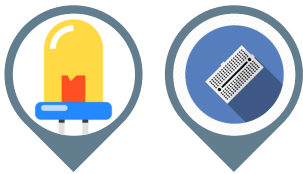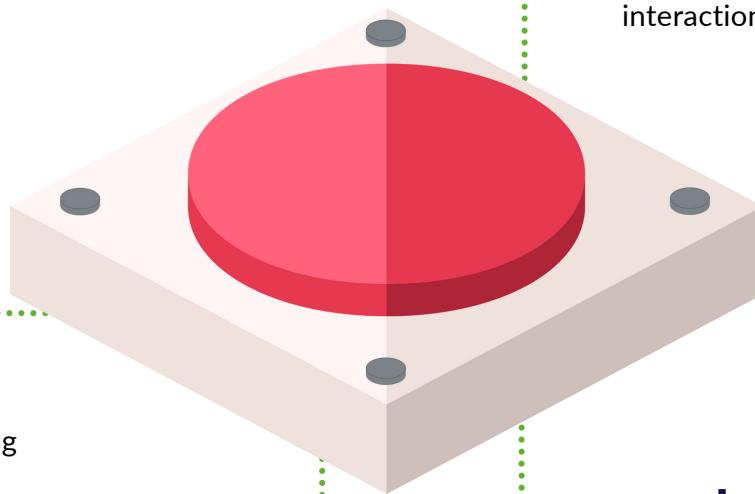# BUTTONS AND LED DISPLAY

#R1AS03

Available on

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!

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

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

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**.

> **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.**

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.

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**". Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finishes blinking, your quiz buzzer is ready!
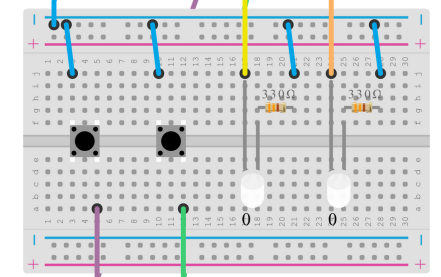
Run, modify, play

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!
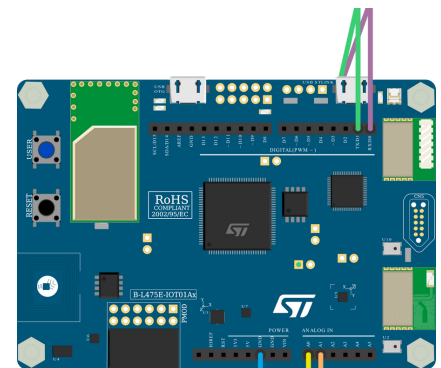
Wiring buttons and LEDs

BUTTONS AND LED DISPLAY

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.

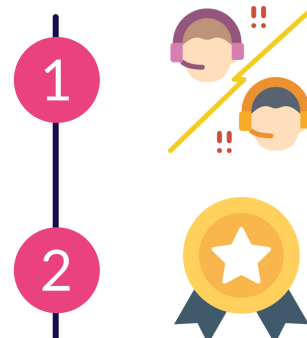# BUTTONS AND LED DISPLAY

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.
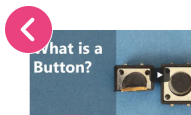
GOING FURTHER

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