

Project number: 2023-1-PL01-KA220-SCH-000154043

IoT4Schools

“Bringing the Internet of Things in school education as a tool to address 21st century challenges”

Weather station project: measuring environment related parameters

Teachers’ guidelines

Authors: C.Papasarantou, R. Alimisi

Organization: EDUMOTIVA

License: CC BY-NC 4.0 LEGAL CODE, Attribution-NonCommercial 4.0 International



**Co-funded by
the European Union**

The European Commission's support to produce this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

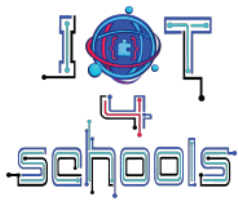
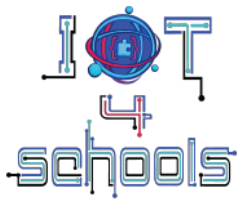


Table of contents

1	Introduction to the project	3
1.1	Scenario and Scope of the project	3
1.2	Learning objectives	3
1.3	Learning pathway – Stages of implementation	3
1.4	Learning prerequisites	4
1.5	Hardware and software	5
1.6	Time plan	5
2	Implementation of the project.....	6
2.1	Level 1: Creating a weather station that measures temperature and humidity.....	6
2.1.1	Circuit making process.....	7
2.1.2	Programming.....	9
2.1.3	Crafting.....	17
2.2	Level 2: Adding a sensor for measuring barometric pressure	21
2.2.1	Circuit making process.....	21
2.2.2	Programming.....	22
2.2.3	Crafting.....	24
3	Tips and recommendations.....	24
3.1	Further expansion of the project.....	24
3.2	Creating an autonomous weather station	24
3.3	Optimizing the appearance of the device	24
3.4	Discussing the advantages and disadvantages	25
4	References.....	25
5	Appendix	26
5.1	Glossary – Definitions of key concepts	26
5.2	ThingSpeak Channel Stats options	26
5.3	Barometric pressure warm-up activity	28
5.4	Adding and programming an LCD I2C screen	28



1 Introduction to the project

1.1 Scenario and Scope of the project

The aim of this project is to introduce students to the concept of IoT in the context of environment and smart homes, and through the lens of measuring and monitoring weather and atmosphere related parameters. They will learn how to create their own weather station (i.e., a device that measures parameters related to weather and atmosphere), how to send the measured data to an IoT platform, and monitor them in real time. Specifically, using the BBC micro:bit microcontroller and several sensors, students will learn how to create and program their own weather station that can measure parameters including temperature, humidity and barometric pressure. Using a WiFi module (the ESP8266 WiFi module) and the ThingSpeak IoT platform (<https://thingspeak.mathworks.com/>), they will also learn how to send the measured data to the cloud via WiFi, and monitor them in real time. This project will help students to become familiar with the process of remotely measuring data and monitoring them in real-time, making decisions about their daily (indoor or outdoor) activities based on this data.

By addressing the concepts associated with the IoT and suggesting ways for integrating and applying these concepts in the context of school education, this project also promotes methods to help both teachers and students improve their digital skills and competences, thereby contributing to the digital transformation of school education. Furthermore, this project encourages interdisciplinarity and the integration of STEM principles through the implementation of practices from diverse fields, including Technology, Physics and Environmental studies. Additionally, the developed educational resources (teacher's guidelines, student worksheets, etc.) provide instructional materials that can effectively support educators in the smooth implementation of the project within the classroom.

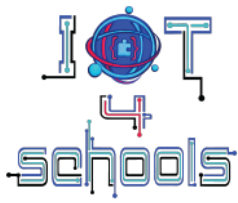
1.2 Learning objectives

Through this project the students will be able to:

- Build and program a device that can measure weather and atmosphere related parameters
- Use and program a WiFi module to send the measured data to the cloud
- Use an IoT platform and create a channel to monitor the measured data in real time
- Learn how to use sensors including the DHT-11 and the BMP280 to measure data related to weather and atmosphere
- Understand how IoT platforms work
- Understand and explain how data can be monitored in real time
- Identify the advantages, disadvantages and risks using such devices and applications for making daily decisions
- Understand what temperature, humidity and barometric pressure are
- Understand the importance of monitoring weather and atmosphere related parameters such as temperature, humidity and barometric pressure
- Identify real-world scenarios in which this monitored data can be used

1.3 Learning pathway – Stages of implementation

The project raises the challenges of environment and smart homes, and the ways that IoT can be implemented to help us on taking a number of decisions through monitoring data relevant to the weather and the condition/status of the atmosphere. To this end, the students will be encouraged to create their own weather station by using the micro:bit board.



Here are some suggested stages to smoothly and effectively implement the weather station project with your students:

Group formation: Divide your students in teams of two or three.

Brainstorming: Encourage each team to find out more about weather stations (e.g., how they work, what data they measure, where this data is sent or stored, how this data can be used to improve our lives etc.). If students are already using weather stations at home, encourage them to reflect on how they use this device (which data they mainly check, how often they check this data, if and what decisions they make based on the measured data etc.). Also encourage them to look up the definitions of key concepts such as temperature, humidity and barometric pressure, and to consider why it is important to measure such parameters.

Discussion and assignment of the activity: Encourage each team to share their findings and ideas about the use of a weather station in the plenary and have a general discussion based on the aspects they have highlighted. Following the discussion, introduce the specific aim of the project, namely the creation of a device that can measure weather and atmosphere related parameters, and the remote monitoring of these data using an IoT platform. (*Note: It is recommended that the specific aim of the project is introduced after the brainstorming in order to encourage your students to consider the weather station project in a broader context.*)

Planning: Encourage each team to think about how they will construct the weather station (i.e., how they will connect and program the sensors, and create a model to embed these sensors etc.), and how they will manage to send this data to the cloud (what type of connectivity they need etc.).

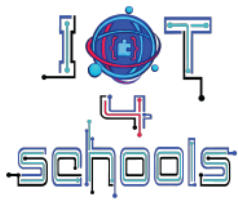
Creation: Using the student worksheets, encourage each team to create their own weather station. Depending on the skills of the students, you may wish to consider role allocation.

Testing - optimization: After completing the project, encourage your students to test their weather stations. They can be encouraged to simulate specific weather conditions (extreme heat, extreme cold etc.) and observe the performance of the weather station. You can suggest that they test another team's weather station to see if there are any differences in the way the devices work. Based on the test results you can encourage each team to optimize their project.

Presentation - sharing: Encourage your students to present their projects in plenary and ask them to reflect on the whole experience. Encourage all teams to consider the impact of such devices on people's daily lives and the advantages and disadvantages of using weather stations to make several daily decisions (what outdoor activities will do etc.).

1.4 Learning prerequisites

The students should be familiar with basic block-based programming methods and software. No other previous experience or learning background is required.



1.5 Hardware and software

Hardware:

- The BBC micro:bit microcontroller board
- External power source: 2AAA battery holder or a power bank of low voltage output (up to 5V)
- Edge connector breakout board (with I2C I/O pins)¹
- F-F jumpers
- ESP8266 WiFi module
- DHT-11 Digital Temperature and Humidity sensor
- BMP280 – Barometric Pressure sensor (level 2)

Software:

- Microsoft Makecode block-based programming environment
- ThingSpeak IoT platform

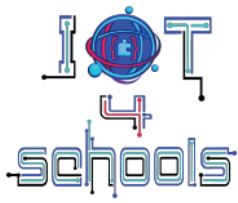
1.6 Time plan

It is estimated that you will need 5 to 7 hours to complete the project

In particular, it is estimated that you will need:

- 30-45 minutes to introduce the project (including brainstorming and discussion)
- 30-45 minutes for planning and warm-up activity
- 1 to 2 hours to complete Level 1
- 1 hour for level 2
- 1 hour for crafting
- 30 minutes for wrap-up and discussion

¹ Bear in mind that various edge-connectors with different pin configurations are available. This guide uses the Edge Breakout for BBC micro:bit (<https://grobotronics.com/edge-breakout-for-bbc-micro-bit.html>), which provides I/O pins for all of the microbit's pins, including pins 19 and 20 (i.e., the I2C pins). This edge connector has a sufficient number of I/O pins for connecting all the sensors used in the project. However, if you have a different edge connector with fewer available I/O pins, you may also need to use a breadboard.



2 Implementation of the project

2.1 Level 1: Creating a weather station that measures temperature and humidity

At this level, students will learn how to create a weather station that measures temperature and humidity and transmits this data to the ThingSpeak IoT platform, via WiFi, for real-time monitoring.

Before starting the circuit making process, ensure that your students are familiar with the concepts of temperature and humidity. To this end, you can encourage them to search the internet to find out more about these concepts. To motivate them, consider sparking their curiosity by posing some of the following questions:

Temperature-related questions:

- Why do you dress more heavily in the winter than in the summer?
- Why do some people prefer hot showers on cold days and cold showers on hot days?
- On a sunny day, which colors would you prefer to wear and why?
- How do meteorologists know what tomorrow's temperature is going to be? What tools and information do you think they use?
- What would happen if the temperature in your country increased or decreased by 20°C? Would you be able to survive?
- Could a polar bear or a camel survive in your country? Why or why not?

Humidity-related questions:

- Have you ever been outside and felt that the air was "heavy" or "sticky"? What do you think makes the air feel that way?
- Why does your hair get frizzy or curly on some days, even when you style it the same way?
- Why does a towel take so much longer to dry in a steamy bathroom?
- Why do some people put a bowl of water over or next to their radiators in winter?
- Why do you feel so much hotter on a humid 28°C day than do on a dry one?
- Why is the sky sometimes cloudy and sometimes clear?
- What would happen if the air on Earth had zero humidity? What would life on Earth be like?

2.1.1 Circuit making process

For level 1 you will need the edge connector breakout board, a DHT 11 sensor and the ESP8266 WiFi module (Figure 1).

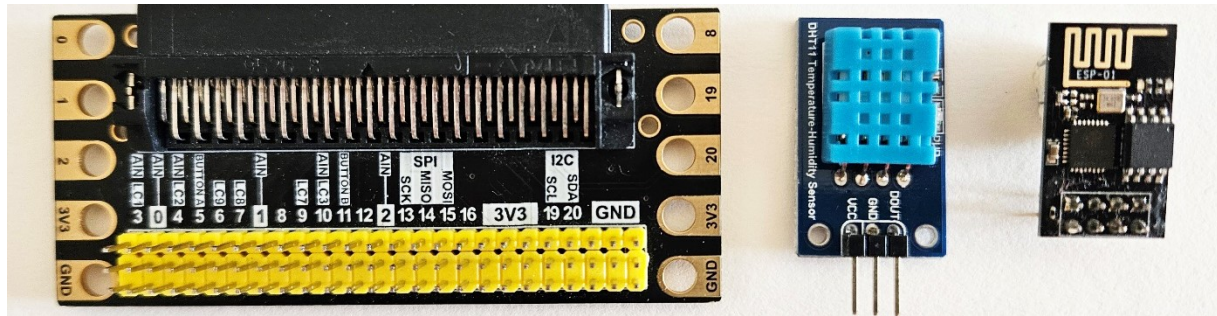


Figure 1: Hardware needed for Level 1

First, snap the micro:bit to the edge connector. Then use 3 jumper wires to connect the DHT-11 sensor to the edge connector. Specifically, connect the VCC and GND pins to the microbit's 3V3 and GND pins, and the DOUT to any other pin (P2 in the example) (Figure 2). This sensor is ready to be programmed.

Important note: Keep in mind that some DHT 11 sensors have a different pin configuration (e.g., the DOUT pin may be in the middle). Therefore, before proceeding with the circuitry, check the names of the sensor's pins.

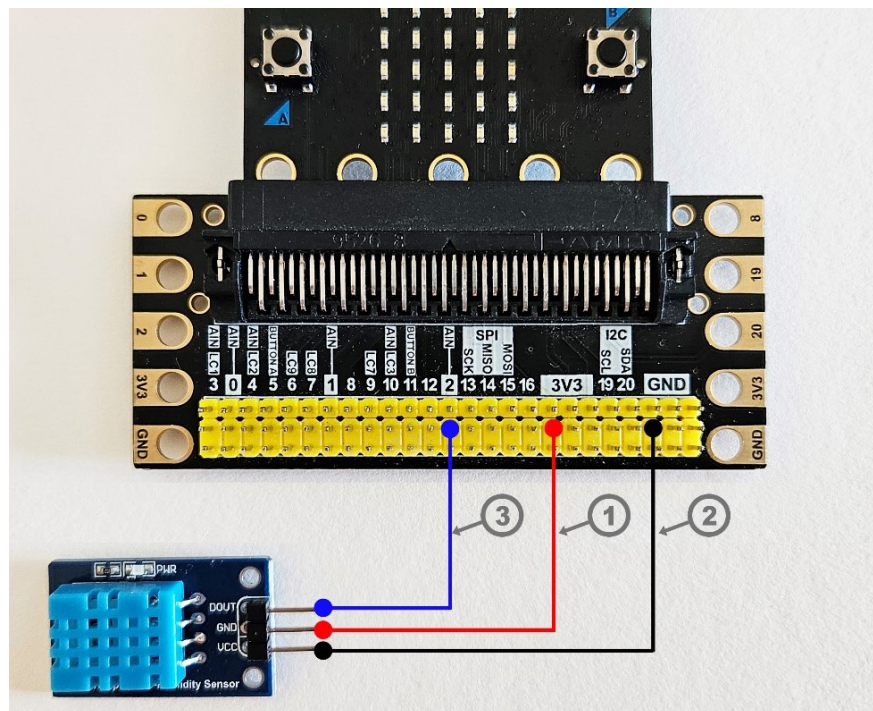


Figure 2: Connecting the DHT 11 sensor to the edge connector

Then, connect the ESP8266 WiFi module to the edge connector. The ESP8266 module has 8 pins (Figure 3). For the needs of this project, you only need to connect 5 of them to the edge connector, using 5 F-F

jumpers. Specifically, (as indicated in the following Table), you need to connect the pins 3V3 and EN to the 3V3 of the Edge Connector and the GND to the GND. Then you need to connect the RX to P0 (i.e., TX of the micro:bit) and the TX to P1 (i.e., RX of the micro:bit).

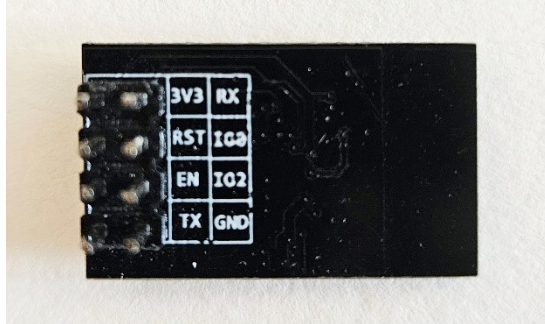


Figure 3: The pins of the ESP8266 module

ESP8266 pins	Edge Connector pins
3V3, EN	3V3
GND	GND
RX	P0
TX	P1

The circuitry is now ready (Figure 4). The last step is to connect the micro:bit to the computer using a USB cable and start to program.

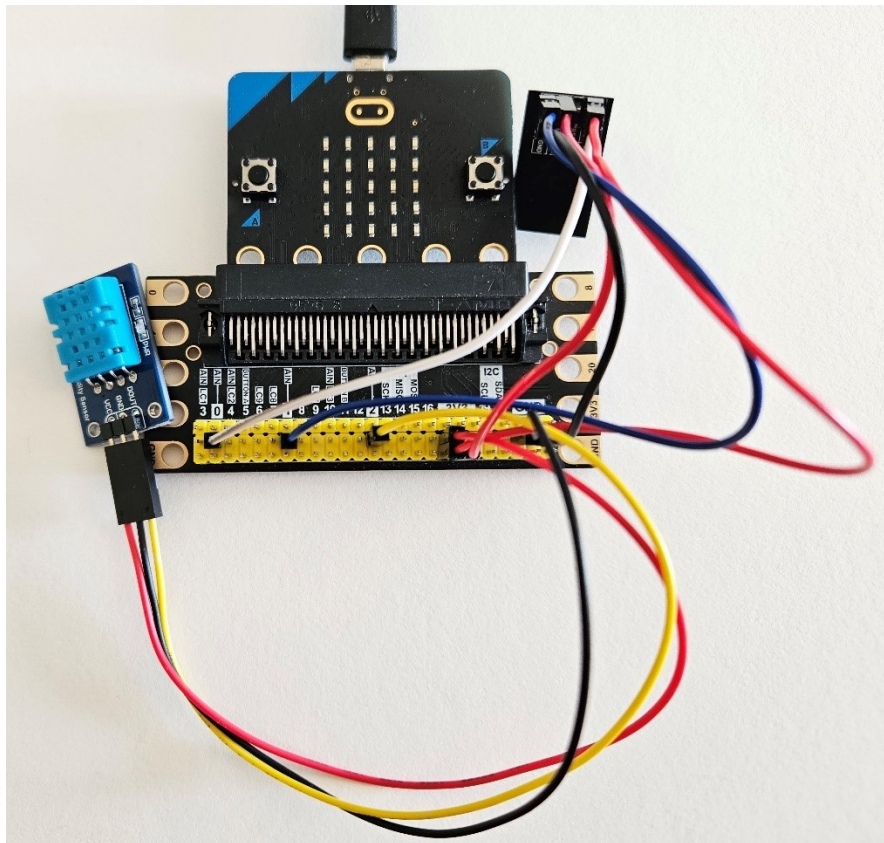


Figure 4: Indicative image of the circuitry

2.1.2 Programming

Open the Microsoft Makecode block-based environment (<https://makecode.microbit.org/>) and create a new project.

Warm up

Before programming the DHT 11 sensor to send measured data to the cloud, it is highly advisable to carry out a warm up activity to check that the sensor is functioning correctly. To do this, you need to program the sensor to display the measured parameters on the micro:bit's LED screen. The following steps will show you how to program Button A and Button B to display the temperature and humidity, respectively, when pressed.

Go to the Extensions menu and import the the “iot-environment-kit” extension (Figure 5).

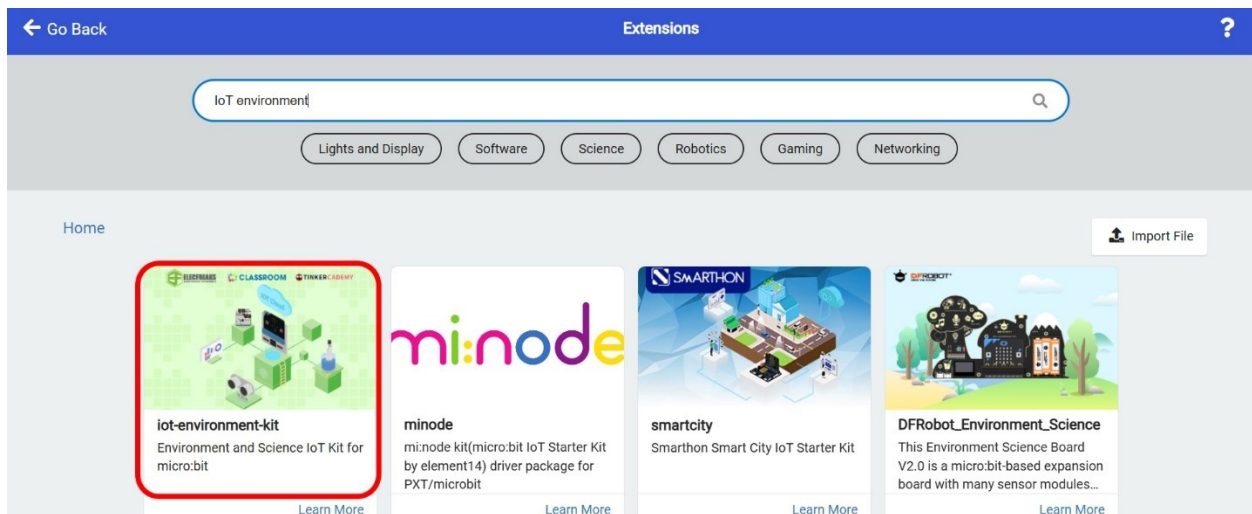


Figure 5: Finding the iot-environment-kit extension

Several new groups of commands have been imported. The block of commands for receiving the values of temperature and humidity (Figure 6, 2) are in the **Octopus...more** (Figure 6, 1) menu.

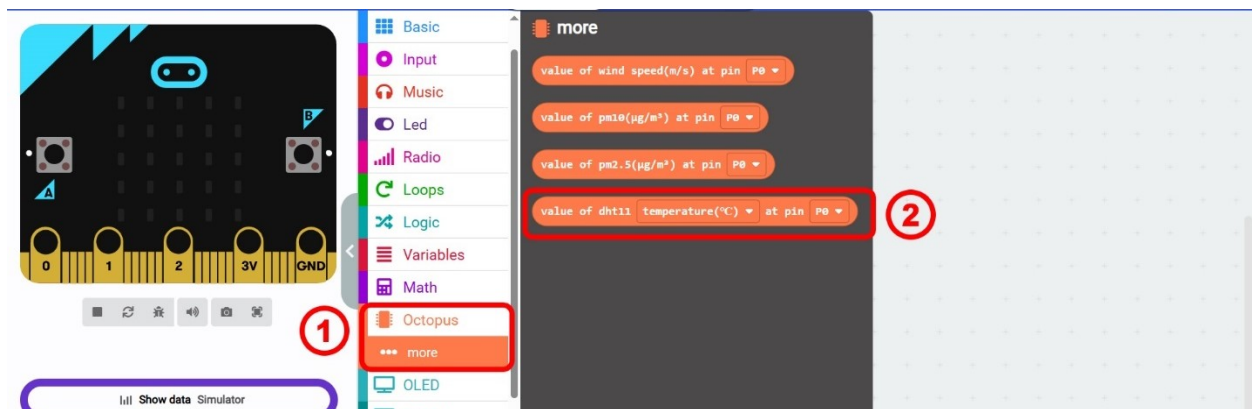


Figure 6: The Octopus...more menu (1) and the block for receiving values from the DHT11 sensor (2)

From the **Input** command group drag and drop the “**on button A pressed**” hat command into the script editor area (Figure 7a). Inside this command place a “**show number**” command from the basic menu (Figure 7b). In the value area drag and drop the “**value of dht11... at pin...**” command (Figure 8). From the drop down menu choose “**temperature**” (Figure 8, 1) and set pin to **P2** (Figure 8, 2). Then, duplicate this script and change **button A** to **button B**, and **temperature** to **humidity**.

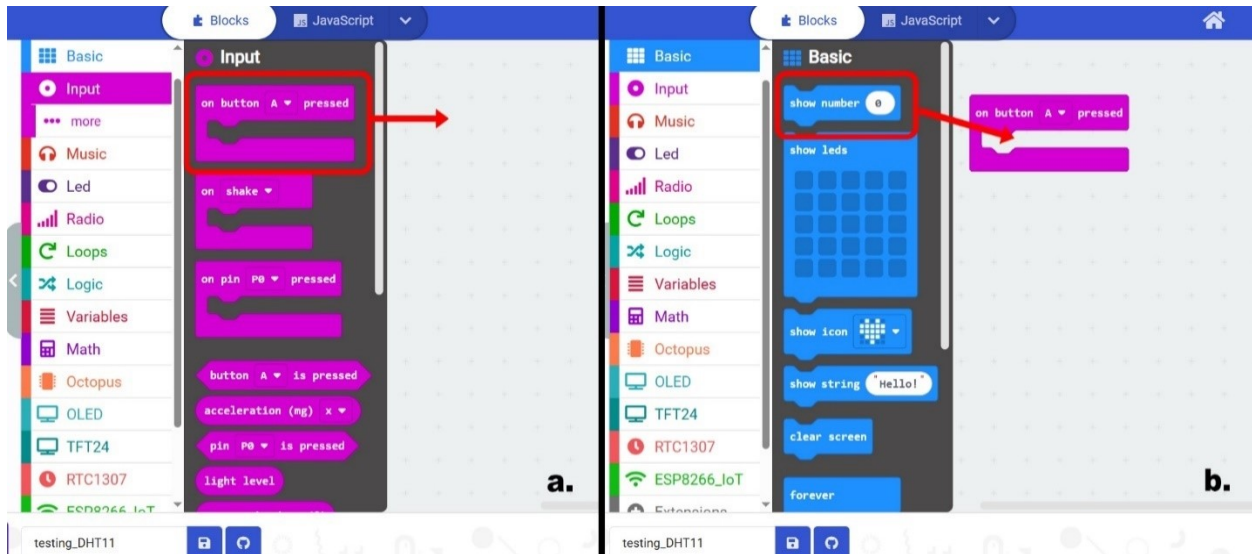


Figure 7: Finding and dragging the “On shake” command

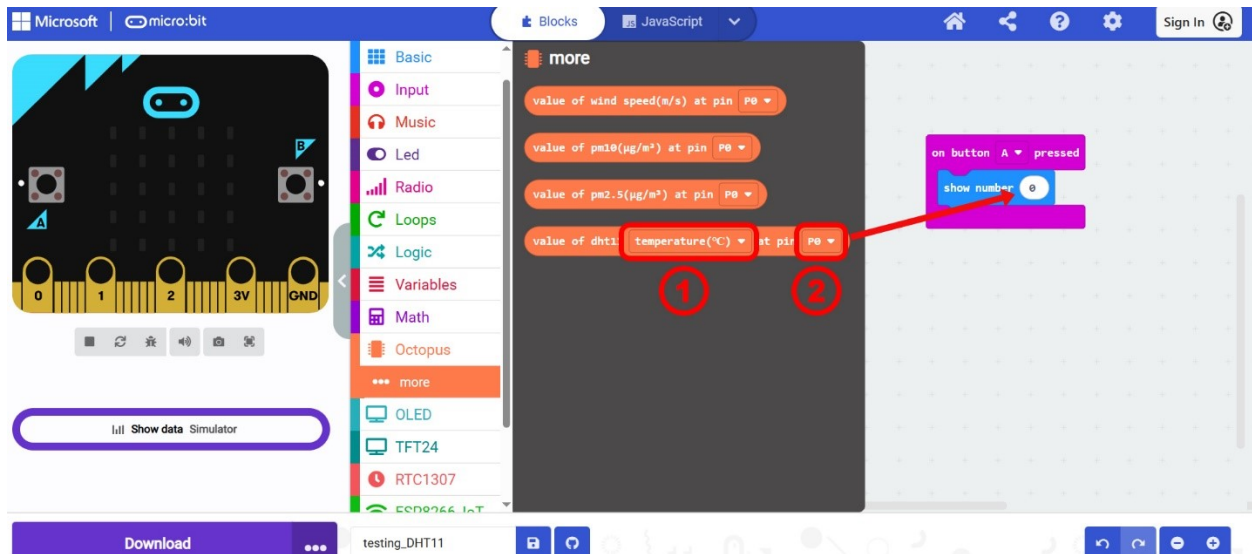


Figure 8: Adding the value of dht11...at pin... command to the show number value area

Upload the final script (Figure 9) to the micro:bit and test it.

Important note: The DHT sensor may take a few seconds to start working. You are highly advised to wait about a minute before testing the sensor.

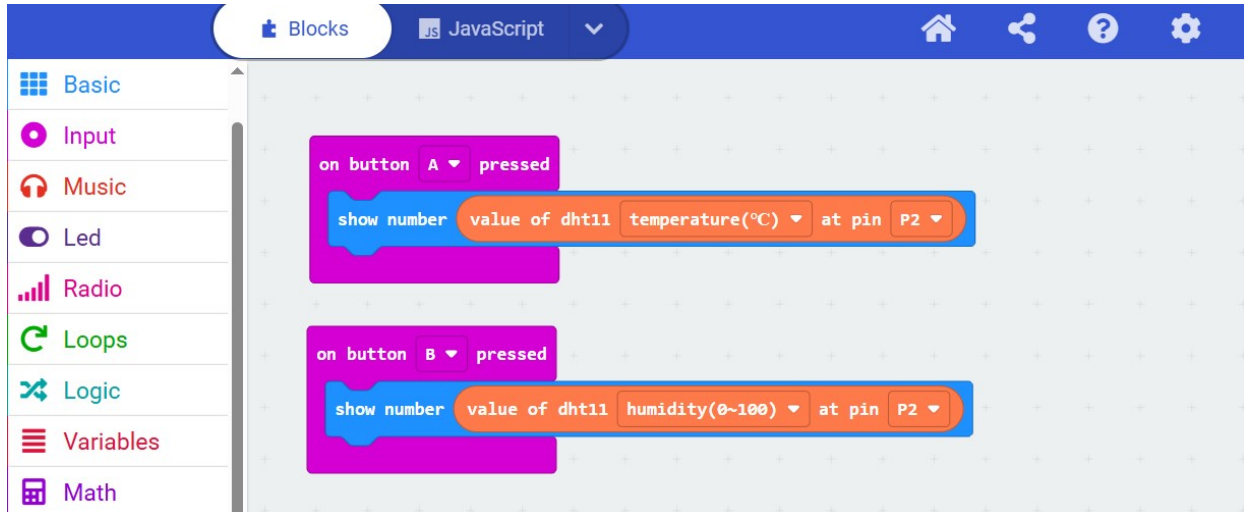


Figure 9: The final script of the warm-up activity

Towards an IoT solution: sending the measured values to the cloud

So far, you have programmed the weather station to use the DHT 11 sensor to measure values, which then displayed on the micro:bit's LED screen. However, this data is not transferred or stored anywhere other than the micro:bit itself. To transfer and monitor this data remotely, one solution is to program the ESP8266 WiFi module to send it to an IoT platform over a WiFi connection.

Programming the ESP8266 WiFi module

Note: You can either continue working on the warm-up file, or create a new one.

In order to enable the micro:bit to send data over WiFi using the ESP8266 module, the ESP8266_ThingSpeak extension must be used (Figure 10). This extension contains the necessary commands to send the measured data to the ThingSpeak IoT platform.

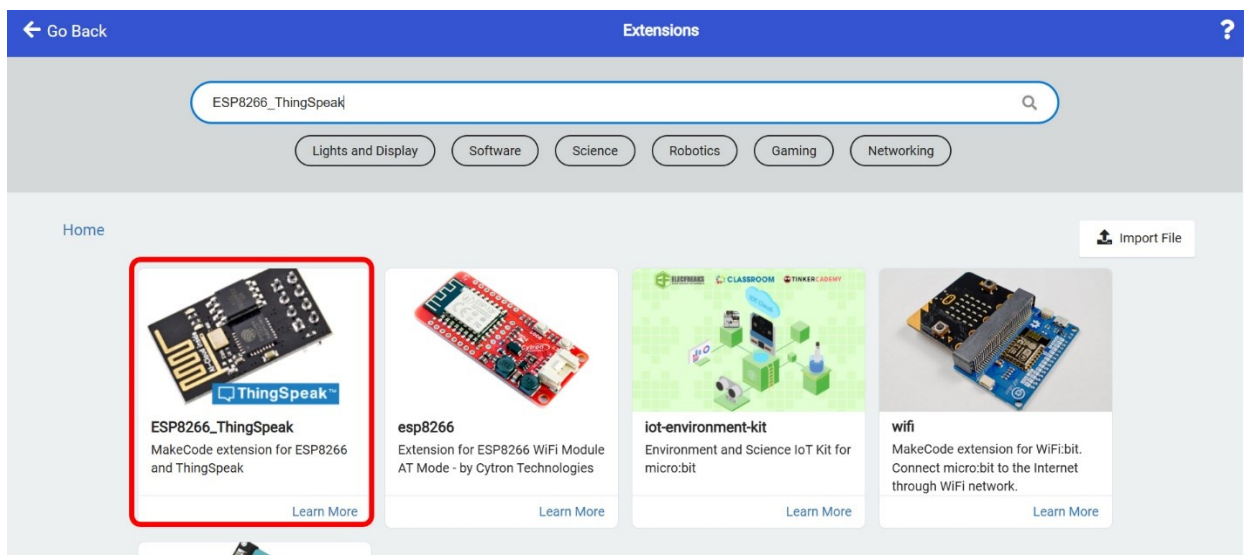


Figure 10: Finding the ESP8266_ThingSpeak extension

A new group of commands has been imported containing all the blocks needed for programming the ESP8266 WiFi module (*Figure 11*).

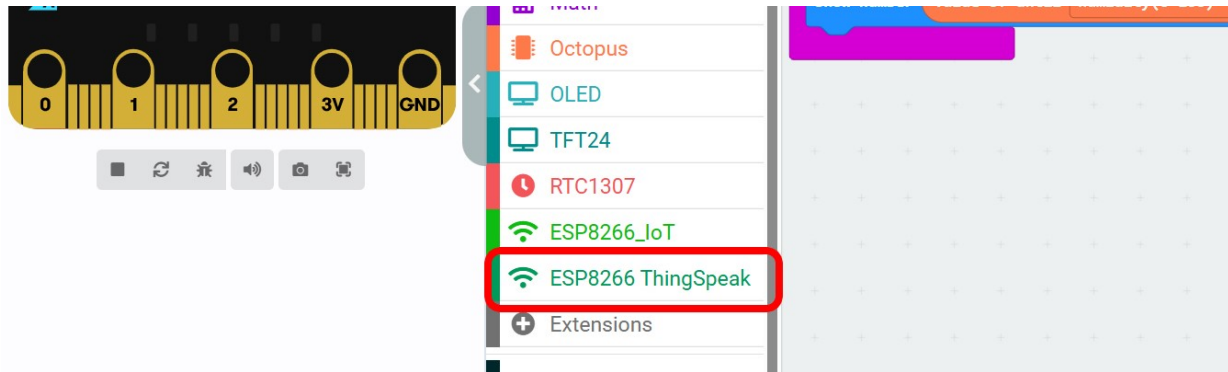


Figure 11: The ESP8266 group of commands

Important note: The “`iot_environment_kit`” extension already contained a group of commands called “`ESP8266_IoT`”. However, the blocks contained therein are not reliable enough for programming this specific module.

From the basic menu, drag and drop the “**On start**” hat block onto the script assembly area, then place the “**Initialize ESP8266...**” command inside it. Then set the RX and TX pins and leave the baud rate at 115200. Enter the name of your WiFi connection and its password in the Wifi SSID and Wifi PW fields, respectively.

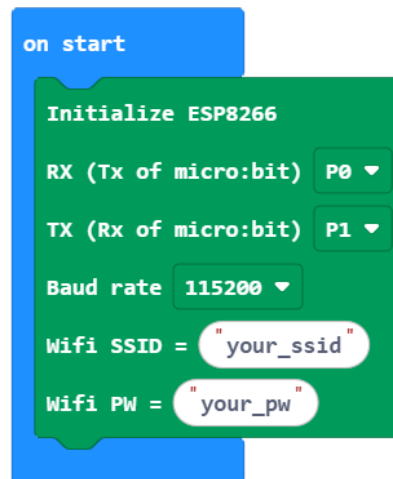


Figure 12: Initializing the ESP8266 module

From the basic menu, drag and drop the “**forever**” loop onto the script assembly area. Then place the “**upload data to ThingSpeak ...**” command inside it. Leave the URL/IP address as it is. In the Write API key field, you will add the corresponding key that will be generated on the ThinkSpeak IoT platform, once you have set up your channel (see the next section for more details). In Field 1, add the “**value of dht11 temperature at pin P2**” command to send the temperature measurements, and in Field 2, add the “**value of dht11 humidity at pin P2**” command to send the humidity measurements. Finally, add a “**pause**”

command to the end of the script and set it to at least 5000ms to allow for a smooth transfer of data to the ThingSpeak platform.

Note: As mentioned, the DHT11 sensor may require some time to initialize and begin transmitting data. Therefore, for debugging purposes, you can also add the “light level” measurement (found in the input menu) or any other measurement related to a built-in sensor, to ensure that the connection between the micro:bit and the ThingSpeak platform has been established.

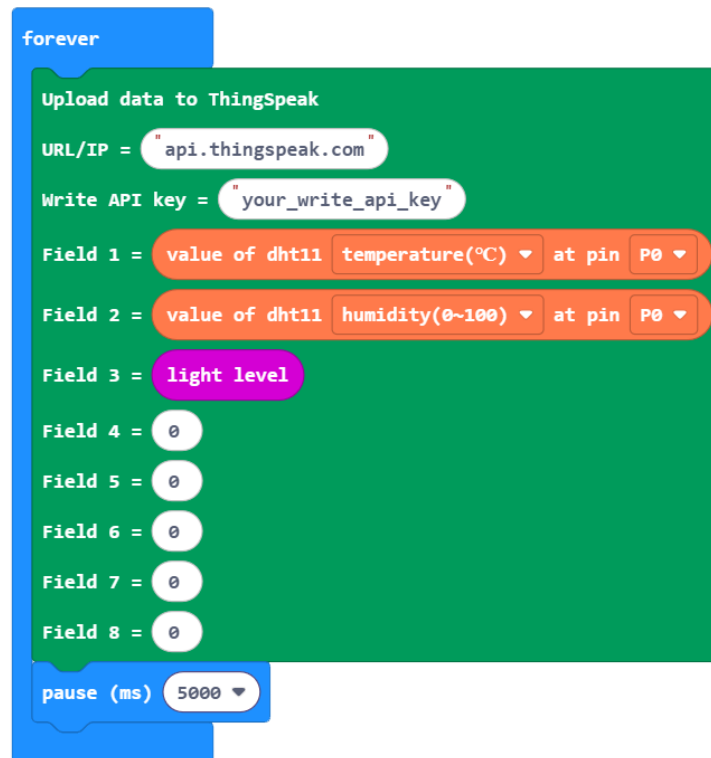


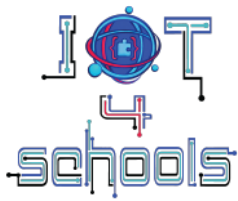
Figure 13: The part of the script for sending measurements to the ThingSpeak IoT platform

The script is almost finished. Do not download it to the micro:bit yet, as you first need to create a channel on the ThingSpeak IoT platform.

The ThingSpeak IoT platform

ThingSpeak is an IoT analytics platform service that enables the aggregation, visualization and analysis of live data streams in the cloud. This can be achieved by creating a channel connected to a device (e.g. the weather station) or an application to display transmitted data. To create a channel, you should first create an account on the ThingSpeak website (<https://thingspeak.com/>) using your email address and setting a password.

Note: The free version of ThingSpeak is reliable, but limited. Each account can have up to four active channels. Therefore, if this number is exceeded, you will need to delete one of the older channels, to create a new one. Since you are working with students, consider of setting up accounts for them.



After signing in to your account, go to the Channels menu and click the “New Channel” button under “My Channels” to create a new channel to receive and display/visualize data from the sensors (Figure 14).

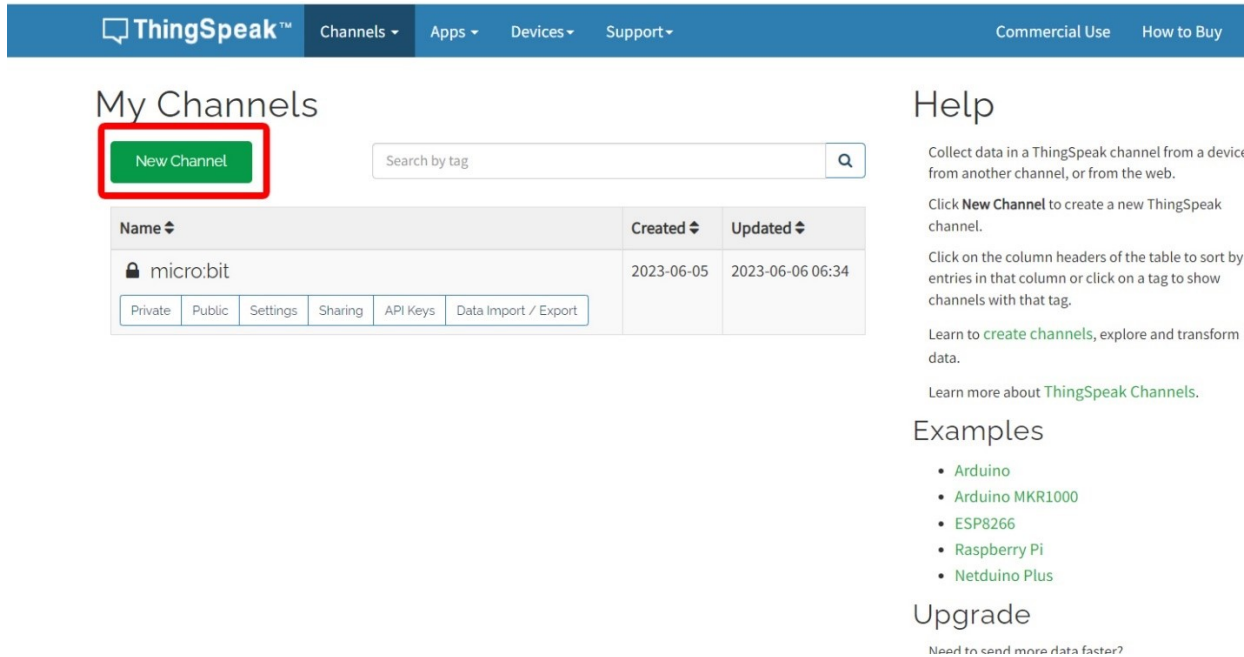


Figure 14: Click on the New Channel button, to create a new channel

The next step is to set your New Channel. Inserting the name (1.) of your channel (e.g., weather station) and the parameters that you want to monitor in the Fields provided (2.) (Figure 15). Specifically, insert temperature in Field 1 and humidity in Field 2. You can activate more fields by clicking the checkbox (3.), next to a Field. You can also optionally write a description to help you remember which parameters your channel is monitoring.

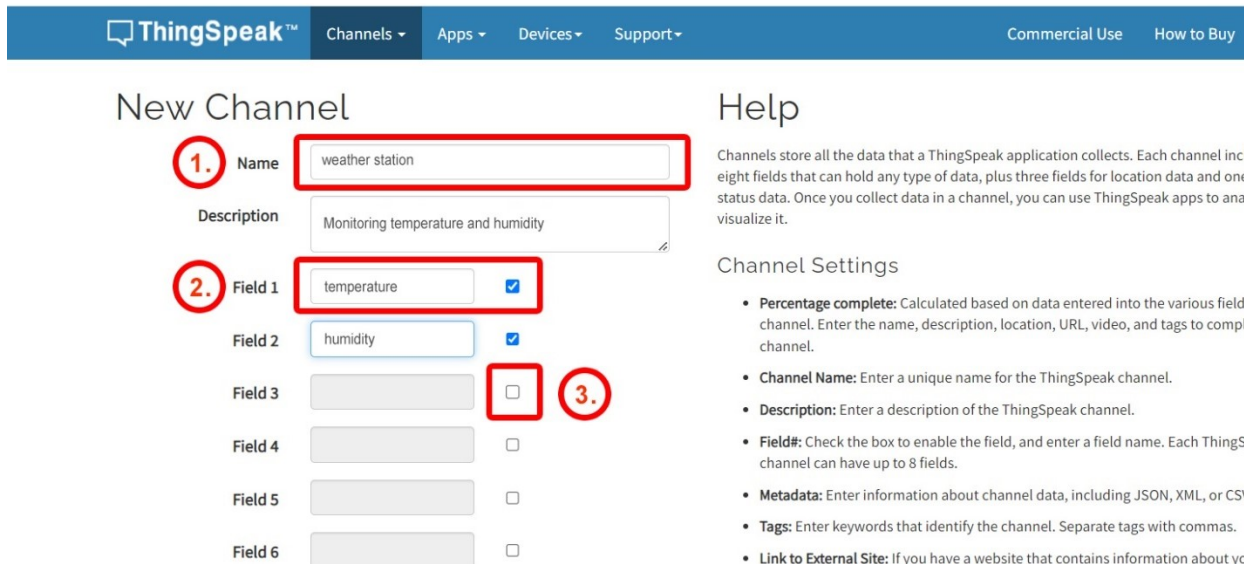
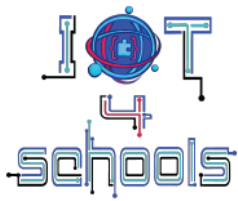


Figure 15: Setting up your channel



Once you have set all the parameters that you want to monitor, scroll down the New Channel page and click the “Save Channel” button (Figure 16), to save all the information that you have entered.

Figure 16: Saving your channel

Important note: You can always revisit your channel and make any changes you want. However, bear in mind that you should always save these changes, before exiting the channel tab.

After pressing Save Channel, you will be automatically redirected to your New Channel. Click on the API Keys menu (1.) (Figure 17). Write down the Key (2.) that appears in the Write API Key field, or copy and paste it into the corresponding field of the “upload data to ThingSpeak ...” command (Figure 13).

Figure 17: The API keys menu

Go to the Makecode (do not close the ThingSpeak) and enter the API key into the script. Then **download** the entire script to your micro:bit. Return to the ThingSpeak platform, and click on Private View (*Figure 18*). After a short wait, you will see the graphs on the Channel Stats change to visualize the transmitted measurements. You can then monitor any changes in their values in real time.

Note: You can find out more about modifying the appearance of the channel stats in the Appendix (subsection 5.2).

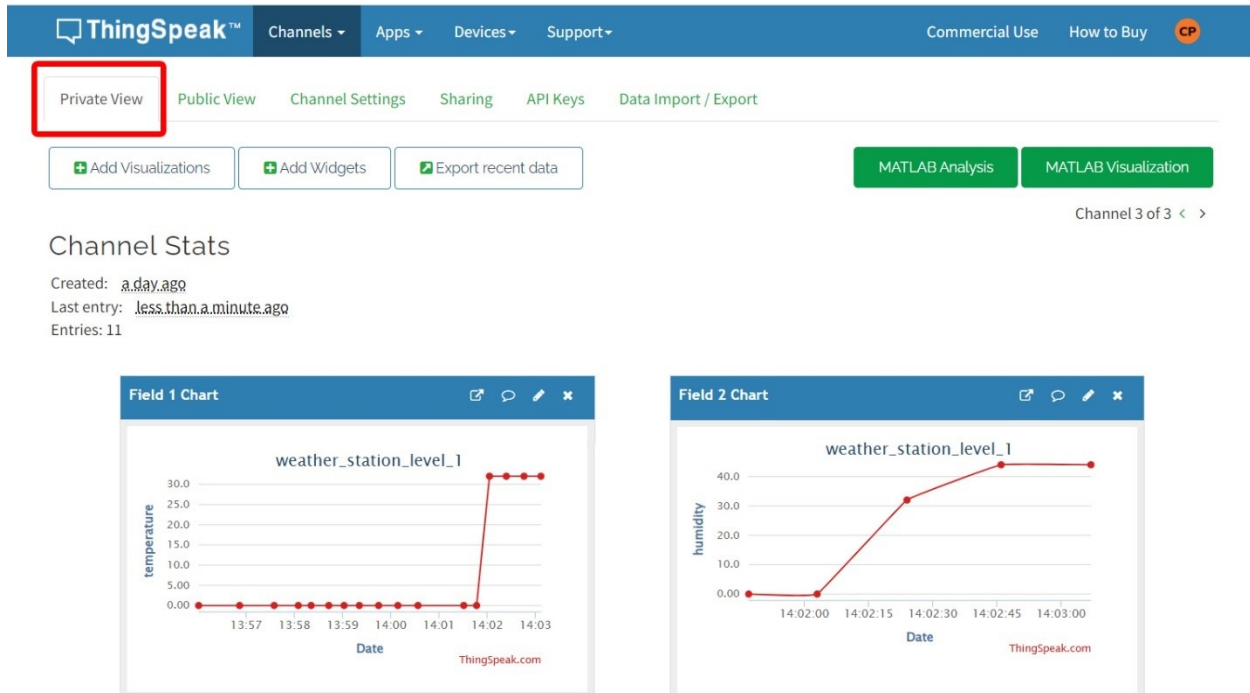


Figure 18: Checking the channel stats in the private view

Ideas for encouraging meaningful student engagement with data collected from the weather station

Below are some ideas to help students understand the importance of collecting data from a weather station:

1. Ask your students to monitor the data for as long as possible (ideally one day) and analyse the results. Encourage them to describe the (daily) cycle of temperature and humidity, highlighting the warmest and the coolest periods of the day and the times of maximum and minimum humidity. Then ask them if they can explain why this happens and whether they can detect any correlation between temperature and humidity changes. You could also encourage them to make some predictions (prior to measurements) about when they think the temperature or/and humidity will be higher or lower.
2. Ask your students to compare the data from their station with data from an official weather station and encourage them to identify any discrepancies in the measurements.
3. Ask them to reflect on the design of the ThingSpeak dashboard. Is the depicted data easy for them to understand? What would they like to improve? This will encourage them to think of different ways to visualize the measured data.

4. Ask them to consider how temperature or humidity could affect human comfort, energy use or school activities. You could also encourage them to keep a mood diary and observe whether there is any correlation between temperature/humidity measurements and their mood, energy levels or behaviour.

2.1.3 Crafting

Encourage your students to create a model (paper-based on any other material they want) for embedding the circuitry of their weather station. Crafting is a rather free process, based mostly on your student's inspiration, intuition and crafting skills, and it is up to them what type of model they create and how far they want to go. It is certainly a creative process involving mathematical/geometrical and, to some extent, engineering processes, and it is closely linked to problem solving.

In this sense, the following instructions are indicative and demonstrate how a paper model, such as the one depicted in Figure 19, can be created.

In this model, the micro:bit with the edge connector have been placed vertically. A rectangular hole has been cut in the cardboard to make the micro:bit's LED display and two buttons visible and accessible. The horizontal part of the model provides space for the jumpers that connect the DHT11 sensor and the WiFi module to the edge connector. This part of the model has been designed to hide the wires while firmly holding the two components (DHT and WiFi) on the exterior of the model, while they are connected to the edge connector.

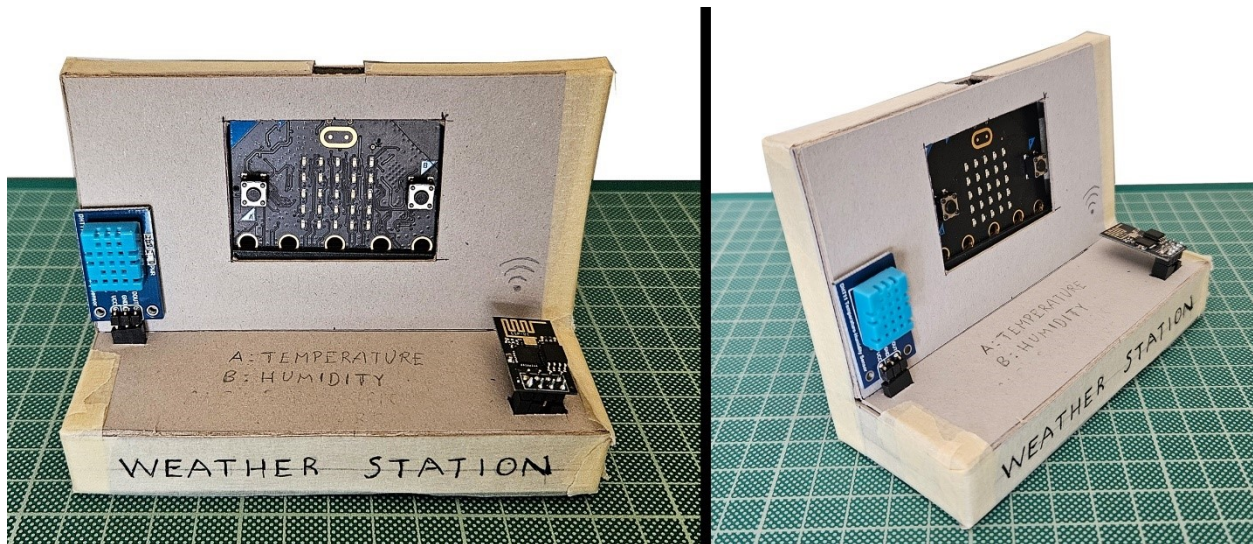


Figure 19: An indicative paper model of the weather station, created from cardboard

Start by measuring the following electronic components: a) the micro:bit clipped to the edge connector, b) the height of the headers of the jumpers when connected to the edge connector, c) the space required to connecting the jumpers to the DHT11 and the WiFi component headers.

Then try sketching the outline of the paper model onto a piece of paper. The net of the model shown here looks like the drawing in Figure 20. The carved parts will enable the net to be folded into the shape of the box that will envelop/enclose the circuit.

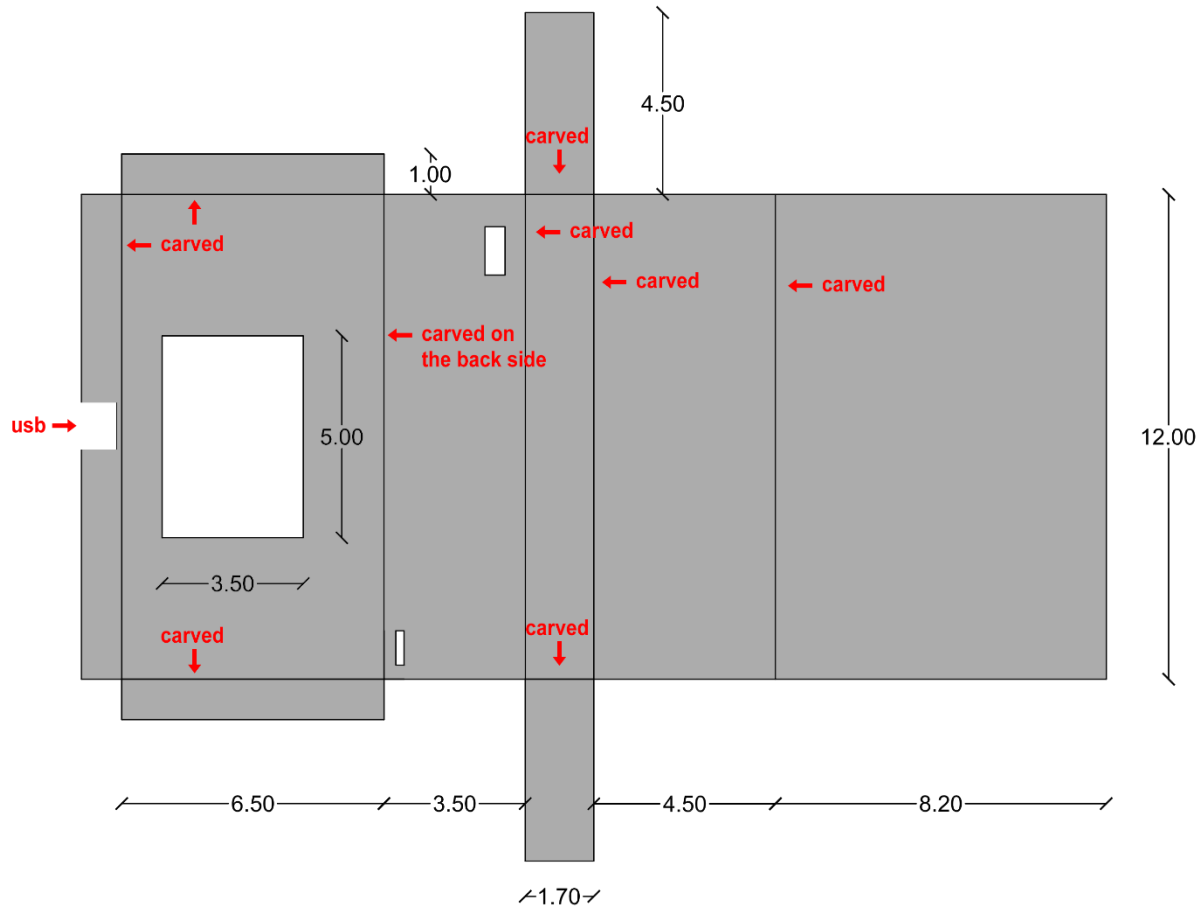


Figure 20: The net of the indicative paper model

The next step is to replicate the net on a cardboard. The result can look like Figure 21.

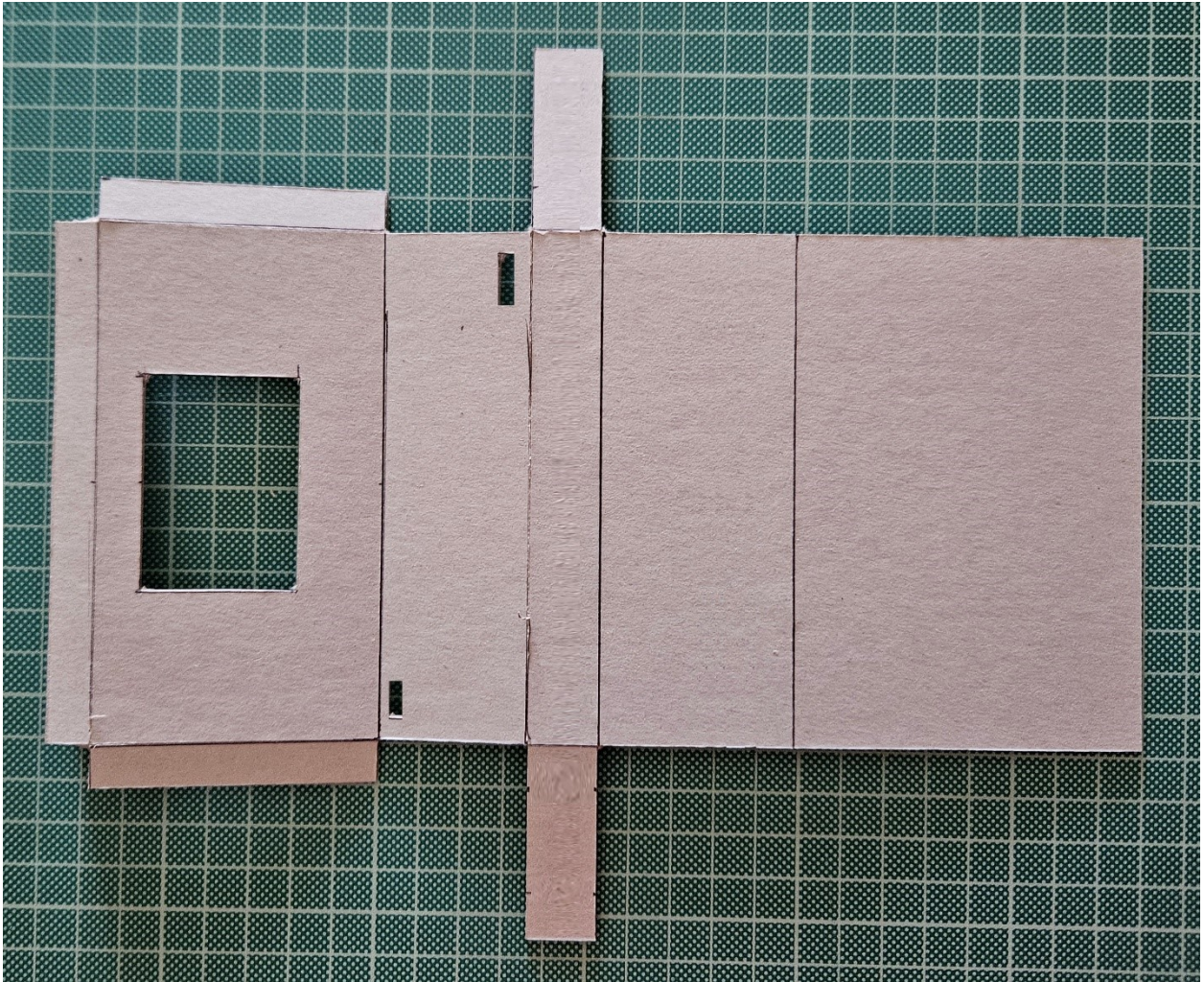


Figure 21: The net of the model replicated on a cardboard

Fold up all the sections of the net to enclose the circuitry. Use glue or paper tape to bind the paper model firmly together. Figures 22 and 23 present some steps of the process.

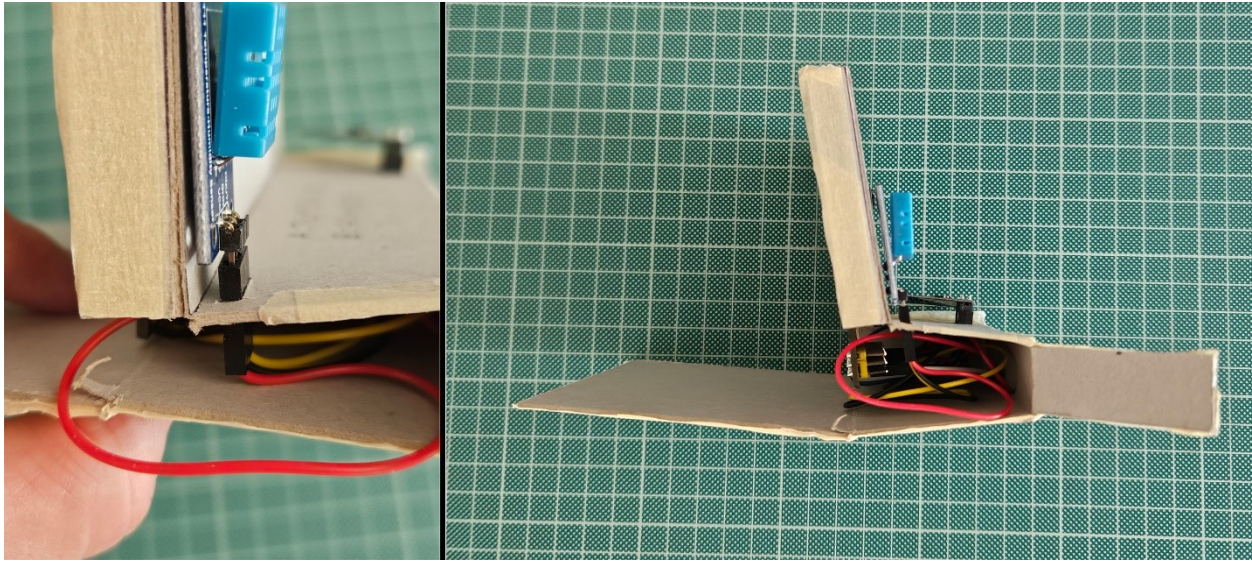


Figure 22: The net is folded up to create the space for hiding the jumpers. A detail of how the DHT11 sensor is embedded into the model is shown on the left.

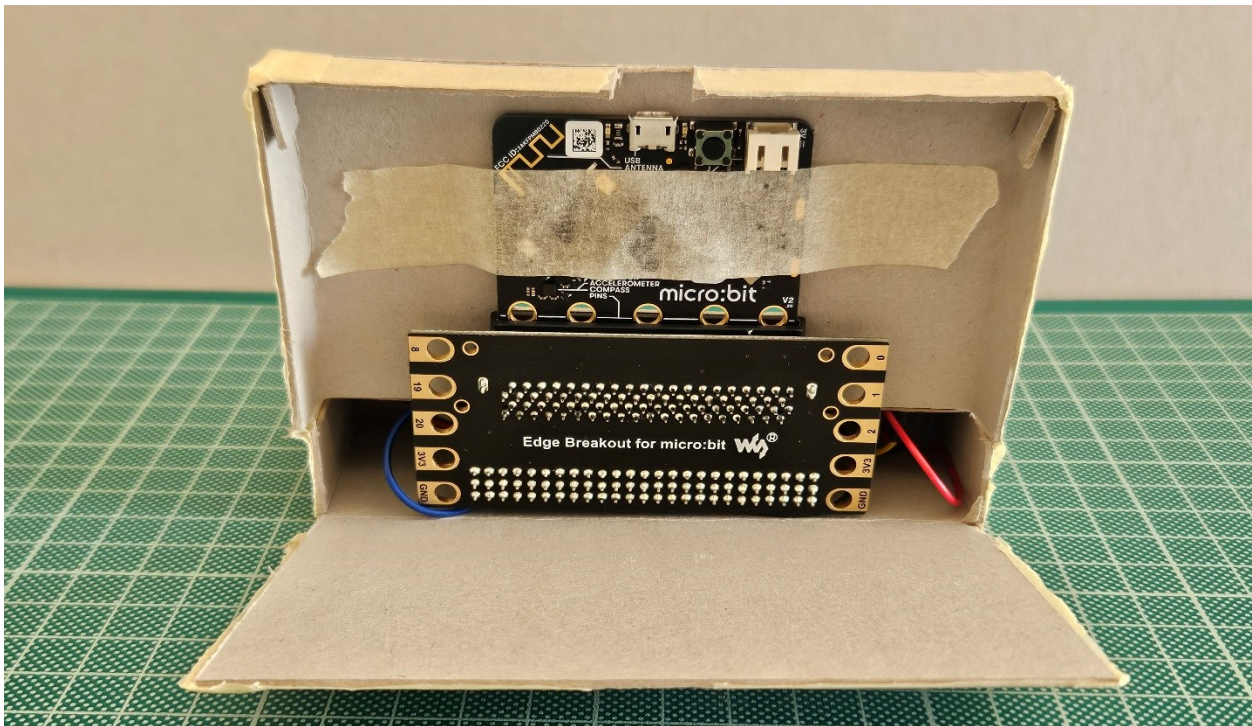


Figure 23: Using a paper tape to stabilize the micro:bit, before folding up the final piece of the net

Use a marker or pencil to decorate the model (e.g., note what the buttons do, what information is displayed on the LED screen, add the WiFi icon etc.). The Weather Station paper model is now complete (Figure 19).

Note: In this solution, you need to add two commands to the scrip to enable Button A and Button B to display the temperature and the humidity respectively.

2.2 Level 2: Adding a sensor for measuring barometric pressure

At this level, students will also enable their weather station to measure barometric pressure. To do so, they will first add the BMP280 barometric pressure sensor to the circuit from the previous level. They will then program the sensor to receive the measured values and send the data to the ThingSpeak platform.

Before starting the circuit making process, ensure your students understand what barometric pressure is and how it is related to weather changes. You can encourage them to search online to find out more about this parameter. You can also pose some of the following questions to start a dialogue:

- Have you ever traveled by airplane or driven up a mountain? If so, have you ever noticed your ears “popping”? What do you think causes that feeling?
- What do you think would happen to an empty plastic bottle if you put it in a freezer? What if you left it in a car on a sunny day?
- Why do weather forecasters talk about “high pressure” and “low pressure” systems? How are these terms related to whether it is a sunny or stormy day?
- Can we “see” the weight of air? How could we prove that air has mass and exerts pressure?
- If we lived on a planet with a very thin atmosphere, like Mars, what would that be like?

2.2.1 Circuit making process

Figure 24 shows how the BMP280 barometric pressure sensor is connected to the edge-connector breakout board. 4 F-F jumpers are required to make the following connections: VCC to 3V3 (1), GND to GND (2), SCL to pin 19 (or SCL pin) (3) and SDA to pin 20 (or SDA pin) (4). The sensor is now ready to be programmed.

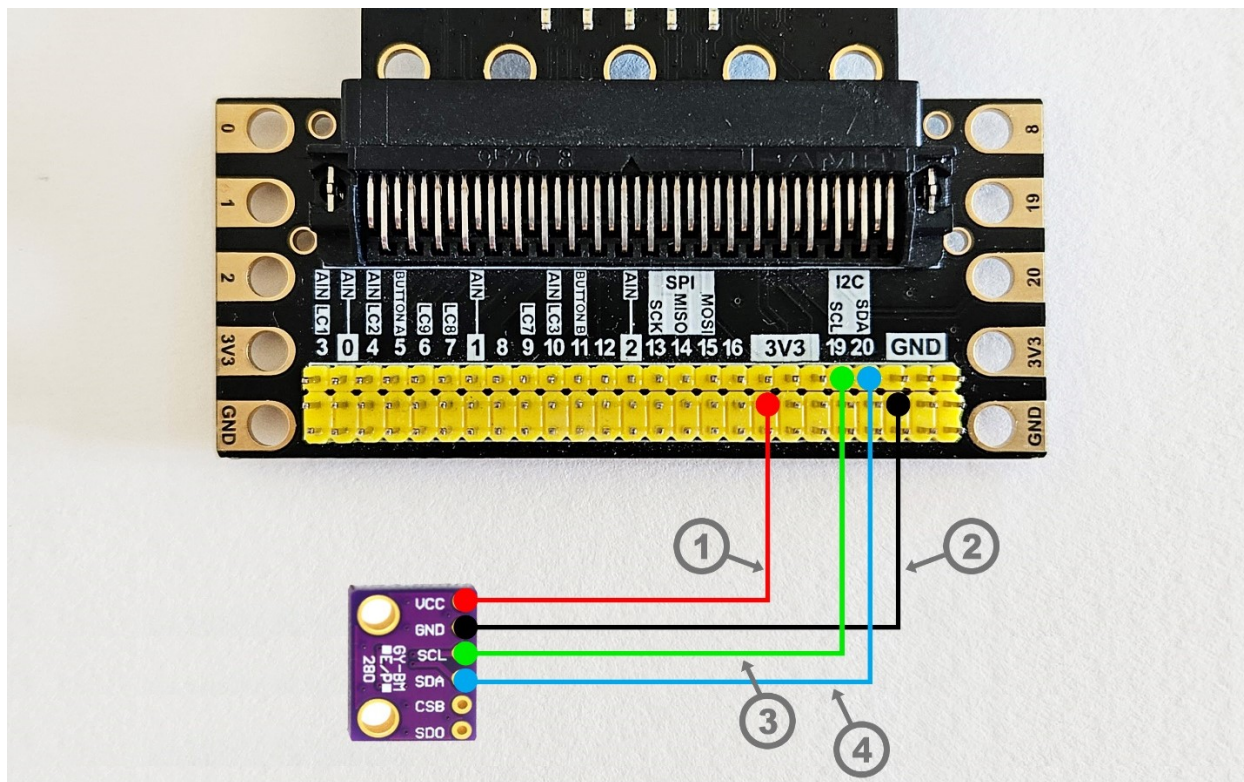


Figure 24: Adding the BMP280 barometric pressure sensor

Important Note 1: To simplify the diagram, the DHT11 sensor and the ESP8266 WiFi module, are not included in this figure. During the circuit-making process, do not disconnect these sensors. You only need to add the BMP280 sensor.

Important Note 2: Bear in mind that the barometric pressure may not have the pins pre-attached. Therefore, you may need to solder the pin headers in place.

2.2.2 Programming

Based on the time available, and your students' level, you may wish to encourage them to do the warm-up activity described in the Appendix (subsection 5.3), to become familiar with the BMP280 sensor and the received measurements.

Makecode programming part

Continue working on the previous file, but first save it as a new file. Then, go to Extensions, find the “envirobit” extension and add it (*Figure 25*).

Note: There are other extensions available for programming the sensor. However, this sensor is more stable when using the commands contained in the “envirobit” extension.

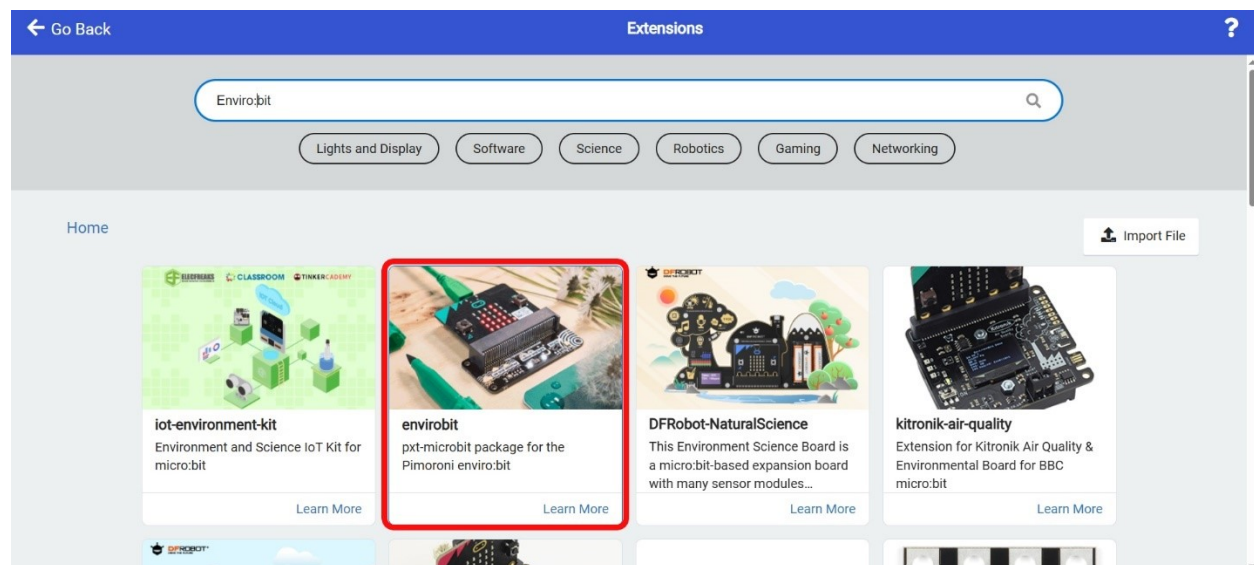


Figure 25: Finding the envirobit extension

Click on the **Enviro:Bit** group of commands (1), then click on the “...Expert” menu (2) to find the “**Get pressure (decimal)**” block of command (*Figure 26*).

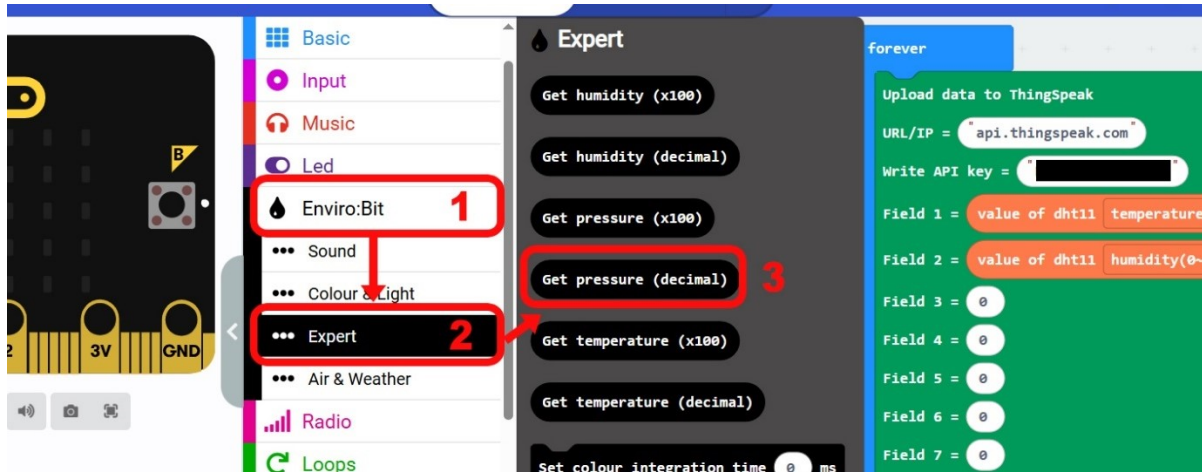


Figure 26: Finding the Get pressure decimal command

Important note: The BMP280 measures the barometric pressure in hPa (hectopascals). The standard reference point for measuring the air pressure is sea level. At sea level, the standard atmospheric pressure is typically 1013.25 hPa. The “Get Pressure (decimal)” command adds the necessary comma to the value so that it is displayed in a meaningful way.

Drag and drop the “**Get pressure (decimal)**” command to **Field 3** of the “**Upload data to ThingSpeak**” command. The script is now ready to be downloaded to the micro:bit (Figure 27). Before doing so, make sure that you are still logged in to the ThingSpeak IoT platform, and that you are synchronized with your channel. Also, ensure that you have activated the 3rd field of your channel (Figure 15) to enable the barometric pressure sensor’s measurements to be displayed.

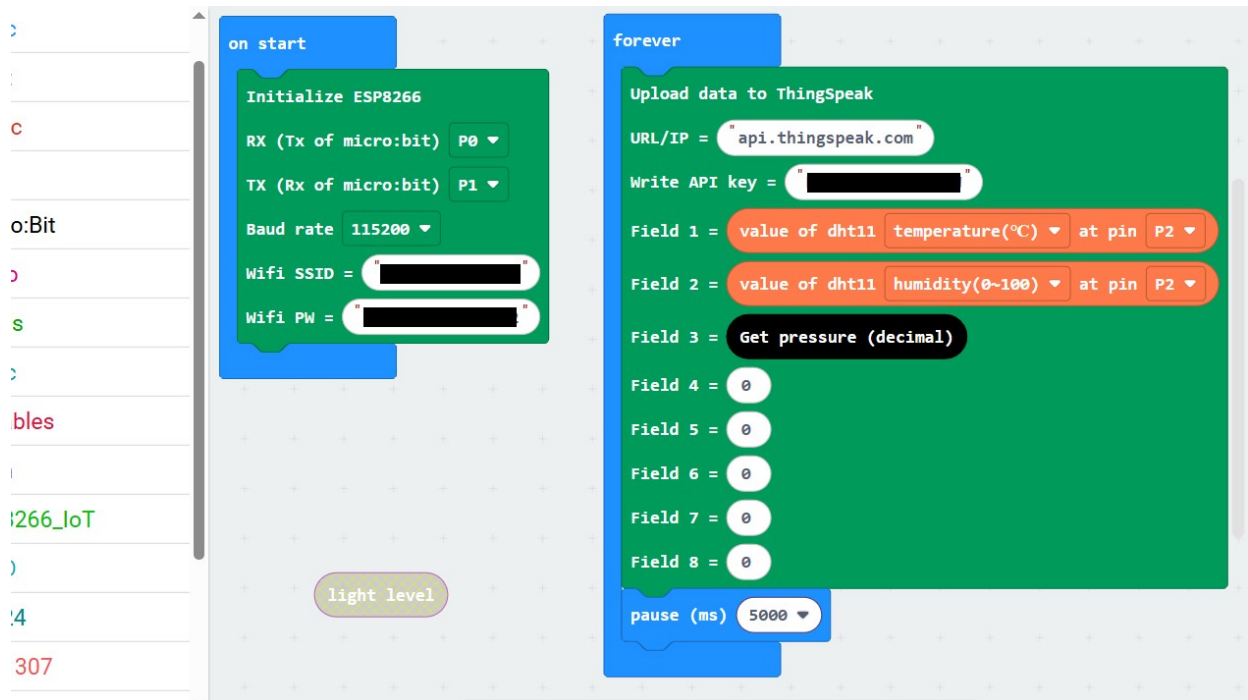
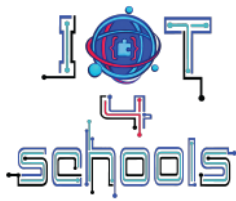


Figure 27: The final script of Level 2



After downloading the script, return to the ThingSpeak platform, and click on Private View of the channel (*Figure 18*). After a short wait, you will see the graphs on the Channel Stats change to visualize all the three transmitted measurements. You can then monitor any changes in their values in real time.

Ideas for encouraging meaningful student engagement with data collected from the weather station

The following ideas are intended to help students understand the importance of collecting barometric pressure measurements from a weather station:

1. As before, ask your students to monitor the data for as long as possible (ideally one or two days) and analyse the results. Encourage them to describe the daily cycle of barometric pressure, highlighting any changes in air pressure during the day and identify any patterns in these changes. Ask them to explain why this happens and whether they can detect any correlation between temperature, humidity and barometric pressure changes. You could also encourage them to predict any changes to the weather.
2. Ask your students to compare the data from their station with data from an official weather station, encouraging them to identify any discrepancies in the measurements.
3. Encourage the teams to cooperate by asking one team to place their weather station indoors and the other to place theirs outdoors in a sheltered area), and monitor the data for as long as possible. Ask them to compare the results and consider why there are differences (if any).
4. Encourage your students to keep their weather station running continuously, especially when a change in the weather is expected, to observe any significant changes in barometric pressure, thus realizing how barometric pressure can help predict the weather.

2.2.3 Crafting

Make any necessary changes to your model to wisely embed the BMP280 barometric pressure sensor.

3 Tips and recommendations

3.1 Further expansion of the project

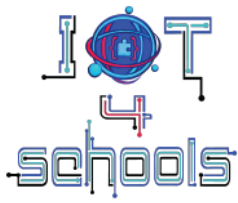
Depending on the level of your students and the time available, you can extend the project to measure more parameters. This means that you need to add more sensors (e.g. CO2 sensor, dust sensor) and activate more fields to the ThingSpeak channel.

3.2 Creating an autonomous weather station

You can create an autonomous weather station (i.e., disconnected from the computer) by adding an external power source (e.g., using a battery holder or power bank). This allows you to move the weather station to different locations inside the school building. However, bear in mind that this solution may not provide many hours of autonomy.

3.3 Optimizing the appearance of the device

You can encourage your students to add an LCD I2C screen to optimize the appearance of the weather station. Instead of using the micro:bit's LED display (which is not ideal for displaying large values), you can consider adding an LCD screen to the circuitry to display the results more clearly. In this case, you will also



need to modify the paper model of the weather station. You can find out how to connect and program an LCD I2C screen in the Appendix (subsection 5.4).

3.4 Discussing the advantages and disadvantages

After completing the project, encourage your students to share their thoughts and ideas about the advantages and disadvantages of using devices such as a weather station to make decisions about different aspects of life. You could also ask them what they think about sharing such data on the cloud and whether they think there are any risks in terms of personal data and security.

4 References

[1] Microsoft Makecode software: <https://makecode.microbit.org/>

[2] ThingSpeak IoT platform: <https://thingspeak.com/>

5 Appendix

5.1 Glossary – Definitions of key concepts

Temperature: the degree or intensity of heat present in a substance or object, especially as expressed according to a comparative scale and shown by a thermometer or perceived by touch. (Oxford Languages dictionary). It is measured in Celsius ($^{\circ}\text{C}$), Fahrenheit ($^{\circ}\text{F}$) or Kelvin (K).

Humidity: the state or quality of being humid/ a quantity representing the amount of water vapour in the atmosphere or in a gas/ atmospheric moisture (Oxford Languages dictionary). Absolute humidity is typically measured in grams per cubic meter (g/m^3) or grams per kilogram (g/kg). Relative humidity is expressed as a percentage.

Barometric pressure: the measurement of air pressure in the atmosphere, specifically the measurement of the weight exerted by air molecules at a given point on Earth (<https://www.setra.com/blog/what-is-barometric-pressure>). It can be measured in several units including Pascals (Pa), atmospheres (atm) and millibars (mb).

5.2 ThingSpeak Channel Stats options

Click on the pen icon located in the field headline to modify parameters related to the appearance of the chart (*Figure 28*).

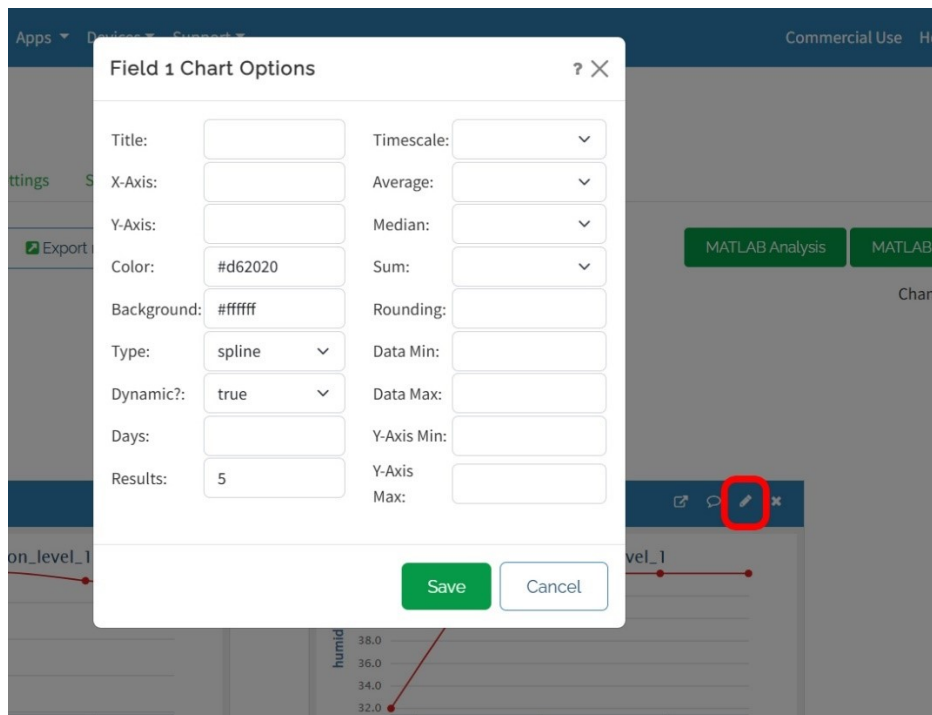


Figure 28: Accessing the chart options window

Click on the “Add Widgets” to make graphically improve the charts (e.g., add a gauge icon to visualize the temperature data in a more perceptible/tangible way) (*Figure 29*).

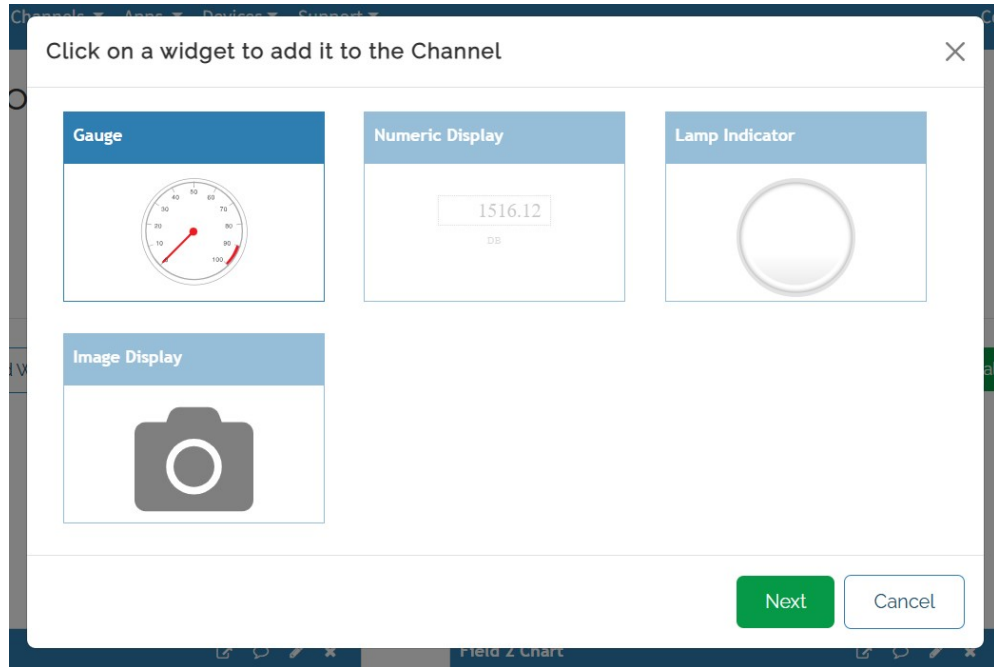


Figure 29: Widget options

Click on the “Export recent data” to export some or all of the data as a JSON, XML, or CSV files (Figure 30).

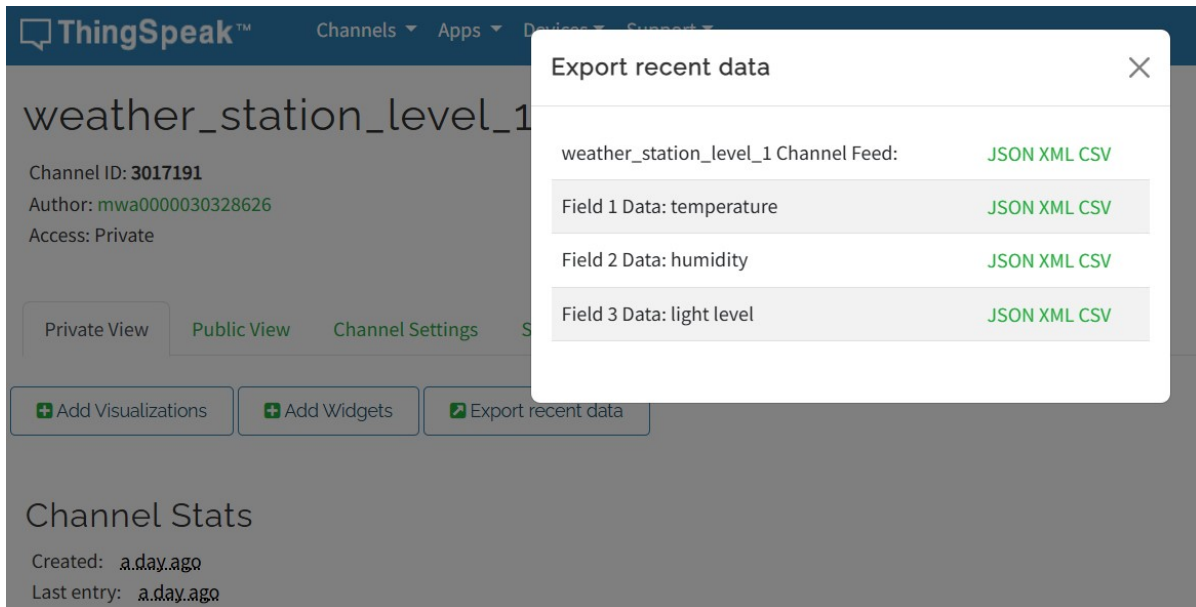


Figure 30: Export recent data window

5.3 Barometric pressure warm-up activity

This warm up activity will enable your students to experiment with the BMP280 barometric pressure sensor before transmitting the measurements to the ThingSpeak platform.

After connecting the sensor to the micro:bit via the edge connector (see *Figure 24*) and importing the “envirobit” extension, create a new project in Makecode. In the **forever** loop, add the **show number** command from the **Basic** menu, and insert the **“Get pressure decimal”** command into the value field. Then add a 1-minute **pause** (i.e., 60000 milliseconds), and download the script to the micro:bit.



Then encourage your students to move the weather station to different locations (and different floors) around the school, and observe if there are any changes.

5.4 Adding and programming an LCD I2C screen

The following instructions are for a 16x2 LCD screen with an I2C interface module. The I2C interface simplifies the circuitry, as only four pins need to be connected to the edge connector, namely: the GND to GND (2), the Vcc to 3V3 (1), the SDA to pin 19 (3) and the SCL to pin 20 (4) (*Figure 31*).

The screen also has a potentiometer for adjusting the brightness and contrast. If you cannot see anything displayed after downloading a script, use a screwdriver to make the necessary adjustments.

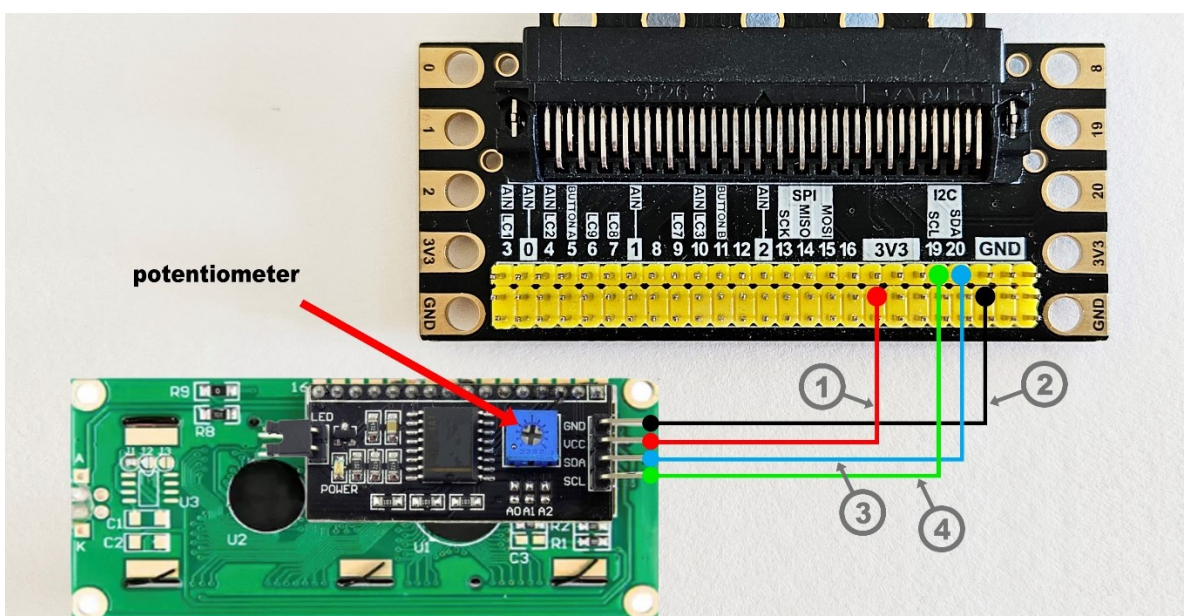


Figure 31: The circuitry for connecting the LCD screen

To program the screen you need to download the “makerbit-lcd1602” extension (Figure 32).

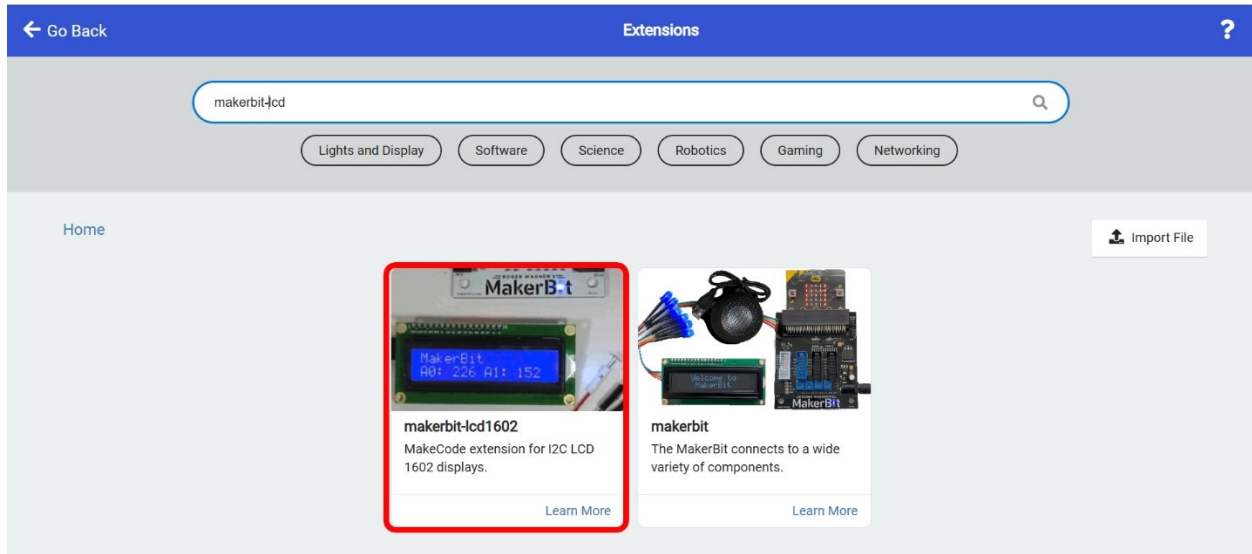


Figure 32: Finding the “makerbit-lcd1602” extension

The LCD screen consists of a total of 32 characters, divided into 2 lines (16 characters per line). To display something (a text/string or a value/number), you need to use the “**LCD1602 show ...at position...with length...**”, command (Figure 33).

In the “show” field (1), enter the string or number you want to display (e.g., temperature). In the “position” field (2) you need to specify where this string or value will start to be displayed. In the “length” field (3) you need to specify the character at which the display should end.

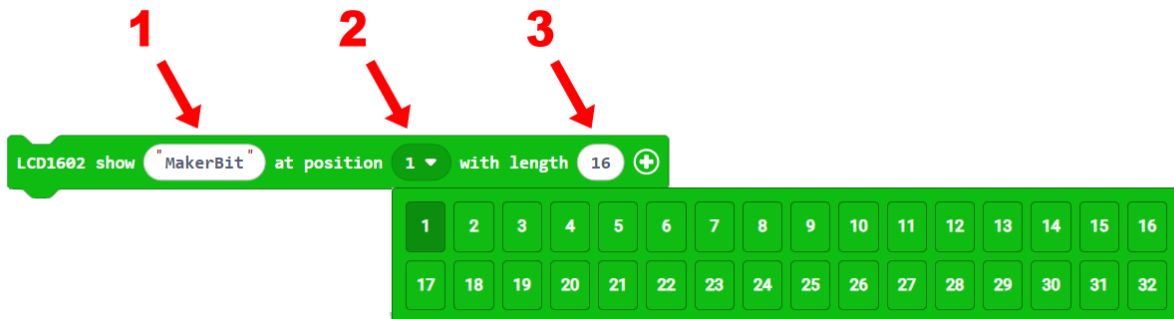


Figure 33: The “LCD1602 show ...at position...with length...”, command

The following scripts are indicative. They show how the screen can be programmed to display only “temperature” and “humidity” (Figure 34), or all the three values (Figure 35).

```
forever
LCD1602 show "temperature" at position 1 with length 11
LCD1602 show value of dht11 temperature(°C) at pin P2 at position 13 with length 16
LCD1602 show "humidity" at position 17 with length 11
LCD1602 show value of dht11 humidity(0~100) at pin P2 at position 29 with length 16
```

Figure 34: Indicative script for displaying temperature and humidity

```
forever
LCD1602 show "Tem" at position 1 with length 3
LCD1602 show value of dht11 temperature(°C) at pin P2 at position 5 with length 8
LCD1602 show "Hum" at position 9 with length 11
LCD1602 show value of dht11 humidity(0~100) at pin P2 at position 11 with length 16
LCD1602 show "Pressure" at position 17 with length 25
LCD1602 show Get pressure (decimal) at position 27 with length 32
```

Figure 35: Indicative script for displaying temperature, humidity, and barometric pressure