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

Pre-requisites

- R1AS04 - Basic Light Sensor

Duration

50 minutes

Material

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

Level of difficulty

Advanced

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

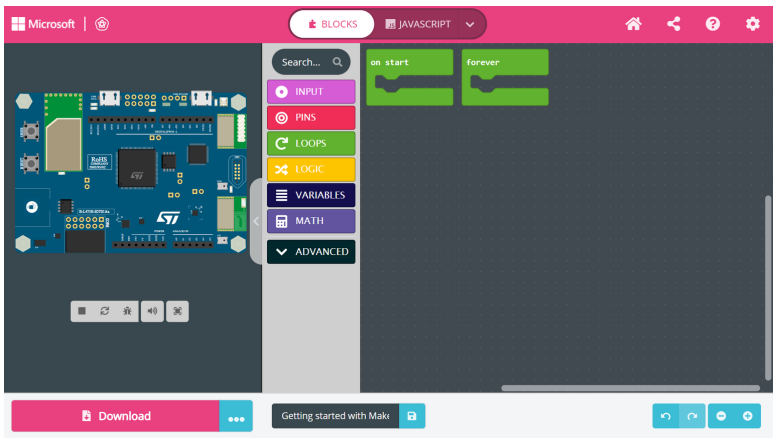
Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.

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

COLLECTING DATA

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.

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!

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.

Get your data

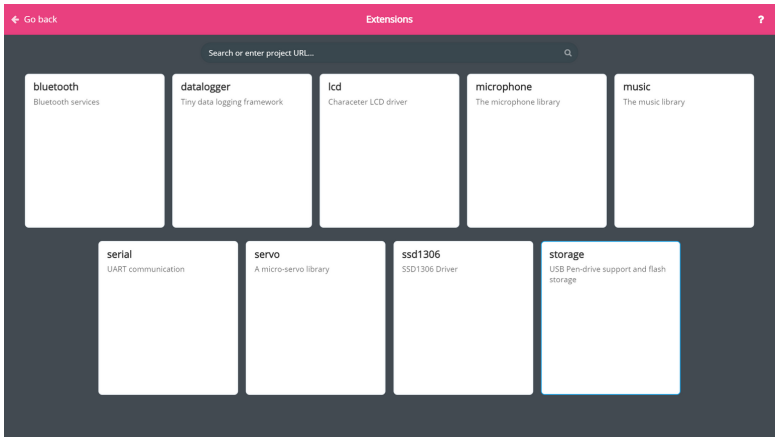
With your USB Cable, connect the board to your computer by using the USB OTG connector (the left one when you look at the board from the upside). When your project is logging, a new flash drive should appear named **MAKECODE**.
The **SPIFLASH** directory contains program data. Logging data is written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

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.

4

Advanced functionalities appeared

5



List of extensions and searching tool

6

Datalogger and associated blocks

COLLECTING DATA

STEP 1 - MAKE IT

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

View your data

7

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!

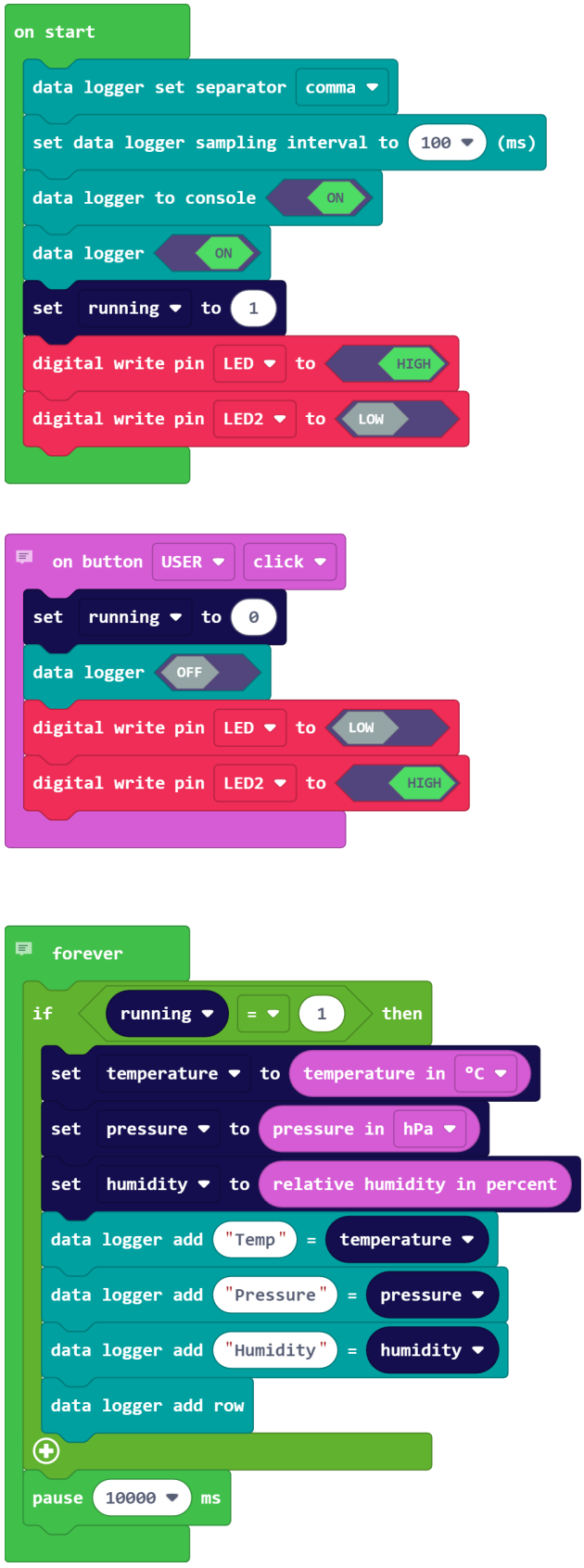
Run, modify, play

8

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

COLLECTING DATA

STEP 2 - CODE IT

```
//Init the data collection
let running = 0
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.

COLLECTING DATA

STEP 3 - IMPROVE IT

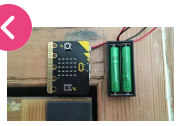
1. **Add a battery pack** to your board to make experiments on environmental sensors in many contexts.
2. Allow to restart the data collection process by **clicking again on the USER button**.
3. **Produce some graphic** that compare multiple data collection sessions.
4. **Record sensors remotely** by 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

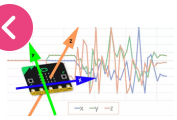
Flash memory - Learn more about flash memory, an electronic non-volatile computer memory storage medium. *https://en.wikipedia.org/wiki/Flash_memory*

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

R1AS08 - Make a theremin with the distance sensor

R1AS11 - Make a very readable thermometer

R1AS12 - Motion Detection Alarm