Multi-sensor temperature

Difficulty 🟡

Required materials

- 2 EMDACC4 with the Temperature + Humidity sensor
- 1 EMDACC4 with the Magnetic sensor
- 1 magnet and some tape to keep it in place
- 1 Micro:bit per group
- 1 Computer per group

Prerequisites

- Basic knowledge of block languages
- Program the central EMDACC4 devices with the last firmware

1. Deploy the central EMDACC4

To begin, each central device needs to be programmed with the latest firmware.

Once this is done, the Micro:bits can be attached to their respective boards. One board will be placed inside the classroom to measure the internal temperature, another one outside the window to capture the outdoor conditions, and the third will be attached to the magnet sensor that monitors whether the window is open or closed. When powered on, each device shows a confirmation symbol on the display: “°C” for the temperature boards and a lock symbol for the magnet board. This means they are properly connected and already broadcasting their readings over the radio channel.

At this point, the sensors can be fixed in place: the outside temperature sensor should be positioned where it is sheltered from direct sun and rain, and the magnet module should be attached securely to the window frame together with the small magnet.

2. Connect the Micro:bit to the computer

Turn on the computer and open a web browser (Chrome is recommended).

Go to the MakeCode website (https://makecode.microbit.org/), create a new project, and give it a meaningful name, for instance **“Multi-sensor data”**.

Plug the Micro:bit to the USB port of the computer and pair it with the MakeCode editor.

Write a simple plotting program to see which are the values that the sensors are broadcasting.

Pay attention to the names of the device, this time we will need to distinguish from which sensors we are receiving the data!

3. Design the algorithm

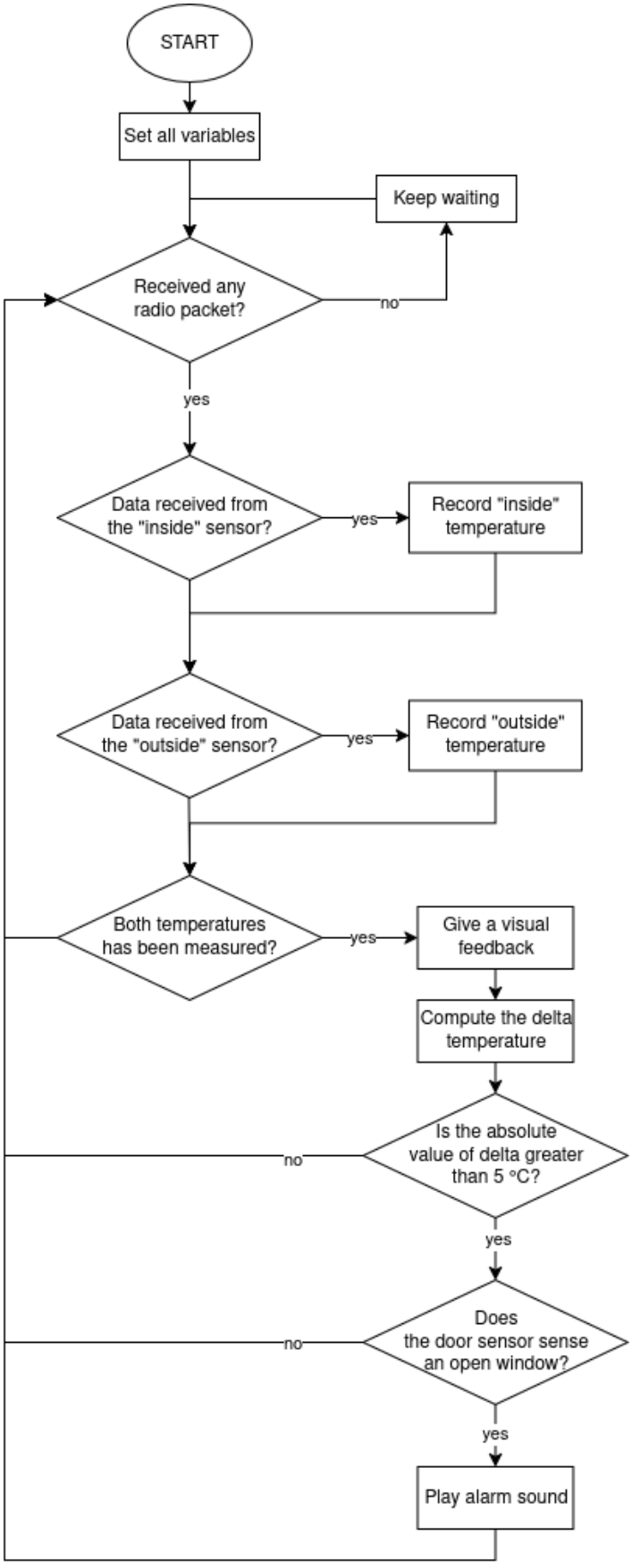
The idea behind this activity is to alert users when a window is left open while there is a significant temperature difference between inside and outside, since this would lead to unnecessary energy waste.

To achieve this, the program running on the peripheral Micro:bit must continuously receive data from the three central devices. The latest values of the indoor and outdoor temperature are stored and compared to calculate the difference between them.

If the window is reported as open and the temperature difference is greater than a chosen threshold (for example 5 °C), the device should trigger an alarm. If the window is closed or the difference is small, no action is needed.

This type of reasoning mirrors real-world energy-saving logic and gives a clear example of how sensors can be combined to produce useful outcomes.

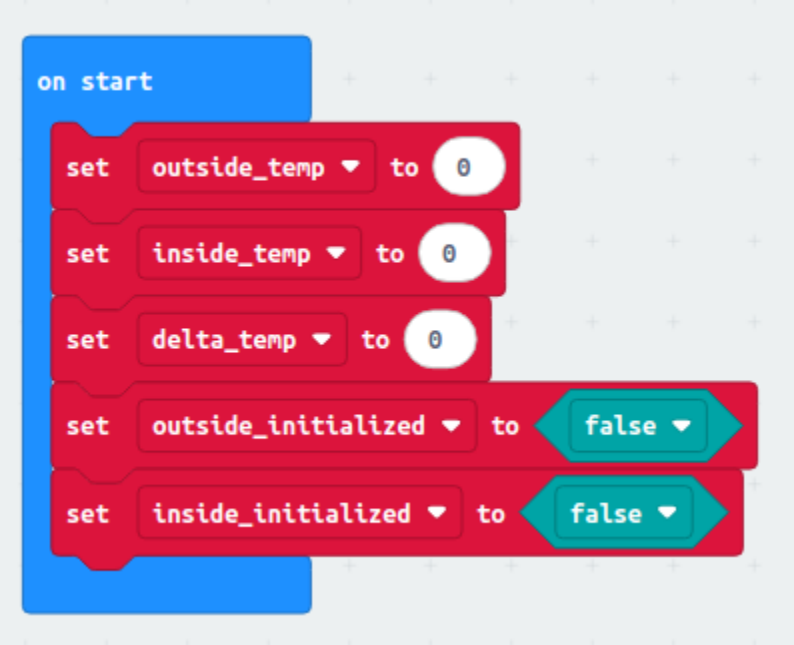
We can sketch out a flowchart of our desired behaviour:


START
Set all variables
Keep waiting
Received any radio packet?
no
yes
Data received from the "inside" sensor?
yes
Record "inside" temperature
Data received from the "outside" sensor?
yes
Record "outside" temperature
Both temperatures has been measured?
yes
Give a visual feedback
Compute the delta temperature
Is the absolute value of delta greater than 5 °C?
no
yes
Does the door sensor sense an open window?
no
yes
Play alarm sound

4. Program the Micro:bit

Now it is time to start composing the code. First of all it is possible to notice that we need many variables: one for the internal, one for the external temperature, other two to check if the temperature has been initialized (to avoid computation problem in case one is initialized and the other no) and, of course, the variable to store the difference in temperature. Such type of data is often addressed as “delta” and usually is the absolute value of the difference between the two values and it is used to measure how far the values are.

To begin with we can initialize all the needed variables “on start”:



The whole code, then, runs every time we receive a data packet from one of our sensors so we need the block “on radio received <name> <value>”.

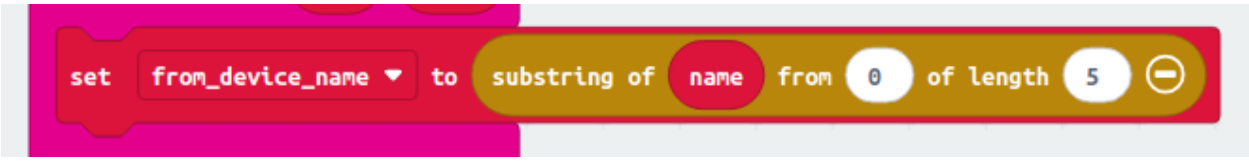
Now we need to make a consideration: our code will be executed in case of temperature packet, door status packet but also humidity. In addition to this complexity we need to address the fact that there will be two temperature sensors, one for the inside and the other for the outside.

To tackle these new problems we can write a simple program to log the data so we can make controlled experiments and understand the name of the source of our packet. In particular, keep in mind that the sources are named with 5 random

characters followed by an underscore and a reference to the type of sensor (TE in case of Temperature).



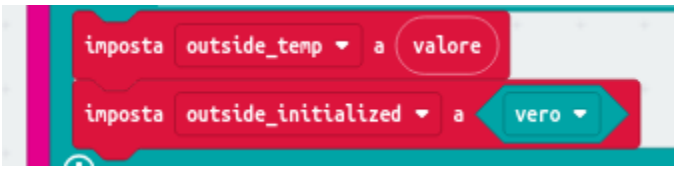
Once we have the 5-character names of the internal sensor and the external sensors, we can continue coding our algorithm. Could be useful to save it into a variable to be a bit more tidy.



Every time we receive a packed we need to identify from which device it is coming from and, since we could retrieve the names, now it is matter of doing a string comparison (Advanced menu > Text, take the “<this> includes <>” block) inside an “if” block:



Inside each “if” block (one per temperature sensor), we assign the measured value to the corresponding variable and we set it as initialized:

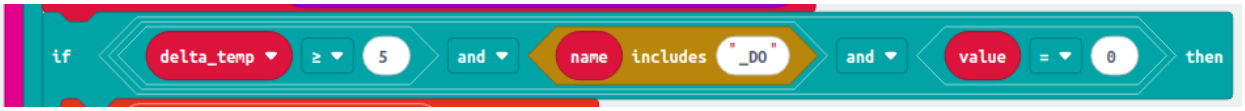


If both the sensors have been correctly initialized we can give a visual feedback to signal that the device is now ready to monitor the status of the window.

At this point we need to implement the formula to compute “delta_temp” as described above:



And then we need an “if” block to check if we are above the chosen threshold (5 °C) and if the window is actually open (the Door sensor is identified by “DO” in the name):



We just need to add the sound to play if the conditions are met and we are done!

Ideas

1. Make the alert visual (LEDs) instead of sound.
2. Log “delta_temp” over time via serial and plot it to see patterns during the day.
3. Add a timer: only trigger the alarm if the window is open for more than 1 minute while the temperature difference is high.
4. Explore energy savings: calculate the potential heat loss if the window stayed open.

Reference code

