Measuring Energy Consumption

Difficulty 🟡

Required materials

- 1 EMDACC4 with the Current sensor board
- 2–3 household appliances to test (lamp, fan, phone charger, laptop, …)
- 1 Micro:bit per group
- 1 Computer per group

Prerequisites

- Basics of electricity
- Being able to plot data with the MakeCode platform
- Program the central EMDACC4 with the last firmware

1. Deploy the central EMDACC4

After having loaded the firmware on the Micro:bit that will work as the central device, take the EMDACC4 module equipped with the **current sensor board** and connect the Micro:bit to it. Attach the battery and close the enclosure. Turn it on and check if the symbol of a plug appears on the display: this means the board is properly connected and is broadcasting **instantaneous power values** (in watts) via radio packets.

Now connect the sensor to the first appliance you want to measure. Start with a simple lamp, then move on to other devices.

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 **“Energy Consumption”**.

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 sensor is broadcasting and monitor how the values change according to power consumption.

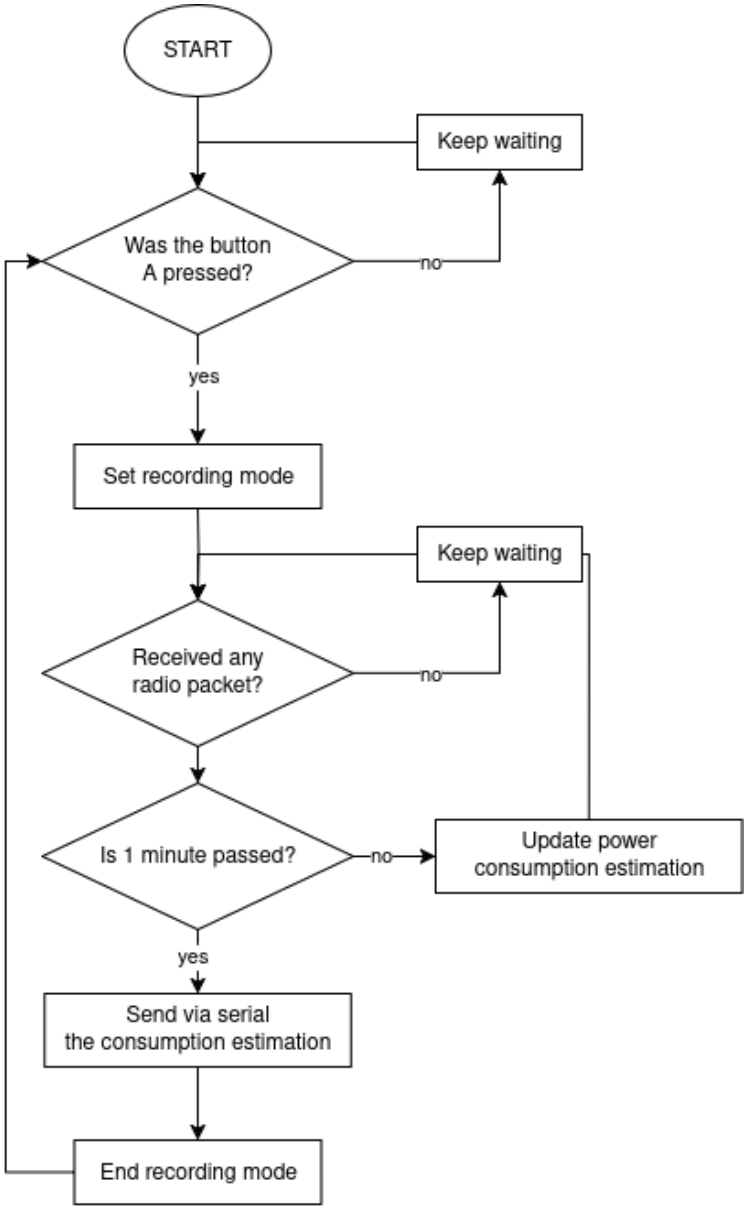
3. Design the algorithm

The goal of this activity is to create a tool to estimate **how much energy (Wh)** an appliance uses. To do so we are going to read the instantaneous measurements of Watts used within 1 minute and we will make an estimation.

The procedure is:

1. Receive instantaneous power readings (in watts) from the central EMDACC4.
2. At every reading, add the value to a running average.
3. Keep track of the elapsed time.
4. After 1 minute, stop the measurement and calculate the energy consumed.

At the end, we will have an estimate of the **Wh consumed by the appliance.**



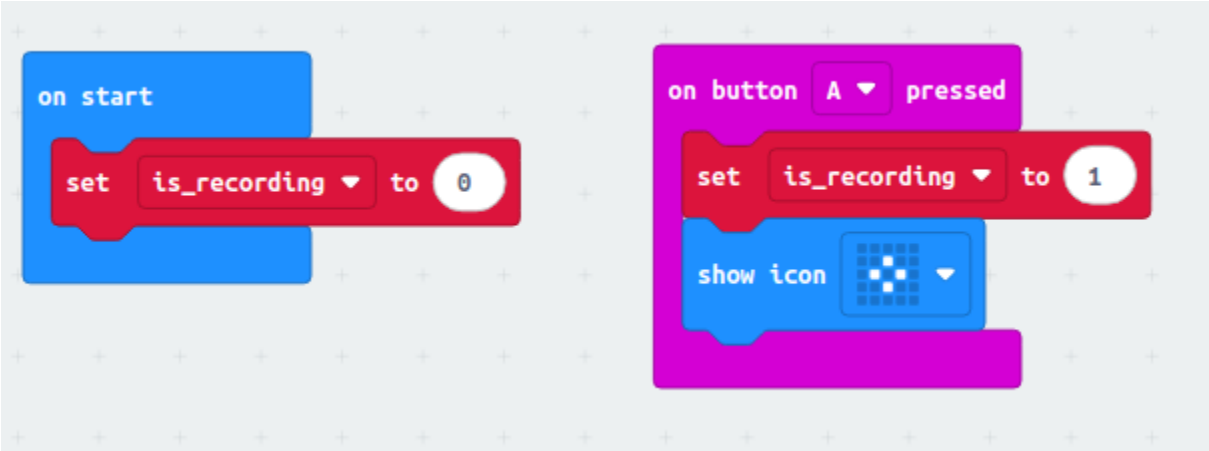
Do not worry if it does not come like this at the first attempt: writing algorithms is done in steps, you start from a vague idea and then you refine it adding the details and checking the corner cases. Do not rush to the code or you might forget something important!

4. Program the Micro:bit

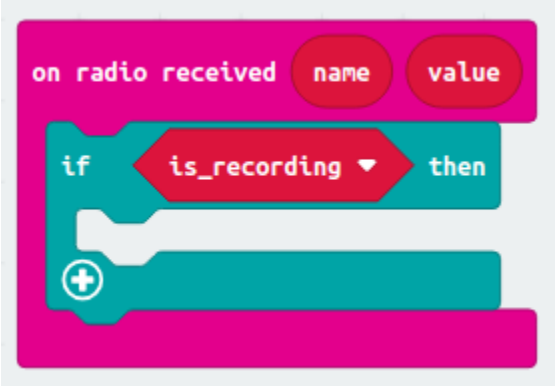
After having the algorithm clear in mind, it is possible to go on with the real programming of the device.

It is important to notice that the tool we are coding is going to present two different states: one while the device is in standby and it is waiting for the button A to be pressed, the other state is while the device is recording the incoming data. We can use a variable “is_recording” to keep track of such state so that every part of our code will have a way to understand which action we are currently doing and, to make it evident also to the user, we can show a specific icon while we are recording.

We can now proceed and code the first part of the flowchart. In order to detect if the button A is pressed we can use a block that we can find in the “Input” menu: “on button A pressed”.



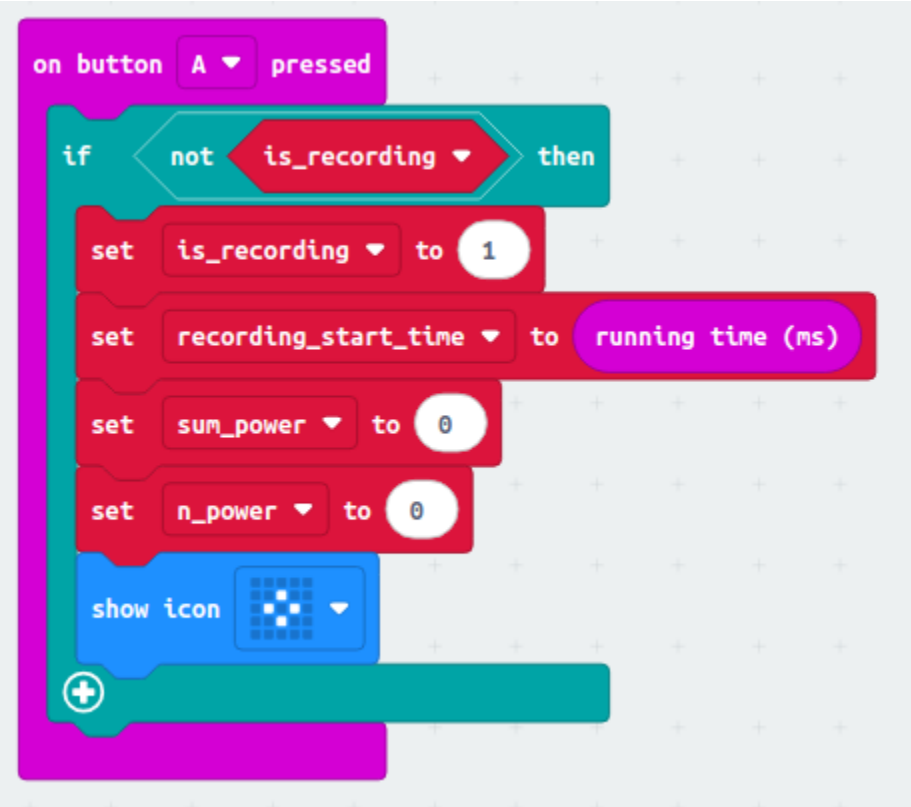
Now it is possible to use the usual block to receive packets from the central EMDACC4 device, the “on radio receive <name> <value>”. In such a block we need to react to packets only if we are recording so we need to set a condition that makes use of the variable we have defined above.



Since appliances can have variable power usage (do add a serial write to check it via web plot) and we want to estimate the average, we need to keep track of the

values that we are going to collect. For such purposes we need to add two more variables: the sum of all the values and the number of measured values, with these two data we can always compute the average power consumption. Since we also need to reset these two variables when we start a new recording, it is a good idea to set them to 0 when we press A.

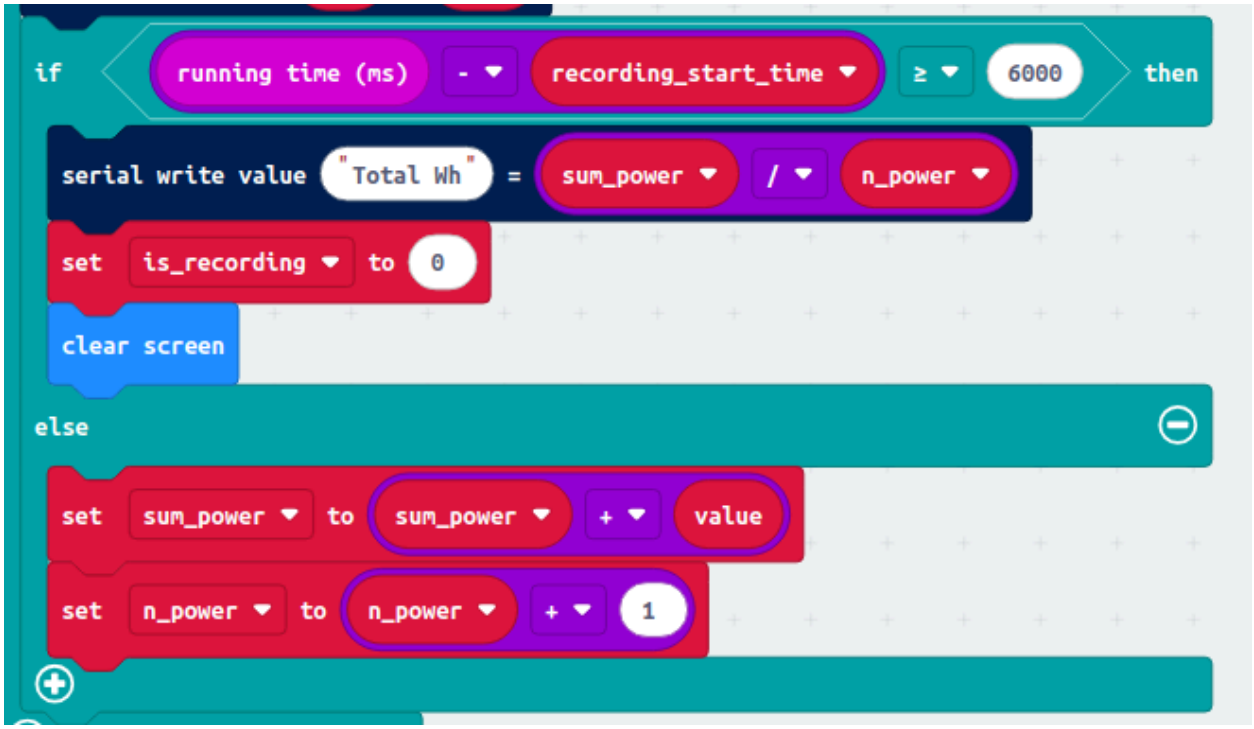
We also need to keep track of the moment in which we started recording so we can understand when a minute has passed. The final result of the block executed on pressing A should look similar to this:



After we take care of all the variables we need, we can proceed and code the block that manages the incoming radio packages. In particular, we need to check if a minute has passed since when we started the recording: we subtract to the current time the time we started recording and we check if it is less than 60 seconds (60000 ms).

If we still have not reached 1 minute, we just keep summing the measured power to each other and we keep track of how many values we summed together, so that when we are done measuring we can compute the average and send it via serial to the computer. Once we are done recording we also set the status

accordingly and we clear what is represented on the display. The whole block should look like:



After having double checked the code, you can load onto the Micro:bit to check if it works as expected!

5. Analyze the data

Repeat the measurement for different appliances and collect the data in a table.

A simple table could include the name of the appliances, the amount of variation of the plot, the Wh measured, the average usage time and other values that you think are relevant.

Try to explain the data you observe by looking at the table and the plots shown during measurement:

- Why does the consumption of some appliances vary and why some others are constant?
- Which appliances consume more? And why?
- Pay attention also to the usage duration: does this parameter change the relevance of the energy used?

Ideas

1. Estimate the **cost** of running each appliance for one hour using the local electricity price per kWh.
2. Compare the same appliance in different states (e.g., a computer idle vs. working hard, a lamp with LED vs. incandescent bulb).
3. Discuss which appliances are worth switching off to save energy and which ones consume so little that it hardly matters.

Reference code

