP 2 - CONTEXTUALISE & EMPHATHISE

Hints for the trainers/teachers: Try to encourage trainees' brainstorming while conducting the first step. There are no right or wrong answers, just different experiences and it is important that they are shared. If trainees get stuck or perform a relatively superficial analysis, try **to engage them in a deeper analysis by prompting the list of questions given in the associated checklist** without providing clues and orientations that will be given in the analyse section.

STEP 3 - ANALYSE

Hints for the trainers/teachers: This is individual work. Try to encourage your trainees to brainstorm as many potential issues as possible, trying to focus on the ones more closely related to equity and inclusion issues. Write down in a document those potential issues. Prior to that activity, you can suggest they read again the canvas of their group.

STEP 4 - DESIGN & IDEATE

Hints for the trainers/teachers: Trainees are expected to design strategies to make the activities more inclusive and equitable. If the group feels stuck, some strategies can be suggested as provided in the additional resources and strategies promoted within the checklist on inclusive design.

CONCLUDE

Hints for the trainers/teachers: The aim of this part is that one group can explain to the other members what they have designed and, at the same time, all trainees can have the opportunity to listen to what other groups have thought. To manage this exchange, one possibility is that one member of the group receives trainees from other groups, while the rest of the members of the initial group can circulate from one group to another ones. Members of the team who listen to the proposal are invited to give feedback. This is the easiest version to be carried out if the training is conducted on-site. If the training is online, you might consider setting up a collaborative space. In this space, for example, each group can upload a video where they explain to others what they have done and why, and invite the rest of the members to provide feedback.

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

CHECKLIST - BEST PRACTICES

Have you considered how students with special needs might face difficulties in the accessibility to technology? You could possibly address some of those issues by:

- Varying the methods for response and navigation.
- Considering different platforms or programming languages for the same activity, according to their level of difficulty.
- Have high expectations of all your students. Research shows that students respond better when they feel that their teacher has faith in their abilities and is not focusing on their inabilities

Have you considered how students with special needs might have difficulties in understanding the purpose and what they are expected to do in the educational activities? You could possibly address some of those issues by:

- Considering a general routine that will be used in all activities.
- Providing clues, help when they are needed (not anticipating their potential issues). Adapt the designed scaffolding to the development of the activity.
- Analysing the level of difficulty of each of the tasks within the designed activities and order them from easy to difficult. Avoiding big jumps in the sequence.
- Considering optional repetition or skipping in the development of each task to achieve the demand.
- Promoting students' customization of their preferred communication.
- Expressing the same in a multimodal way (i.e. using text, images, videos).
- Considering automated speech-to-text software. Using captions to images and subtitles to videos.
- Providing equivalent alternatives and different learning paths. Considering different levels of achievement in the same activity focusing on each students' successes, but not forcing all students to succeed in the same level of difficulty in the demand.
- Considering different and additional "aids" to build an adaptive scaffolding (e.g. prompts for students, hints, additional materials, mentor texts, sample solution, pictographic hints, possible peer support...), and/or graphic organizers (concept maps, etc.).
- Considering different ways of students' participation: independent work, dyads, small groups... and how these collaborations will be managed to promote inclusion.
- Providing opportunities to show what they have learned.
- Providing opportunities to interact with peers, and establish rules for it. Being careful of the language used.
- Clarifying vocabulary and symbols. Illustrating difficult terms, provide visual hints (i.e. highlight patterns, main ideas, etc.).

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

Have you considered how women, racial minorities, and students from low socioeconomic backgrounds might feel that STE(A)M activities are “not for them”?

There is a biased representation of women, racial minorities, and students from low socioeconomic backgrounds in the image of what is a STEM person, which results in a stereotype threat. The white-male centralism of the stereotype of STEM people is also translated in the design of educational activities. You could possibly address some of those issues by:

- Considering using diverse cultural data sets.
- Equilibrating the representation of cultural/racial diversity in examples (i.e. names used, illustrations... etc.).
- Equilibrating the presence of girls/women.
- Equilibrating the presence of cultural groups.
- Using neutral and non-sex language when addressing students and when referring to STEM careers/activities.
- Using gender-neutral language to describe groups of students (instead of ‘Now, guys’ consider expressions like, ‘Now, everybody’).
- Equilibrating the role of students within the activity.
- Ensuring that everyone has the same opportunities to participate by providing different and changing roles in the group work, for example.
- Allowing exploring aspects of their own culture and/or gender identity concerning computing. For example, give students creative freedom to express culture and/or identity affiliations.
- Assessing and identifying integrated gender, racial, and/or cultural stereotypes and biases (in own teaching and in the way students’ behave) and creating teachable moments by challenging these undermining stereotypes.
- Incorporating the gender perspective (in a broad sense, either in the language used and the role-model references). Increase the diversity of role models used by showing how women, people from diverse socio-cultural backgrounds have contributed to STEM (avoid showing male professional STEM and STEM as a masculine discipline).
- Giving time to students to think before allowing them to answer a question raised to the whole group. Choose different students to answer.
- Identifying and celebrating achievements for all students valuing their effort and strategies.
- Creating a common safe space. Building a “judgment-free zone”. Students at disadvantage have fear participating in public spaces due to peer judgment.
- Providing opportunities for all to participate (by first thinking/writing, sharing with peers, etc.).
- Promoting collaborative learning rather than competitive learning. Provide constructive and formative feedback at all times.

Have you considered how students from diverse cultural backgrounds may have issues understanding the main language of the lesson?

- Consider using different languages: the dominant language of the school and their native language.

Have you considered how students from low socioeconomic backgrounds will have difficulties accessing the resources?

- Design activities with low-cost and accessible materials.
- Consider other materials to use.

Have you considered how to improve the design of your STE(A)M activities so they can be more aligned with the universal design for everyone?

Not all students will engage in the same way in robotics & computational thinking activities. You could possibly address some of those issues by:

- Promote different opportunities for engagement. Problematise the activity (is not about doing a task, but solving a particular problem).
- Possibility to adapt the activity into their own interests (setting the question to explore) and provide relevance, value, and authenticity. The teacher needs to explore what relevant questions students will be more inclined to answer. Provide equivalent alternatives and different learning paths.
- There is not only one solution but different and valid solutions. As well, consider possible different ways to carry out the activity. The teacher needs to explore how to make students' choose different paths in the same activity and help them to set appropriate and manageable goals to promote students' choice and autonomy.
- Students will need to know at all times what is expected from them, and what have they done. At different times within the same activity, the teacher needs to remember their students the aims of the activity and provide constructive feedback. Especially, focus on mastery-oriented feedback (praising achievements).
- Promote students' self-reflection about their successes throughout the activity. — Allow students to express what they have learned in different ways (e.g. presentation, video essay, drawing a comic... etc.).
- Allow revising and resubmitting assignments/tasks. - Not all students will express themselves in the same way within an activity.
- Consider multiple representations of information. Offer alternative means of expression.
- Not all students know the risks of using digital technologies. Consider the introduction of specific topics: Copyright law, Fair Use Act and Creative Commons matter (give credit to the original source); self-image on the Internet and related risks.

STEP 1 - ORIENTATE

Hints for the trainers/teachers: Trainers need to encourage the groups to revise their activities and try to improve them by considering the suggestions of other working groups. If there have not been any, trainers can also suggest other improvements listed on previous parts. Finally, trainees are invited to outline a strategy to assess the impact of their activity. As a trainer, try to make them consider all the possibilities (inclusion of students with special needs, increase the equity in terms of gender, racial minorities and students from low socioeconomic backgrounds, universal design).

Try to suggest some indicators for those groups who are stuck. You can have great examples here:

- **For students with special needs**: http://inclusionworks.org/sites/default/files/QualityIndicatorsGuidebook.pdf
- **For students with special needs and other students with fewer opportunities**: https://www.britishcouncil.es/sites/default/files/british-council-guidelines-for-inclusion-and-diversity-in-schools.pdf
- **Inclusive classrooms**: http://www.csie.org.uk/resources/inclusion-index-explained.shtml

STEP 2 - INVESTIGATE

Hints for the trainers/teachers: This activity can be carried out in two different modalities. Choose one of those depending on the possibilities of the training.

Modality A - Real implementation

The first modality, and the most desirable one, is that trainees use their revised Let's STEAM activities (resulting from the previous training activity "Inclusive Design") and implement them with their students. Each teacher is invited to collect the evidence agreed upon and fill in the table of initial analysis (page 106 in this coursebook) after each implementation.

After all the Lets' STEAM activities are implemented, each teacher will share all the information gathered with the rest of the group and their tables of initial analysis. While he/she is explaining their implementation, the rest of the trainees will also fill in a table of initial analyses to structure their interpretation of the implementation of the activities.

The aim of this constructive analysis is that trainees progressively become aware of what can be improved in the implementation of the activities in terms of educational aims and promoting the engagement of all students (inclusion).

Each time one teacher present the results of their implementation, the rest of the trainees are expected to use the **table for initial analysis (page 106 in this coursebook)** to structure the feedback. They are encouraged to provide their feedback in a constructive way and direct the dialogue towards the identification of main issues regarding the **promotion of equity** in the Let's STEAM (e.g. *What do you think has happened? Which evidence do you have? How can we interpret it in terms of the promotion of inclusiveness? Can you think of a similar situation in your teaching experience? What did you do then*?) activities and development of **strategies to overcome these issues** (e.g. *What do you think it could be done to change this situation? What do you think we can learn from this situation? What could be applied to other learning situations*?).

Additionally, trainees can have the guidelines for the promotion of an inclusive implementation of Let's STEAM activities as an inspiration in this analysis. Ideally, all trainees in the group should have the opportunity to receive constructive feedback on their implementations.

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

STEP 2 - INVESTIGATE

Modality B - Micro-teaching
If trainees cannot implement the activity with their students, the trainees would implement a Let's STEAM activity with the colleagues in the group.

The leading trainee will conduct the activity as if they were in their regular classroom and the other trainees will act as students. Note that is important that the trainees know the type of students and try to reproduce potential issues that can occur in a regular lesson.

The implementation would not be longer than 7-12 minutes. After the implementation, all trainees of the group use the Table of initial analysis to provide constructive feedback to the leading teacher. The aim of this constructive analysis is that trainees progressively become aware of what can be improved in the implementation of the activities in terms of educational aims and promoting the engagement of all students (inclusion).

Each time one teacher conducts a micro-teaching episode, the rest of the trainees are expected to use the **table for initial analysis (page 106 in this coursebook)** to structure the feedback. They are encouraged to provide their feedback in a constructive way and direct the dialogue towards the identification of main issues regarding the **promotion of equity** in the Let's STEAM (*e.g. What do you think has happened? Which evidence do you have? How can we interpret it in terms of the promotion of inclusiveness? Can you think of a similar situation in your teaching experience? What did you do then?*) activities and development of **strategies to overcome these issues** (*e.g. What do you think it could be done to change this situation? What do you think we can learn from this situation? What could be applied to other learning situations?*).

Additionally, trainees can have the guidelines for the promotion of an inclusive implementation of Let's STEAM activities as an inspiration in this analysis. This cycle can be carried out as many times as the group likes. Ideally, all trainees in the group should have the opportunity to act as leading trainees.

STEP 3 - CONCLUDE

Hints for the trainers/teachers: The aim of this part is that one group can explain to the other members what they have designed and, at the same time, all trainees can have the opportunity to listen to what other groups have thought. To manage this exchange, one possibility is that one member of the group receives trainees from other groups, while the rest of the members of the initial group can circulate from one group to another ones. Members of the team who listen to the proposal are invited to give feedback. This is the easiest version to be carried out if the training is conducted on-site. If the training is online, you might consider setting up a collaborative space. In this space, for example, each group can upload a video where they explain to others what they have done and why, and invite the rest of the members to provide feedback.

STEP 1 - ORIENTATE

Hints for the trainers/teachers: This activity can be carried out both in students and teachers. Instructions for trainers in this part will consider both scenarios. Thus, in this part trainees can be equally referred to as participant teachers or students and the trainer may refer to as the teacher trainer or the teacher itself when the activity is conducted in a classroom.

At the very beginning, the context of 'Google spying on us' is presented. Ask trainees if they have heard these statements before, and in which contexts, and if they have evidence to believe in them. Afterwards, present the case of Google spying through the non-encrypted routers: https://www.wired.com/2012/05/google-wifi-fcc-investigation/https://www.theguardian.com/technology/2010/may/15/google-admits-storing-private-data

You can ask trainees if they understand the content and its implications by asking them what they know about routers and encryption. Provide appropriate explanations in case needed, according to the knowledge and age of trainees (teachers or students).

Afterwards, ask trainees (teachers or students) if they are aware of what kind of information can be accessed from WiFis which are detected (as done in the previous Let's STEAM activities). You can invite trainees to perform a small brainstorm on the type of information they share online. (*optional: Discuss with trainees how they could prevent someone to have access to the information you transfer through your WiFi and briefly introduce the different encryption protocols. Try to foresee if the WiFi's detected with previous Let's STEAM activities*.)

STEP 2 - CONCEPTUALISE

Hints for the trainers/teachers: The trainer introduces to trainees that the information shared through routers can be gathered from the activity of using the Internet for different purposes.

The trainer can ask trainees different questions to trigger the discussion about potential risks of using the Internet, with:

- **What do you use the internet for? What searches do you do? What pages do you visit?**
- **Do you usually accept cookies? why? (if it is needed, briefly explain to trainees what are cookies and their uses).**
- **Do you know what type of information is shared (and can be stored about you), when surfing the internet? (examples: Location, Date // Year of birth, Mobile number, email address, Gender, Personal information...)**
- **How could you know which data about you have been stored? If trainees are teachers, you can also shift the discussion towards the kind of data that their students share on the Internet.**

Trainers are also encouraged to review the privacy policy of Internet resources that they have used in previous Let's STEAM activities, for example, Scratch. Try to direct the discussion towards if they knew which type of information was stored and the purposes of the storage of this information.

As well, if they agree or not with the purposes of the use, how it makes them feel and which actions they might consider undertaking in consequence. After some minutes, try to make a list with the trainees sorting the types of information that can be stored when using the Internet based on their potential risk (harm/ sensibility).

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

STEP 3 - INVESTIGATE

Hints for the trainers/teachers: Trainees will be asked to search their full names on the Internet. See what results come up. Ask trainees: **Do you think that the results reflect who you are and/or what you do? How? Which image of you that information is projecting?** Invite trainees to search for one or two close friends (or peers who they already know). Discuss with them: **Do you think that the results reflect who they are and/or what they do? Have you contributed to providing more information about them on the Internet? How? Which information do you think your friends share about you?** Trainees can update the list of types of information and potential risks if it is needed. Encourage trainees to evaluate again the impact of sharing sensitive information that can easily put their privacy and security at risk if it is shared by mistake, thoughtlessness, or misleading prompts. Continue encouraging the discussion with new questions: **Do you think that sharing information is good? Are there any examples of positive sharing online? How about negative ones? Do you think you have given consent to share this information? Have you ever received a spam email or call without knowing how they had your data?**

The aim is that trainees become more aware of the potential risks of sharing information, especially sensible information. You can also introduce broader challenges and opportunities at a societal level concerning privacy and security in the Internet-of-Things era, the commercialization of data, the needs for top-down and bottom-up regulation and standardization, etc. will be considered. The internet knows everything and never forgets, which also there is a need to have rules for a right to oblivion. Content posted online can last forever and could be shared publicly by anyone.

Being aware of the super-exposure of private data in online environments, trainees will be invited to identify best practices to reduce the risks of sharing information and keep privacy in personal data shared online. For example: Which actions could we undertake to keep our information private? (Is it better to have a public profile on social media or a private profile? // download any app from the AppStore // navigate on the internet logged in on your Google account...). To do so, trainees will work in groups of 3 or 4 people identifying 10 different actions. As a trainer, try to foster the dialogue prompting different situations, such as:

- **You want to add your close friend to a group chat. - You take a funny picture of your neighbour's dog and want to post it online.**
- **You've just got a new boyfriend/girlfriend and want to change your relationship status.**
- **You see someone asleep on the bus and want to take a photo and share it online.**
- **You want to share your location and tag your friends.**
- **You find an old photo of you and your sibling and want to share it online.**
- **You want to wish your friend a happy birthday by posting on their social media account.**
- **You're sent a photo of yourself and a friend and you look really good in it. Your friend doesn't but you want to post it online anyway.**
- **You're on your holidays with your family and want to share a photo and tag them in it along with sharing the place you are staying.**

Trainees can search on the Internet for other examples if needed it.

Afterwards, the group of 3 or 4 trainees will merge with another small group, trying to: identify which guidelines are similar between groups and merge them; discuss the relevance of the different guidelines trying to order them from more relevant to less relevant. This procedure can be repeated at different times to reach a consensus with the whole group.

STEP 4 - CONCLUDE

Hints for the trainers/teachers: Discuss with the whole group the most important actions/best practices to reduce the exposure of private data. You can additionally ask trainees to elaborate on infographics with these guidelines to distribute to other peers from other groups.

STEP 1 - ORIENTATE

Hints for the trainers/teachers: To start with this activity, try to engage trainees in a discussion trying to imagine the positive implications of sharing information on the Internet. You can foster the discussion by giving some ideas on crowdfunding campaigns, viral challenges, being recognized and promote your work.

STEP 2 - CONCEPTUALISE

Hints for the trainers/teachers: Engage trainees in a brief discussion to consider to which extent they would share activities or products (such as pictures, videos, or images) that they have made and what prevents them from doing it.

If your trainees are teachers, ask them what they do with the Let's STEAM activities which they adapted to their students to promote their inclusiveness: if they will share them with other colleagues and/on the Internet, or just store them in their computer, and why. The aim of this discussion is to evidence that sharing might make one feel a lack of control of who has this information and for which purposes it will be used/shared again.

After this discussion, ask how they would feel about being shared activities that they have designed on the internet without permission. Ask trainees if they have used images, videos, music, or other resources with their activities/work and, if they had if they knew that these images could be used.

The aim of this discussion is to make trainees realize that it is good to share some information on the Internet, but everyone has the right to intellectual property in the products that are created.

Additionally, other examples can be used to discuss the right to intellectual property - These examples have been extracted from *https://americanenglish.state.gov/files/ae/resource_files/business_ethics_ch7.pdf*

- A well-known T-shirt manufactured logo is used on T-shirts produced in another country. Who should get the profits for the sales of the T-shirts?
- Some software is loaded on a computer at a large company. Employees are downloading the software for use on their home computers. Should someone pay? If so, who? How much? Why?
- A television program uses the same plot and characters as another show. Should the program obtain permission to use the copyrighted elements of the original show? Why/why not?
- A student in the class copies this handout and uses it in her business class at another class. Is that a violation of the copyright of these materials? - A teacher uses an article from the newspaper in her class. She copies the article and gives it to her students. Have intellectual property rights been violated? If so, whose? If not, why not?
- A company makes copies of a famous painting. The company sells copies. Who should pay for the right to copy these paintings? Why?
- An architect copies the design of a building and sells it to a client. Whose intellectual property rights have been violated? What should be done? Who should pay?

At the end of this part, trainees need to be aware of the needs of setting an intellectual property and respect it.

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

STEP 3 - INVESTIGATE

Hints for the trainers/teachers: The aim of this part is that trainees get familiarized with the framework of Creative Commons and the type of licenses that are offered and try and define a license that they would use if they were to share documents, images, videos, or other resources created by them on the Internet.

STEP 4 - CONCLUDE

Hints for the trainers/teachers: The objective of this final part is, on one hand, of the direct advantages of mutual benefit of resources that are made online and public, but also on the other on the importance of ethical use and responsibility by retaining and asserting copyright and authorship. To this end, a final discussion will be promoted in which trainees will share the types of licenses they have chosen to share their created resources.

The trainer can also consider introducing in the final discussion other patents that can be used, as described here. It is relevant to discuss with trainees the difference in terms of sharing and rights of use, highlighting the issue between very restrictive patents (which guarantee the rights of use and exploitation but impede other users to benefit from, such as drugs and vaccines), and patents as Creative Commons licenses, which allow the users to benefit from other's creations and develop their own ones.

Additionally, the trainer can engage trainees in the discussion of other best practices (suggested above) to ensure the ethical use of the information. Trainees can be invited to explore different repositories in which resources without royalties are being shared, as well as explore how the authors of these resources can receive credit for their work.

Reminder: Feel free to reuse the activity sheets and templates presented in this section in your classroom and share them with your students! You are free to print, reproduce, modify, reuse and draw inspiration from all the resources in this manual without constraint. Our content has been developed entirely under a Creative Commons license.

ACTIVITY SHEETS & TEMPLATES

REPLICATE IBL IN YOUR CLASSROOM - GUIDELINES & TEMPLATE

Authors: Margarida Romero, Despoina Schina, Stéphane Vassort

In order to create your course resources using the training programme of Let's STEAM, the proposed inquiry approach has been translated into an open and directly usable template, divided into 3 parts i.e. how to collect data, how to show these data and how to analyse them to learn from the experimentation. The following template gives you hints and information on how to use it to produce your own lesson plans.

Through this phase, you will search into some documentation and lead experimentations with programmable boards.

We are fully aware that starting from scratch on the proposed template might be a challenging process. It is not always obvious to exemplify a pedagogical concept or topic based on programming practices, especially for beginners regarding the use of microcontrollers and boards. This should not block you from going further in the development of meaningful coding activities, and this is why we have prepared examples gathered in this coursebook regarding what can be the deployment of our IBL template in the framework of the classroom. You can also feel free to use the following online resources as content for inspiration such as:

www.hackster.io/projects

www.microsoft.com/en-us/makecode/resources

www.instructables.com/projects/

makezine.com/projects/

hackaday.io/projects

These makers and developers' communities are sharing thousands of projects and ideas that can be adapted to a fruitful learning activity thanks to your pedagogical knowledge!

CONCRETELY, HOW IS THE IBL TEMPLATE STRUCTURED?

You can find in the next pages an open and directly usable template, divided into **4 parts:**

Step 1 - Present the project as a whole - 1 page

Describe the project you want to launch and reflect on the first main questions to be asked regarding inclusiveness before developing the content of your activity

Step 2 - Collect data thanks to the board and its embedded sensors - 2 pages

At this stage, you are required to find a programming solution to collect your data, identify which sensors to be used and how to program them on MakeCode for the platform to communicate with your board.

Step 3 - Display the data to get the needed information - 2 pages

At this stage, you are required to find a programming solution to display your data, enabling, now you have asked a sensor to obtain information, to make this information known to the user.

Step 4 - Analyse the data and learn from them - 2 pages

Now we are able to display data instantly, we need to analyze them to perform monitoring of our information (for instance, monitoring of temperature, of alerts, motion, frequency ...). This stage is made for enabling this analysis on the editor.

This division has been selected to ensure that your project is **readable** and **well defined**: from **data collection** to **display** to **exploitation**. You can **change or add as many parts as you wish**, as soon as you respect the inquiry approach steps in each of them. We consider that **3/4 parts** are a good ratio. Here is the definition of the expected contents for each of the steps of the inquiry-based learning approach:

Orientation	*Arouse astonishment and curiosity by proposing a triggering situation* *Define what is the problem to be solved*
Conceptualisation	*Structure the questioning, organise the ideas, clarify the vocabulary* *Formulate a hypothesis to answer the problem*
Investigation	*Proposes programming activities considering the experiments to be made* *Imagine how to verify the hypothesis and test them*
Debrief	*Identify the knowledge mobilized during this phase* *Think about your classroom and identify possible learning* *Add references issues that may come up*

In addition, you will find at the end of this coursebook a **list of 8 projects' ideas** that you can get inspired from, use, develop or modify:

- **Idea 1: How to make the invisible visible?** Reproduce the natural environment of frogs to ensure their survival (complete example)
- **Idea 2: Preserve biodiversity.** Monitor the number of plant species in your neighbourhood. Explore the streets and parks in your neighbourhood to find out more about the ecosystem and use technology to make this process easier! Use the STM32 card to record your findings!
- **Idea 3: Temperature control in the classroom.** It is too hot in the classroom. When students come in, they know to close the blinds, but during break time, the classroom gets really hot. How can we create a more autonomous system through programming?
- **Idea 4: Build a welcoming classroom.** Identify the particular light intensity needs in your classroom to perform a specific activity.
- **Idea 5: Your ideal (and sustainable) home.** Dream about where you would like to live, what your ideal home would be like and how this ideal home could be more sustainable.
- **Idea 6: Washing hands.** We need to ensure that children wash their hands when they come back from the playground. Although new routines have been put in place to ensure that all children wash their hands, we are not sure that they do it well enough. How can programming help us stick to the barrier actions?
- **Idea 7: Reasonable heating use.** Identify the optimum position for using heating appliances at given times to save electricity.
- **Idea 8: Music: Can you play what you hear?** Have you ever wished you could play a song on the piano just by listening to it?

These are proposed by the Let's STEAM consortium members. Feel free to contact each project responsible to co-create with us a solution.

Enjoy programming in the Let's STEAM way! Unleash your creativity and get started!

Step 1 - Present the project as a whole

We invite you through this template to get creative while getting technical support in designing a unique and inclusive project! You are free to develop your own solution or to be inspired by solutions proposals. In the end, depending on the path you choose, your solution will be unique!

Describe your project

Name your project: ____

Short introduction of what your project is about, the problem tackled behind, the pedagogical objectives

____

____

Reflect on equity and inclusiveness

ASPIRATIONS & MOTIVATIONS

How do you feel when doing STEM? What motivates you in STEM? What motivates your students? Are all your students motivated by the same? What would they like to do?

____

____

ISSUES AND BARRIERS

What worries your students? What frustrations do they have? Are there any differences that make them be at a disadvantage to other students? And regarding robotic and digital in STEM activities?

____

____

KEYWORDS

Indicate 3 or more keywords that describe the reality of your students regarding STEM/STEAM activities

____

- Review the table of potential use cases available at the end of this book for inspirations
- Review the **"Resources on inclusive education - Activity sheet 1 - R2AS1"** for reflecting on inclusiveness.
- Use the **Canva 1 - Emphatise (page 102)** for performing the activity.

Step 2 - Collect data thanks to the board and its embedded sensors - 1/2

At this stage, you are required to find a programming solution to collect your data, identify which sensors to be used and how to program them on MakeCode for the platform to communicate with your board.

ORIENTATION

Define what is the problem to be solved, what are data to be collected, what are the learning objectives behind the programming topic?

____

____

____

____

____

____

____

CONCEPTUALISATION

Formulate a hypothesis to answer the given problem regarding data collection

____

____

____

____

____

____

____

INVESTIGATION

Describe the steps you need to collect the data that will be necessary for your project

For helping you in your developments and choices, check the resources available in **PART II - PROGRAMMING EASILY THANKS TO LET'S STEAM ACTIVITY SHEETS.**

Provide screenshots of the MakeCode platform and of your board

DEBRIEF

Identify the knowledge mobilized during this phase, think about your classroom and identify possible learning, add references issues that may come up

INCLUSIVENESS

At this stage, you start to have a clear idea of how the project and the activity will be performed! But have you thought of the inclusiveness and equity requirements while designing it!? Let's check this out by answering the **Canva #2 - Checklist available page 103**.

At this stage, you are required to find a programming solution to display your data, enabling, now you have asked a sensor to obtain information, to make this information known to the user.

ORIENTATION

Define what is the challenge in the display of the data you need? For you? For your classroom? For the user?

____

____

____

____

____

____

____

CONCEPTUALISATION

Formulate a hypothesis to answer the given problem regarding data display

____

____

____

____

____

____

____

INVESTIGATION

Describe the steps you need to display and show the data that will be necessary for your project

For helping you in your developments and choices, check the resources available in **PART II - PROGRAMMING EASILY THANKS TO LET'S STEAM ACTIVITY SHEETS.**

____

____

____

____

____

Provide screenshots of the MakeCode platform and of your board

DEBRIEF

Identify the knowledge mobilized during this phase, think about your classroom and identify possible learning, add references issues that may come up

____

____

INCLUSIVENESS

Getting a bit further in your project, let's perform an addition inclusiveness check! Data collection is a crucial step in terms of potential privacy and sharing issues! Reflect on this and on the whole process by answering the **Canva #2 - Checklist available page 103**.

Now we are able to display data instantly, we need to analyze them to perform monitoring of our information (for instance, monitoring of temperature, of alerts, motion, frequency ...). This stage is made for enabling this analysis on the editor.

ORIENTATION

Define what is the challenge in this step according to your project. What is your challenge in analysing and extracting the relevant information applied to your context?

____

____

____

____

____

____

____

CONCEPTUALISATION

Formulate a hypothesis to answer the given problem regarding data analysis

____

____

____

____

____

____

____

INVESTIGATION

Describe the steps you need to analyse and monitor the data that will be necessary for your project

For helping you in your developments and choices, check the resources available in **PART II - PROGRAMMING EASILY THANKS TO LET'S STEAM ACTIVITY SHEETS.**

Provide screen shots of the MakeCode platform and of your board

DEBRIEF

Identify the knowledge mobilized during this phase, think about your classroom and identify possible learning, add references issues that may come up

INCLUSIVENESS

At this point, it is relevant to reflect about the whole learning process proposed by your activity. You can go through the **Canva #2 - Checklist available page 103** one last time. When you have implemented the whole activity in your classroom, we encourage you to also fill in the **Table of Final Analysis available in this manual on page 107.**

INSPIRING EXAMPLES

8 PROJECT TOPICS FOR APPLYING THE IBL APPROACH

Authors: Mercè Gisbert Cervera, Carme Grimalt-Álvaro, Toon Callens, Maryna Rafalska, Margarida Romero, Despoina Schina, Cindy Smits, Lorena Tovar, Stéphane Vassort, Eleni Vordos

Idea 1: How to make the invisible visible? Reproduce the natural environment of frogs to ensure their survival (complete example)

Collect data thanks to the board and its embedded sensors

In order to reproduce the natural environment of frogs and ensure their survival, different parameters of their living environment must be taken into account. What information do we need to know in order to provide them with the most appropriate living environment? As the main parameter to be controlled to ensure the frog's survival is the temperature, and that it must be between 21 and 26 °C, the solution that seems to be the simplest is to use the temperature sensor integrated into the STM32 programming board.

Display the data to get the needed information

We could see in the previous part how to ask a sensor to obtain information. It would be useful now to be able to make this information known to the user. In order to inform the user of the measured temperature, the first solution that comes to mind is to use the LED display integrated into the board. Other solutions are possible such as a pointer and a dial like on a car speedometer.

Analyse the data and learn from them

We are able to display data instantly. In order to be able to analyze variations in climatic conditions and identify when the temperature level becomes critical for our frogs and the frequency of these alerts, it would be useful to be able to perform this monitoring over a long period of time. In order to be able to analyze the data from the temperature sensor over a long period of time, using spreadsheet software would be a simple solution. For this, it is necessary to be able to retrieve the data from the programmable board. The solution that I will implement will be to write via the serial port the data in CSV format (comma-separated value) which is exploitable by a spreadsheet program.

This project includes a final step: How to notify in case of emergency?

We are now able to measure and analyze the data from the sensors. It would be useful, in case of detection of an abnormal parameter, to be able to alert the user. There are two tasks to perform here: identify a temperature that is too high and alert the user. In order to automatically detect a too high temperature, we will use a conditional "IF" loop. Concerning the user alert, we can use the speaker embedded in the programmable board.

Idea 2: Preserve biodiversity. Monitor the number of plant species in your neighbourhood. Explore the streets and parks in your neighbourhood to find out more about the ecosystem and use technology to make this process easier! Use the STM32 card to record your findings!

Collect data thanks to the board and its embedded sensors

To ensure that the ecosystem in your local area is balanced and healthy, we propose you to monitor the diversity of plants species. How can we register different plant species? The parameter to be monitored is the number of species found in the ecosystem. The simplest solution is to use the STM32 board as a counter, to count the number of different plant species encountered in a walk in the streets, parks etc. of a neighbourhood.

Display the data to get the needed information

We could see in the previous part how to use an input device to obtain information. It would be useful now to be able to make this information known to the user. In order to inform the user of the number of species, we can add a screen.

Analyse the data and learn from them

The data collected can help us understand a lot about ecosystems and their characteristics. We can compare the biodiversity in neighbourhoods of the same or different cities, within the same or different country. If we collect and monitor these data over a long period of time and along different seasons, we can learn a lot about ecosystems, their characteristics and evolution.

In order to be able to draw conclusions about the biodiversity in our region and make comparisons, we need to share the data collected with our project partners in other cities and countries. We can organize the information collected in an excel spreadsheet and send it to our project partners. When information from all project partners is put together, we can draw very interesting conclusions about biodiversity and create our own map of biodiversity...

Idea 3: Temperature control in the classroom. It is too hot in the classroom. When students come in, they know to close the blinds, but during break time, the classroom gets really hot. How can we create a more autonomous system through programming?

Collect data thanks to the board and its embedded sensors

To make sure the blinds close when we need them to, we need to collect outside information. We need to gather if (and how strong) the sun is shining and we also need to know if the classroom is running too hot. To measure the outside brightness, we need a light sensor. To measure temperature, we need a temperature sensor. We need to think about where we place these sensors: a temperature sensor placed into the sun will give a temperature that is higher than the rest of the room. Assemble a breadboard with a light sensor and use the onboard temperature sensor to measure data. To achieve this, we need to program the board in MakeCode. To collect the data we will use the data logging from the MakeCode environment.

Display the data to get the needed information

After we have measured the light and temperature we need to use these data to keep a nice climate in the classroom. We will learn how to use sensor data and have multiple outputs react based on the data measured. Use the sensor data (from light and temperature sensors) to control the motor. When the temperature gets above a certain threshold, 22°C, the motor should automatically turn on to close the blinds. Likewise, when the brightness is too high, the blinds should also close. When the temperature drops back down and/or the outside light decreases, the blinds should automatically open again. We will also program a button to act as an override so that we can still manually open and close the blinds. We have to program one or multiple motors to act based on certain values the sensors pick up. We also need to program a button (or another kind of witch) to manually override the sensor so that we can close the blinds ourselves.

Analyse the data and learn from them

We now have automatically closing blinds. We have to monitor the system to see if it works in multiple different situations. This might be a process that takes time, as temperature and daylight greatly vary between seasons and we may for example not want the blinds to close in the darker months at all. To improve our system, we need to register the different situations in which our system works.

Idea 4: Build a welcoming classroom. Identify the particular light intensity needs in your classroom to perform a specific activity.

Collect data thanks to the board and its embedded sensors

In building a cosy classroom we need to ensure that we have the appropriate amount of light for the type of activity we need to perform. Which lighting needs do we have?
This activity could also be carried out with many variations, depending on the type of sensors available. For example, with temperature and CO2 sensors we could investigate how to keep a good air quality with a warm enough temperature or keep the classroom to an appropriate level of noise.

This project is focused on achieving good lighting for different types of activity (for example, an activity that needs concentration, and a general activity, such as listening to the teacher). The aim is that students identify that the lighting might need to be different according to the needs (both because of how it makes you feel and for visual health). So the main solution would be to use the light sensor.

Display the data to get the needed information

We need to show the data gathered about light intensity to study the different lighting requirements, or if we need to add an extra light (and where). Different ideas can be implemented, such as the use of a LED to show low levels of light. The optimal solution would be to transfer the data gathered to a computer so we could obtain a graph of the measure in real time.

Analyse the data and learn from them

As we are able to gather and display data, we can learn about different topics as:

- (Bio) Living beings interact with the environment and adapt themselves to external circumstances. A variation of this project could be to study how different plants are adapted to different light intensities, and which features make them to better capture the sun and where do they live so they are adapted to shadow and study these adaptations in relation to plant photosynthesis.
- (Phys) light travels in straight lines. The intensity of light declines as we are far from the light source (that is why in winter and at the beginning and at the end of the day there is less light intensity). We could also study how the light intensity declines (quadratic measure) to study which is the best high to install extra lights.

Data can be shown in real-time, but for longer data gathering it would also be good to download the gathered data in a CSV format and use a spreadsheet to analyse it.

Idea 5 - 1/2: Your ideal (and sustainable) home. Dream about where you would like to live, what your ideal home would be like and how this ideal home could be more sustainable.

Collect data thanks to the board and its embedded sensors - 1/2

Dream about what your ideal home would be. Which features? How you would distribute the space in it? And if you had to make it more energy-efficient, how would you do it?

As a first step, it would be better if students would draw their designs. Afterwards, a classroom discussion about their designs could be undertaken, putting a special emphasis on making them more energy efficient. So teachers/educators should guide students in the dialogue to identify different sources of energy (e.g. sun, heating systems...) and what they could do to not waste these energies. The aim of this dialogue would be to focus on the materials used to build the house, as they have a key role in saving energy. Then, students would be invited to reflect again on their own designs and think about which materials do help to save energy (i.e. isolate the heat) and which materials do not help to save energy (i.e. act as a heat conductor) and why students think they are thermal isolators or conductors. Some examples can be provided, such as glass, brick/chalk, metal, plastic, wood... In the end, the teacher would invite students to think about how they could better study if the material is isolator or conductor, introducing the need to use a data-gathering device.

Now that you have identified the relevance of the materials for building and you have to build the first design of your ideal home, we will test how these materials behave and which of them would make your home more energy-efficient. For this, we will need to try how different materials allow or not the transference of heat. Remember that a home in which there is a big heat transference cannot be considered energy efficient: you need to keep the inside isolated from the outside as possible.

Think about which evidences will you need to collect to study if a material is a heat conductor or an isolator. What would you measure? Which other conditions may affect the measure? How would you design an experiment so the heat conductor/isolating capacity of material could be tested?

It is important to guide the students so they can design a proper experiment to collect data about the isolating capacity of different materials provided. Other factors that affect the measure could also be considered here, such as material thickness, time of exposure to the heat, climate... The experiment could be carried out in two different approaches: in summer, where we need to isolate our houses from the sun as a source of heat; or in winter, where we need to isolate our houses so the heat produced by the heating systems is not lost to the environment. Both approaches are valid, but one might be more relevant than the other considering the climate in which the students live.

Idea 5 - 1/2: Your ideal (and sustainable) home. Dream about where you would like to live, what your ideal home would be like and how this ideal home could be more sustainable.

Collect data thanks to the board and its embedded sensors - 2/2

This part is designed to connect to the physics model of particles (matter), in which heat is a way of energy transfer, related to the movement of particles. It is important to identify where the energy source is (sun, heating system) and the transference process (from the source). Two important misconceptions (https://journals.flvc.org/cee/article/download/87720/84517/) in this part are that **isolating materials** "heat" (i.g. a wool jumper "heats" us) and that the **cool also "travels"** (i.e. we can feel how the "cool" enters through the window if we open it during winter). It is important for the teachers to identify if students' are holding these misconceptions and offer alternative experiments to build on these ideas (i.e. explore what would happen if we put an ice surrounded by wool. Would it melt faster?).

Display the data to get the needed information

In the previous section, we built a sensor and design an experiment to test the energy efficiency of our homes. However, in order to assess this efficiency, we would need to gather this information and assess the materials used.

To show the temperature that is measuring the sensor, the first solution could be to use the LED display. Another possibility is to program the board so this information is stored and transferred to a computer in a CSV format afterwards.

A function to interrogate the temperature sensor in the board can be used.

Analyse the data and learn from them

Instant temperature data have allowed us to explore the heating conduction or isolating capacity of different materials. In this part, we will analyse this data and try to imagine how could we explain these different behaviours and use that knowledge to build our ideal home.

If students have decided to analyse the data over a certain period of time, spreadsheet software would be required. In that case, the data gathered would be needed to be retrieved from the board. Otherwise, they can take notes about the temperature of the sensor displayed on the LED. After the data analysis, students should define isolators as materials that help to keep or maintain the temperature on the inside of the home, and a conductor as a material that contributes to modifying the temperature inside the home. It is important in this part that students are able to relate the temperature gathered with the energy the air particles have (which can be described as the movement of the particles). And how this particle movement can be more or less transferred from one particle to another and from the outside to the inside and vice-versa. That is, students should be able to use the particle model to explain heat transferences, so science ideas are developed as well as technical ones.

Idea 6: Washing hands. We need to ensure that children wash their hands when they come back from the playground. Although new routines have been put in place to ensure that all children wash their hands, we are not sure that they do it well enough. How can programming help us stick to the barrier actions?

Collect data thanks to the board and its embedded sensors

A distance sensor will detect when a kid is near the sink and a time counter starts.

Display the data to get the needed information

When the time count is over, a positive sound is displayed. If the distance sensor detects the kid to leave before finishing to wash their hands, a negative sound will be played.

Analyse the data and learn from them

We can increase awareness of the required time to correctly wash our hands. If teachers identify kids which are not washing well enough their hands they can develop specific actions towards them to improve their behaviour.

Idea 7: Reasonable heating use. Identify the optimum position for using heating appliances at given times to save electricity.

Collect data thanks to the board and its embedded sensors

Using the temperature sensors of the board and installed several boards in different parts of the gymnasium or the classroom. We can also set alarms to notify the users when the temperature riched the min level.

Display the data to get the needed information

The data is saved in csv files from each board and analysed.

Analyse the data and learn from them

Using the data, we can study the transmission of heat in different points of the gymnasium/classroom with the time needed to heat the points that are situated the most far away from the heating device. The data collected will be used for making maths computations in order to optimise the heat consumption.

Idea 8: Music: Can you play what you hear? Have you ever wished you could play a song on the piano just by listening to it?

Collect data thanks to the board and its embedded sensors

If your students do not own a piano or keyboards, you may use the board to train them to play music by ear. You may play a song (e.g. https://www.youtube.com/watch?v=5M_YKXax2IA) and then ask them to use the board to reproduce the song using the music activity sheet.

Display the data to get the needed information

Ask your students to use the MakeCode blocks to reproduce the melody by setting the beat, tone, volume and tempo.

Analyse the data and learn from them

What have your students learnt about the songs' beat, tone, volume and tempo? Ask them to reflect on the learning outcomes and the difficulties they faced. Try other popular songs for extra practice.

Contact the Let's STEAM members for more info

IDEA #1, IDEA #2 & IDEA #8 - STÉPHANE VASSORT - AIX MARSEILLE UNIVERSITE - FRANCE
stephane.vassort@lets-steam.eu

IDEA #3 - CINDY SMITS & TOON CALLENS - DIGITALE WOLVEN - BELGIUM
cindy.smits@lets-steam.eu - toon.callens@lets-steam.eu

IDEA #4 & IDEA #5 - MERCÈ GISBERT CERVERA, CARME GRIMALT-ÁLVARO - UNIVERSITAT ROVIRA I VIRGILI - SPAIN
merce.gisbert@lets-steam.eu - carme.grimalt@lets-steam.eu

IDEA #6 - MARGARIDA ROMERO - UNIVERSITE COTE D'AZUR - FRANCE
margarida.romero@lets-steam.eu

IDEA #7 - MARYNA RAFALSKA - UNIVERSITE COTE D'AZUR - FRANCE
maryna.rafalska@lets-steam.eu

FULL EXAMPLE

HOW TO MAKE THE INVISIBLE VISIBLE?

Author: Stéphane Vassort, stephane.vassor@lets-steam.eu

Step 1 - Present the project as a whole

We invite you through this template to get creative while getting technical support in designing a unique and inclusive project! You are free to develop your own solution or to be inspired by solutions proposals. In the end, depending on the path you choose, your solution will be unique!

Describe your project

Name your project: How to make the invisible visible?

Short introduction of what your project is about, the problem tackled behind, the pedagogical objectives

This project consists in developing communicating terrariums for frogs. It aims to raise awareness of climate issues through the discovery of the dendrobatid environment. We propose to monitor the temperature in a terrarium to ensure that ideal conditions (between 21 and 26°C) are present.

Reflect on equity and inclusiveness

ASPIRATIONS & MOTIVATIONS

How do you feel when doing STEM? What motivates you in STEM? What motivates your students? Are all your students motivated by the same? What would they like to do?

- Finding possibilities to apply concretly knowledge and skills into concrete projects
- Creativity as a way to promote inclusiveness
- Provide different oportunites for students' to develop their own relevant projects
- Use of digital technology for fun purposes/playful environments
- Excited about the possibility to create new artefacts

ISSUES AND BARRIERS

What worries your students? What frustrations do they have? Are there any differences that make them be at disadvantage to other students? And regarding robotic and digital in STEM activities?

- Financial resources to access continuous education in technology-enhanced learning topics
- Different objectives according to gender (service vs. fighting)
- Potential difficulties on technological material

KEYWORDS

Indicate 3 or more keywords that describe the reality of your students regarding STEM/STEAM activities.

- NEW
- EXCITING
- SCARY

Step 2 - Collect data thanks to the board and its embedded sensors - 1/2

At this stage, you are required to find a programming solution to collect your data, identify which sensors to be used and how to program them on MakeCode for the platform to communicate with your board.

ORIENTATION

Define what is the problem to be solved, what are data to be collected, what are the learning objectives behind the programming topic?

Context: In order to reproduce the natural environment of frogs and ensure their survival, different parameters of their living environment must be taken into account. What information do we need to know in order to provide them with the most appropriate living environment?

Learning objectives: Identify useful sensors and the procedure to implement them with a programmable board.

CONCEPTUALISATION

Formulate a hypothesis to answer the given problem regarding data collection

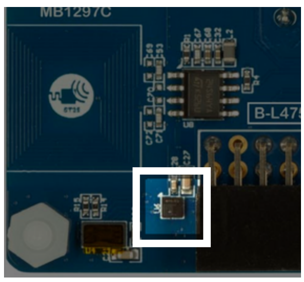
As the main parameter to be controlled to ensure the frog's survival is the **temperature**, and that it must be between **21 and 26 °C**, the solution that seems to be the simplest is to use the **temperature sensor** integrated into the STM32 program card.

INVESTIGATION

Describe the steps you need to collect the data that will be necessary for your project.

This step can be implemented thanks to activity sheet **#R1AS11 - Make a very readable thermometer.** In this activity, we learn how easy it is to read the temperature sensor of the board and display its value.

This temperature sensor is located next to the "time of flight" sensor on the right, it is used to implement activities linked to the monitoring of heat or to approaching meteorological concepts. In our case, it will help to monitor the temperature inside the vivarium.

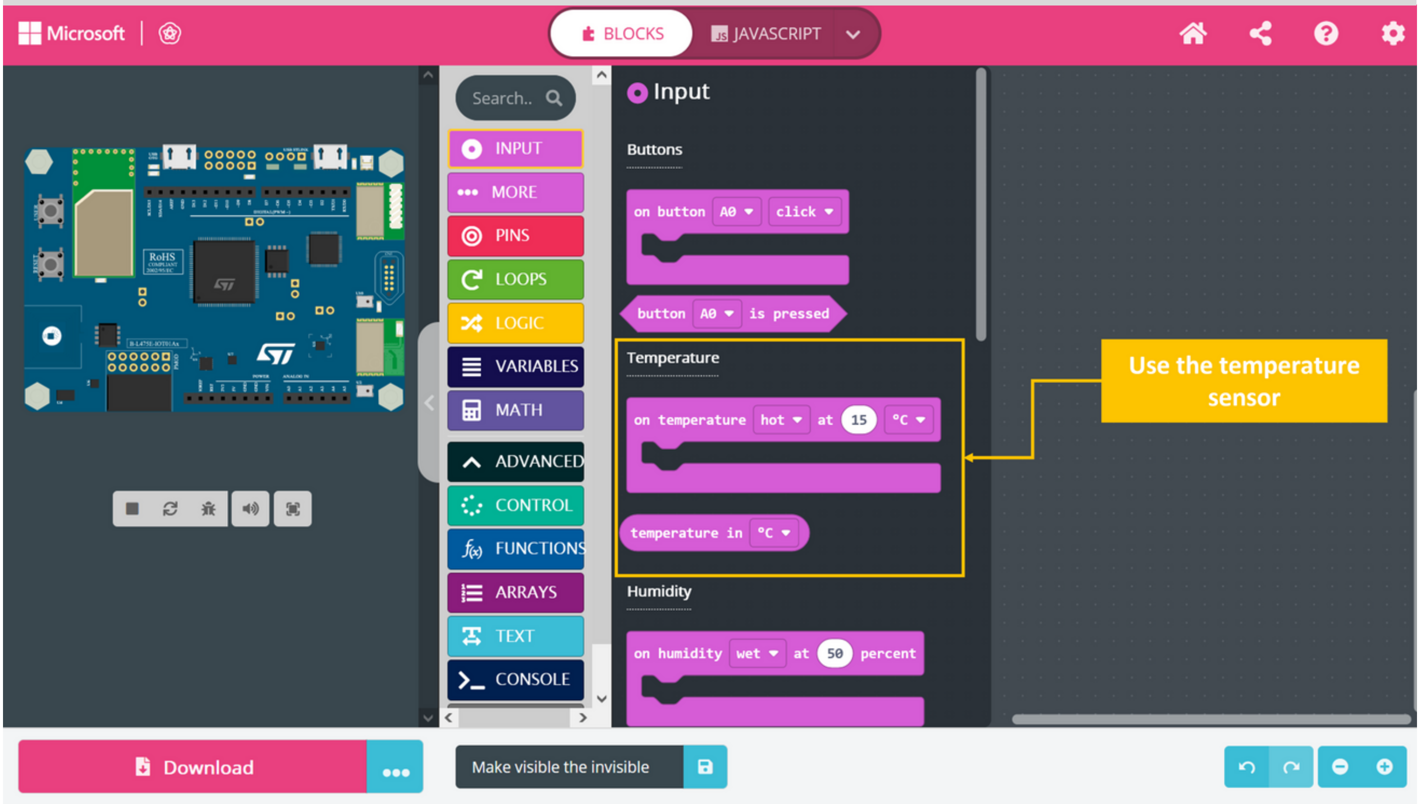


It is possible to ask for the temperature sensor integrated into the board with the block-based programming software available in MakeCode in the "INPUT" blocklist.

Ability to measure temperature

In order to be fully functional, it is necessary that the temperature sensor can operate at least up to 50°C. In order to verify that the sensor will be operational, I looked at the indicator of the temperature of the STM32 board that shows the range measurable by the sensor from -5°C to 50°C. Thus, the choice to use the integrated sensor seems quite satisfactory and sufficient.

Provide screenshots of the MakeCode platform and of your board



DEBRIEF

Identify the knowledge mobilized during this phase, think about your classroom and identify possible learning, add references issues that may come up

Through this step, we were able to define that in order to obtain information about the external environment, a programmable card can use sensors.

For the example of the STM32 card, if we want the program with visual block-based programming software, functions exist to dialogue with its integrated temperature sensor and thus obtain the temperature in degrees Celsius.

A sensor does not have an infinite measuring range, so it is important to check the adequacy between its possible measuring range and the measurements to be made.

At this stage, you are required to find a programming solution to display your data, enabling, now you have asked a sensor to obtain information, to make this information known to the user.

ORIENTATION

Define what is the challenge in the display of the data you need? For you? For your classroom? For the user?

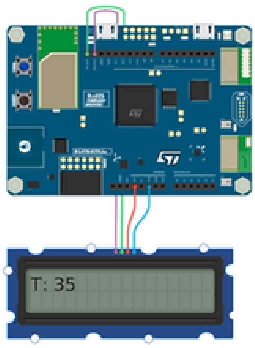
Context: We could see in the previous part how to ask a sensor to obtain information. It would be useful now to be able to make this information known to the user.

Learning objectives: Identify an actuator and control it in order to be able to deliver information

CONCEPTUALISATION

Formulate a hypothesis to answer the given problem regarding data display

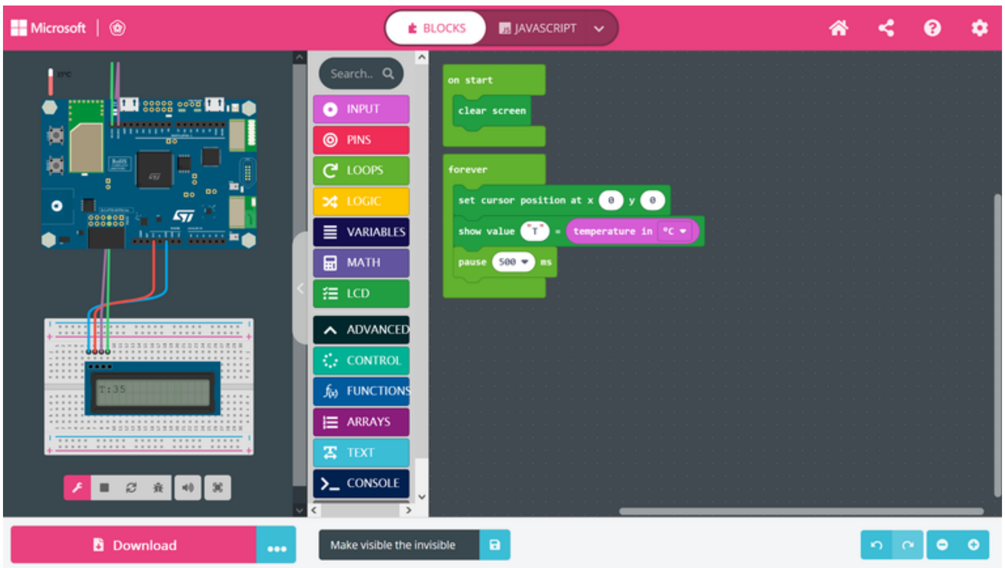
In order to inform the user of the measured temperature, the first solution that comes to mind is to use the **external LCD Text Display**.



INVESTIGATION

Describe the steps you need to collect the data that will be necessary for your project

This step can be implemented thanks to activity sheet **##R1AS10 - Text display with an OLED screen,** a screen helping you to display some pieces of information hidden inside your electronic components. From the documentation of the STM32 card, we can read the functions used for showing the data on LCD display: "**set cursor position at x: y:** " and "**show value** ".



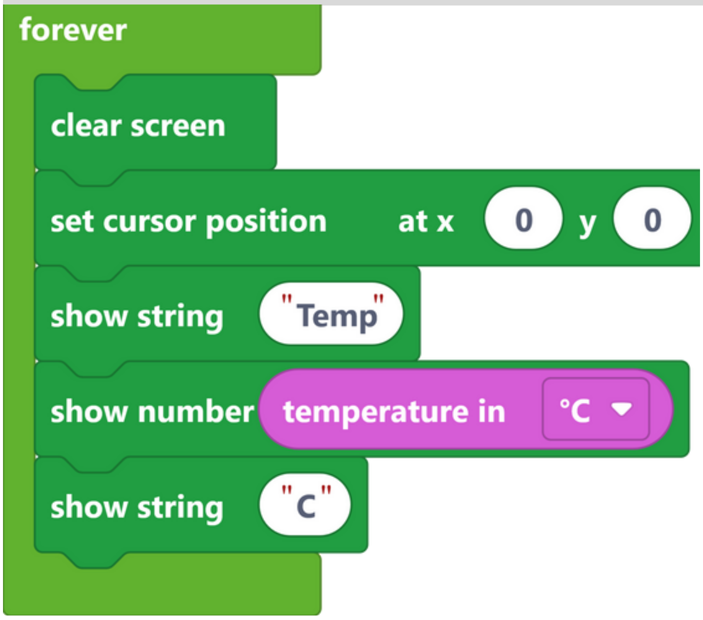
Test program

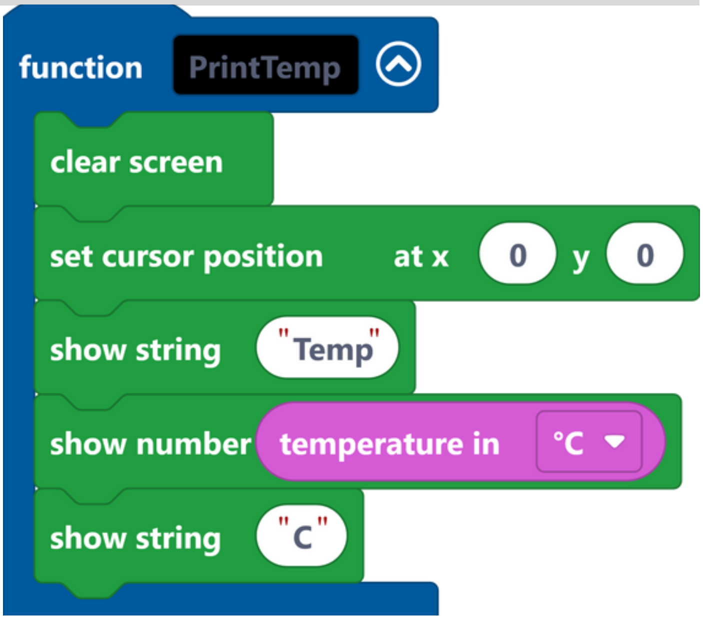
In order to check if it works, I tested a first program that performs the following task:

- Clean the LCD screen,
- Identify the position of the cursor (on x=0 and y=0),
- Write the word “Temp”, display the value measured by the temperature sensor and write the word “C” (in order to indicate that the temperature is measured in the Celsius scale).

In order to be able to call this program (sequence of blocks) from another program, I replace the loop “forever” with the function block. The function is called “**PrintTemp**”.

Provide screenshots of the MakeCode platform and of your board





DEBRIEF

Identify the knowledge mobilized during this phase, think about your classroom and identify possible learning, add references issues that may come up

Thanks to this step, we were able to connect the LCD screen to the STM32 board.

Note on data types

The data provided but the temperature sensor is an integer and the letter C for the unity is a string, that’s why we used two different blocks:” show number” and “show string”.

In order to structure a program, it is possible to define a function for each task to be performed.

Now we are able to display data instantly, we need to analyze them to perform monitoring of our information (for instance, monitoring of temperature, of alerts, motion, frequency ...). This stage is made for enabling this analysis on the editor.

ORIENTATION

Define what is the challenge in this step according to your project. What is your challenge in analysing and extracting the relevant information applied to your context?

Context: We are able to display data instantly. In order to be able to analyze variations in climatic conditions and identify when the temperature level becomes critical for our frogs and the frequency of these alerts, it would be useful to be able to perform this monitoring over a long period of time.

Learning objectives: Analyze data and extract relevant information

CONCEPTUALISATION

Formulate a hypothesis to answer the given problem regarding data analysis

In order to be able to analyze the data from the temperature sensor over a long period of time, I think that using spreadsheet software would be a simple solution. For this, it is necessary to be able to retrieve the data from the programmable board. The solution that I will implement will be to write via the serial port the data in CSV format (comma-separated value) which is exploitable by a spreadsheet program.

INVESTIGATION

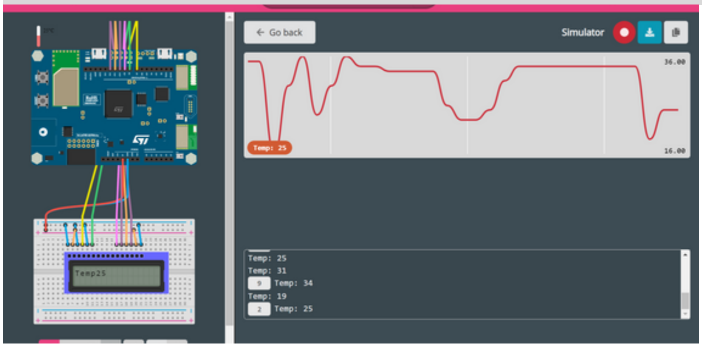
Describe the steps you need to analyse and monitor the data that will be necessary for your project

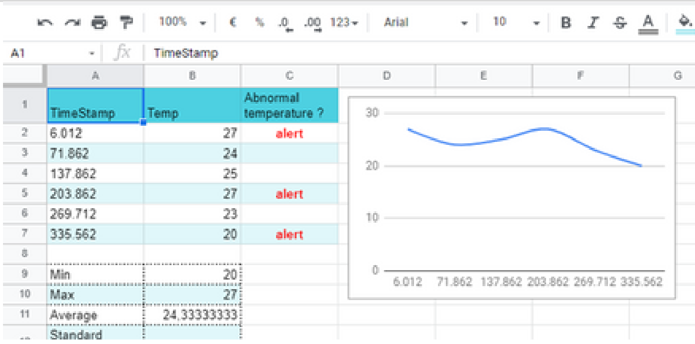
You can use the following resources as a start: https://en.wikipedia.org/wiki/Comma-separated_values. According to the documentation, a CSV file is a simple text document containing data to be presented in table form. The table headings are on the first line, and the data are then inserted line by line. In order to differentiate the data, they are separated by a comma, hence the name of this file format.

Test program

In order to retrieve the data for analysis, I will write to the serial console the temperature provided by the onboard sensor every minute. I'll just have to show the graph and download the data as a CSV file. This document can then be opened with a spreadsheet program. It is thus possible to obtain the mean, minimum, maximum temperature or standard deviation.

Provide screenshots of the MakeCode platform and of your board





	A	B	C
1	TimeStamp	Temp	Abnormal temperature ?
2	6.012	27	alert
3	71.862	24	
4	137.862	25	
5	203.862	27	alert
6	269.712	23	
7	335.562	20	alert
8			
9	Min	20	
10	Max	27	
11	Average	24.33333333	
	Standard		

DEBRIEF

Identify the knowledge mobilized during this phase, think about your classroom and identify possible learning, add references issues that may come up

Thanks to this step, we were able to discover that a programmable card could also send information via a serial console.

This feature allows to send information faster than using the integrated screen but requires a connected computer.

CSV format

The serial console allowed us to send a text file in CSV format which could then be opened by spreadsheet software to analyze the data.

From this data, a spreadsheet program can easily draw graphical representations or perform statistical calculations.

Now we are able to collect, display and monitor data, we can actually create a solution for using these data in real life for a concrete purpose. This additional step to this project will enable creating a real use case for the whole activity.

ORIENTATION

Define what is the challenge in this step according to your project. What is the concrete objective for the user?

Context: We are now able to measure and analyze the data from the sensors. It would be useful to be able to notify the user about the temperature in the vivarium and in case of detection of the temperature in that it is too high to be able to decrease it.

Learning objectives: Identify a condition and implement a conditional block.

CONCEPTUALISATION

Formulate a hypothesis to answer the given problem regarding this additional step

There are two tasks to perform here:
1. **Notifying the user** about the temperature in the vivarium in the most visible way, for example, by changing the colour of the LCD screen;
2. **Open a window** while the temperature becomes too high.

In order to automatically identify in what temperature range the current state is, and show the corresponding colour of the LCD screen to the user, I will use a conditional "**IF**" block.

INVESTIGATION

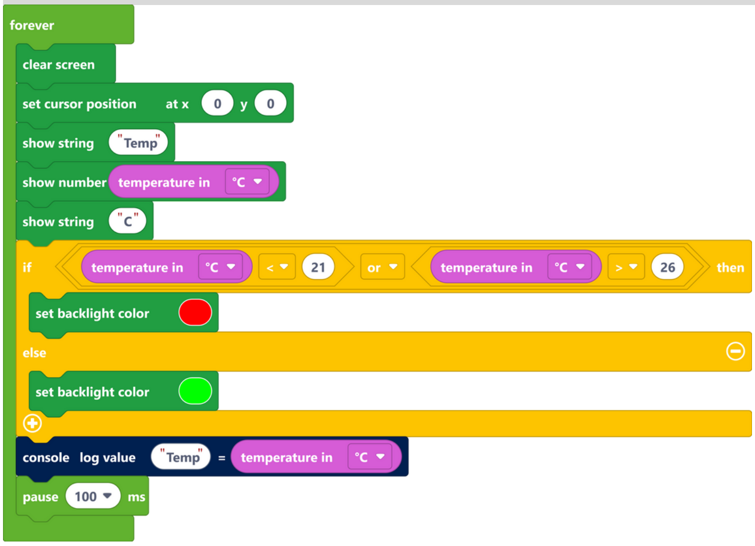
Describe the steps you need at this step in your project

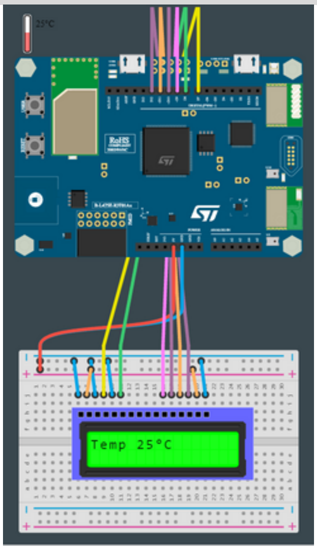
Test program

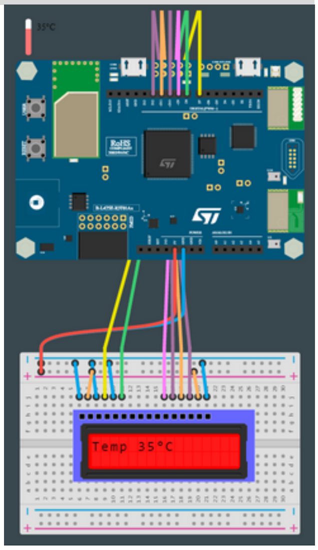
In order to notify the user, the program will change the colour of the LCD screen according to the temperature in the following way:

- **-5..21 C° - red light**
- **21 .. 26 C° - green light**
- **26..50 C° - red light**

Provide screen shots of the MakeCode platform and of your board





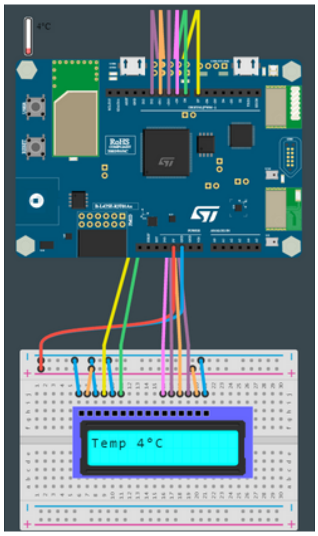


I decided to add one more light in order to notify the user more precisely about the temperature in the vivarium:

- if the temperature is less than 21 C° - turn on the blue light,
- if the temperature is between 21 et 26 C° - green light
- if the temperature is more than 26 C° - red light.

For this, I used the conditional block "**If .. then.. else** ". In each case, I call the function "**PrintTemp**" (that I created at the first stage of my work) to print the current temperature on the LCD screen.

In order to be able to open the window, I connected the stepper motor to the STM32 card. Then I create the function "EmergencyVentilation" that I call in case the temperature is more than 26 C°.



DEBRIEF

Identify the knowledge mobilized during this phase, think about your classroom and identify possible learning, add references issues that may come up

Conditional Loop

Thanks to this step, we were able to discover what a conditional instruction is and its versions: short "if .. then" and long one "if.. then .. else ".

It is an algorithmic structure that will execute an action only if a condition is verified. In our case, an LCD screen with turn on the blue, green or red lights if the temperature is respectively in one of the ranges -5..20, 21..25 or 26..50 C°.

Adding new devices

In order to benefit from new features, it is possible to add extensions providing additional functions.

Here we have added the stepper motor to turn on the ventilation in case the temperature becomes more than 26 C°.

WANT TO GET INVOLVED?

ASK US QUESTIONS

Get in touch with us by mail: manon.ballester@lets-steam.eu

GET TRAINED THROUGH THE LET'S STEAM SESSIONS

Physically in one of the partner countries: Greece, France, Italy, Spain, Belgium

Online through our e-learning platform - Coming soon

GIVE US FEEDBACK AND CORRECTIONS

This coursebook has been made with the best quality as possible and a true will to participate in the emergence of amazing content in the field of programming. Though, we are only humans! Should you discover mistakes or corrections to be made, do not hesitate to get in touch with us! We will make sure you get rewarded and credited for your help!

PARTNER WITH US IN NEW PROJECTS

All the members of the Let's STEAM consortium are open to new cooperation, either with schools but also with creative companies and actors. We are launching regularly new initiatives. Keep us updated if you want to join them with us!

FIND US ONLINE

www.lets-steam.eu

@letssteamproject

@lets_steam_eu

www.lets-steam.eu