Energy light timer

Step 1: Make it

What is it?

Time how long your lights are left on to track your energy use. You could also use this project to track hours of sunshine in a weather station project.

This project is part of a series created to offer problem-solving and prototyping activities designed to explore technology as a solution to the challenges of the Global Goals for Sustainable Development.

What you'll learn

- how to use sensors and code to monitor real world events
- place a data-logger to record reliable data
- collect and collate data over time to spot patterns in energy use
- interpret and analyse data in order to effect changes in behaviour
- how variables can be changed by a user to configure a system prior to use

How to use it

- First use the Energy light meter project to find out the readings when your lights are on and off. Make sure you do this in the same place and lighting conditions that you will place your monitoring micro:bit, and ensure daylight doesn't trigger a false reading that your lights are on.
- Put your light reading into the code where the LIGHT variable is set. We have given you the number 100 but you will probably need to change this before flashing the code on to your micro:bit.
- Attach a battery pack and place your micro:bit under the light you want to monitor. You should see a dot on the display when the light is off, and the display lights up when your light is on. If this doesn't work, consider using the Light meter project again to find the light level when the light is on, or move the micro:bit.
- The micro:bit will keep timing and when you press button B it will show how long the light has been switched on in minutes.

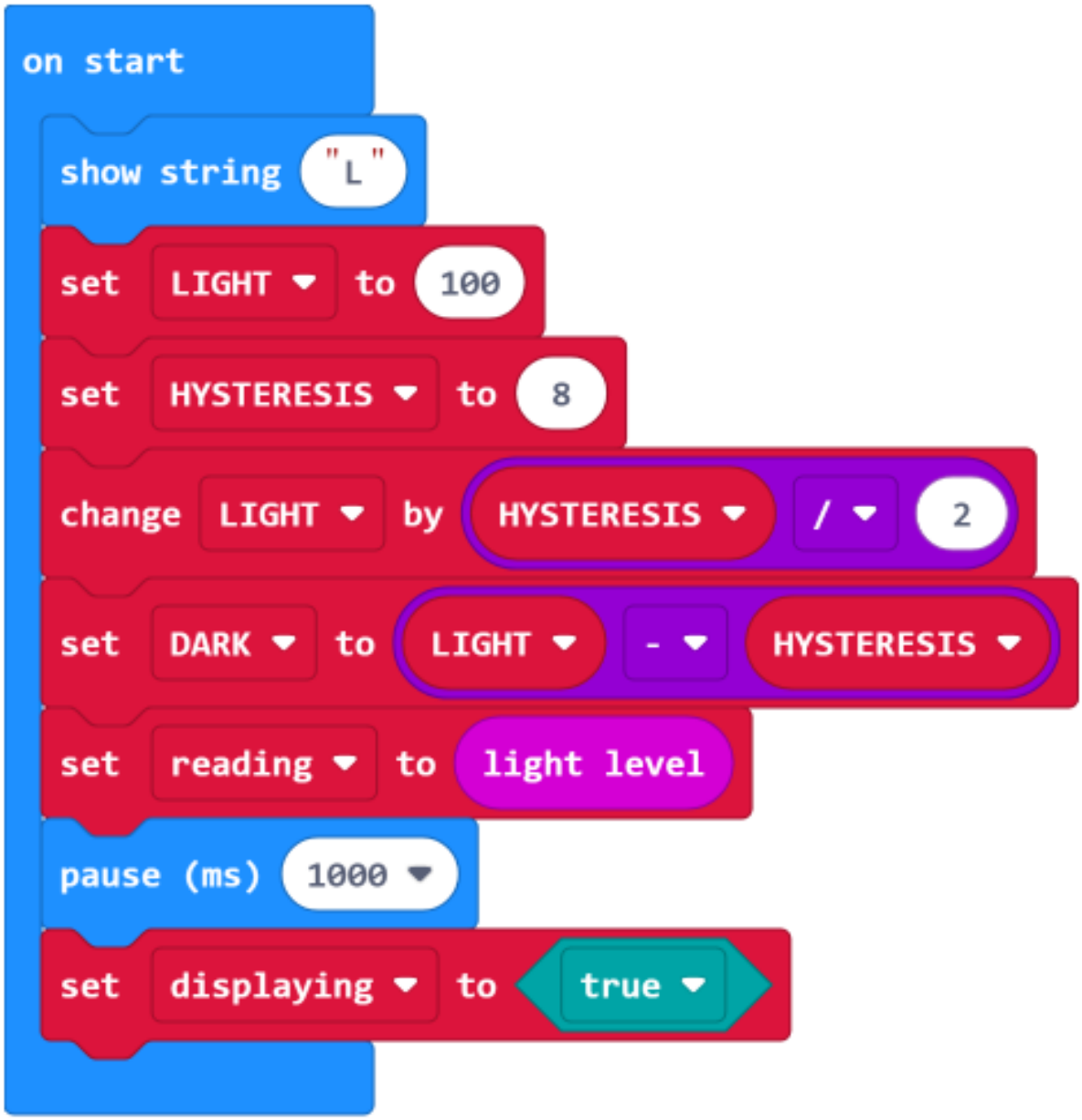
How it works

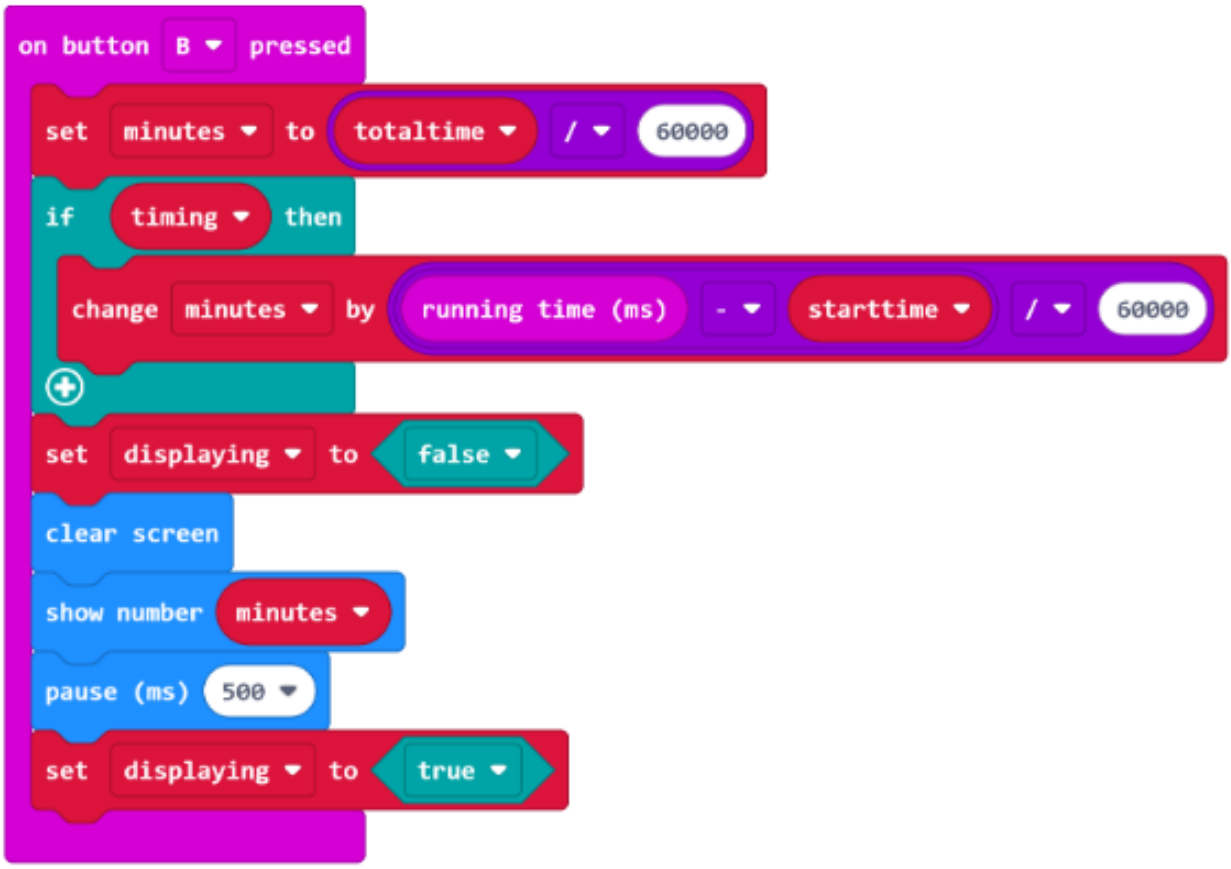
- The program uses a Boolean variable called **timing** to control the program. Boolean variables can only have two values: true or false.
- If the light sensor reading is above the level you set, it starts the timer and lights up the LED display. If it falls below this level, it stops the timer and shows a dot on the display.
- This project uses **hysteresis** to make sure the timer doesn't switch on and off too often when the light level is changing slightly around the threshold for triggering the timer. It creates a larger band around the threshold which has to be crossed before the timer is turned on or off.
- Hysteresis is a common feature in control systems that use sensors, for example in heating systems that have a thermostat. If you set your thermostat to a certain temperature, you don't want the heating repeatedly switching on and off very quickly when the temperature hovers around the number you have set. Hysteresis prevents this happening.

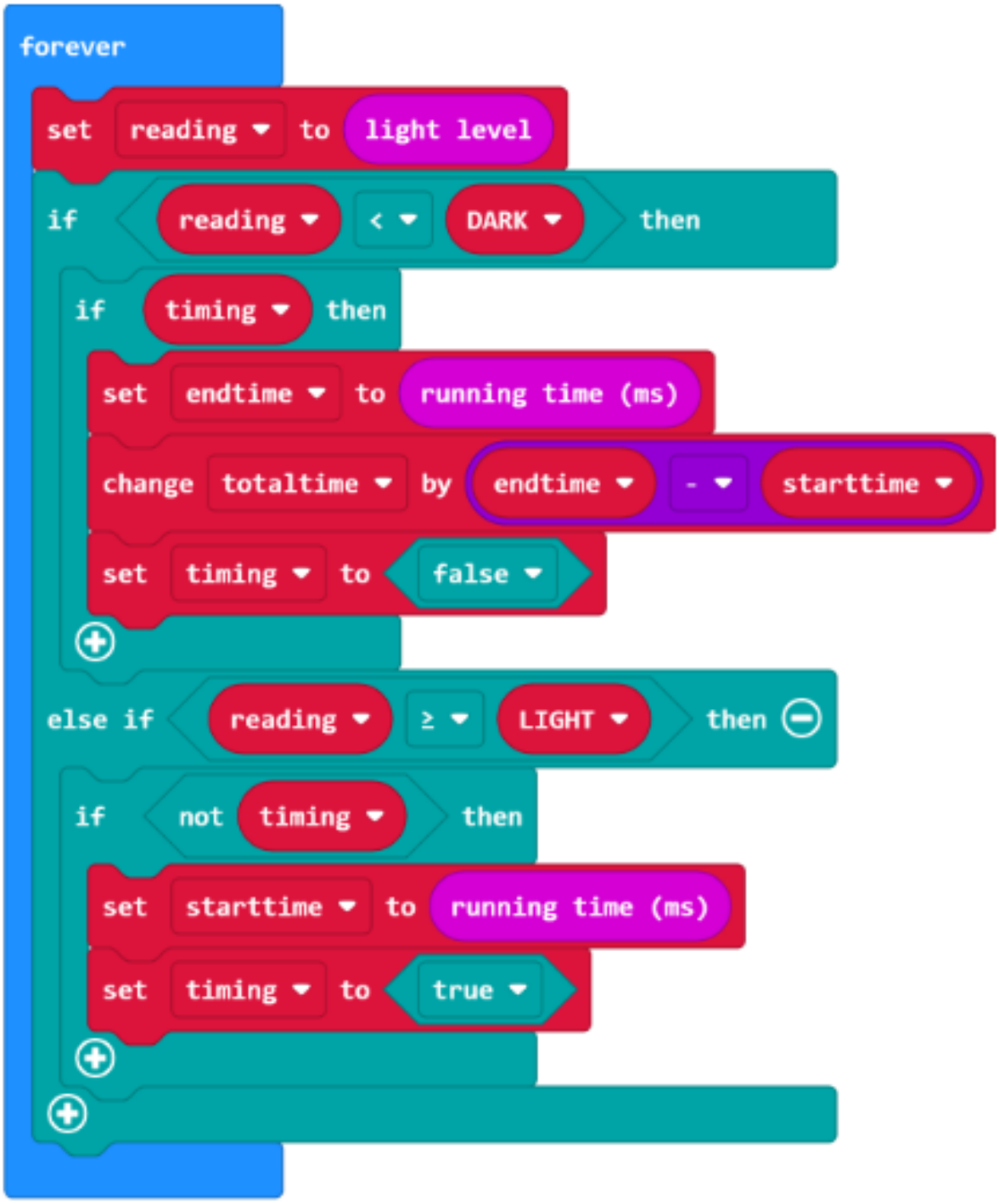
What you need

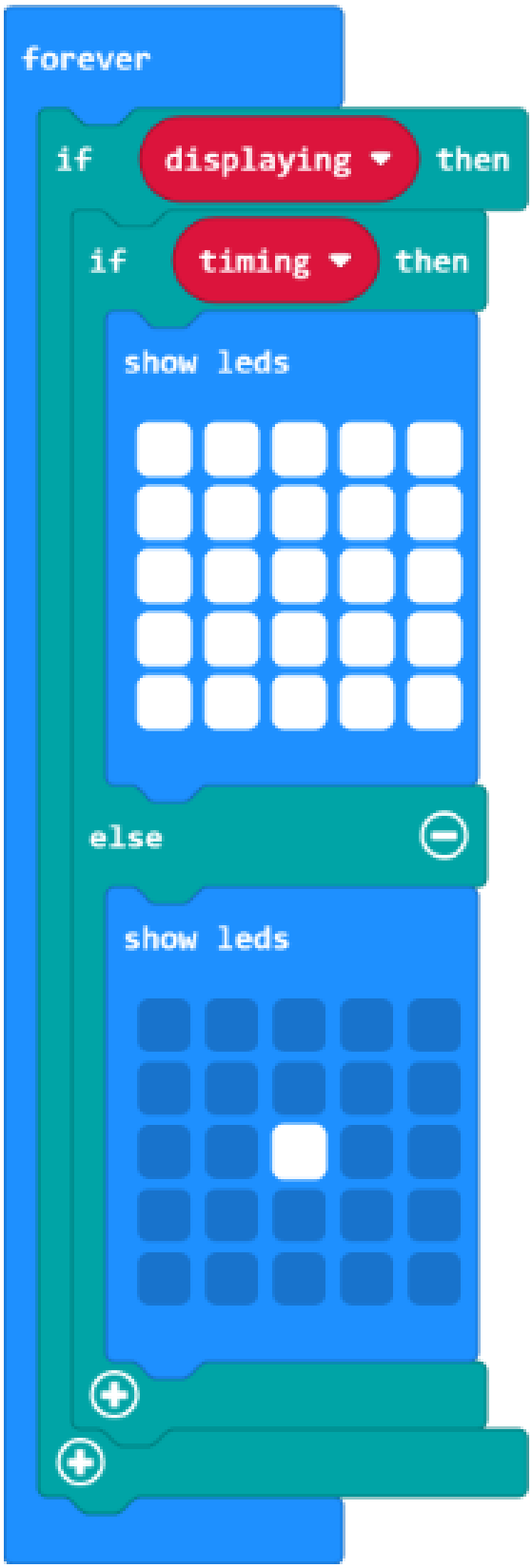
- micro:bit
- MakeCode or Python editor
- battery pack (recommended)

Step 2: Code it




on button B pressed
set minutes to totaltime / 60000
if timing then
change minutes by running time (ms) - starttime / 60000
set displaying to false
clear screen
show number minutes
pause (ms) 500
set displaying to true


forever
set reading to light level
if reading < DARK then
if timing then
set endtime to running time (ms)
change totaltime by endtime - starttime
set timing to false
else if reading ≥ LIGHT then
if not timing then
set starttime to running time (ms)
set timing to true


forever
if
displaying
then
if
timing
then
show leds
else
show leds

Step 3: Improve it

- Change the patterns shown on the LED display to make the batteries last longer, or dim the display.
- Use the time recorded to work out how much electricity you have used and what it may cost. Details are in our Energy cost calculator project
- Use the same code to measure hours of sunlight. Use the Light meter project to calculate the light value when sun is shining on the micro:bit and when it's cloudy. You may need to make a container for your micro:bit with a translucent lid to diffuse the light and protect the micro:bit from rain, for example an old plastic food container.