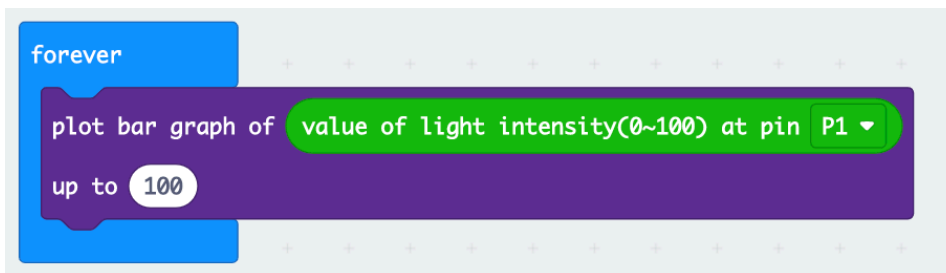


Handout

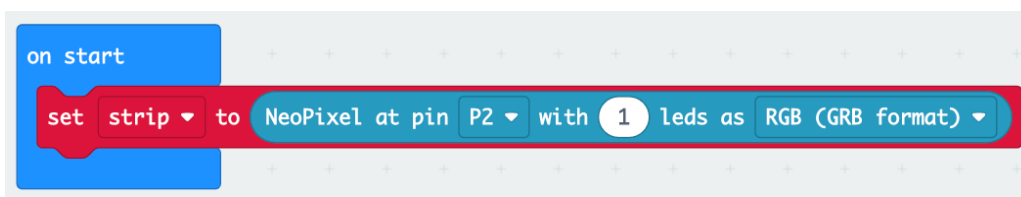
Smart Light

The smart light works by turning on when it is dark, and turning off when it is light outside. This is applicable to outdoor lights that you only want on when needed. To start, you will need to use the light sensor to record different light values. You'll want to record what the light value is during different times to get a sense of what value should turn on/off the light. To do this, attach the light sensor to Pin1 and use the plot feature at different times.



Once you have your reading, delete the bar graph code. In our experiment, we found that a light intensity of 20 was sufficiently dark enough to require lighting.

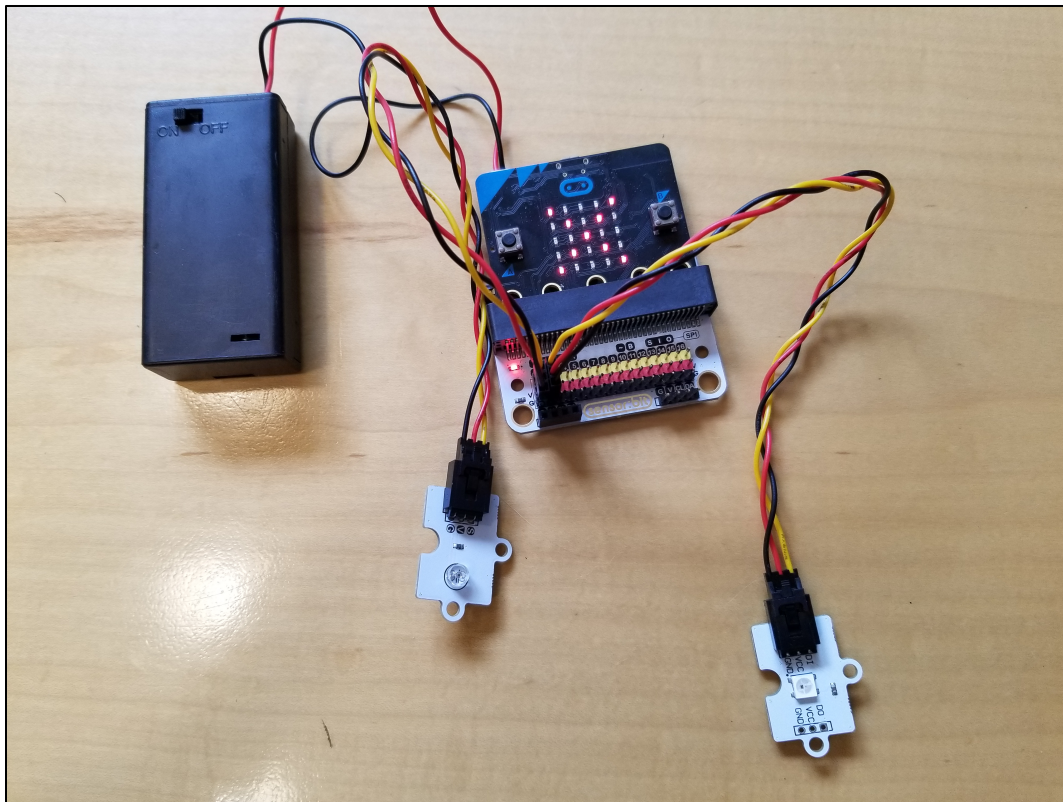
Keep the light sensor at Pin1, then attach the LED to Pin2. Set-up the LED strip with 1 LED using the RGB format.



Using the determined value, create a conditional statement that uses the light intensity to turn the light on or off. In this set-up, if the light intensity is greater than 20 (it's bright outside), the LED strip is set to black (off) and an X is shown to indicate the light is off. If it is not greater than 20, i.e. it's the less than 20 and dark, the LED strip is set to white (on) and a Y is shown to indicate the light is on.



Here is the final set-up:



Smart Fridge

The smart fridge is another energy saving set-up, that while equally simple in concept requires more code. This device can be used to warn you if the fridge has been left open for too long. When the fridge is closed, the crash sensor is pressed and nothing happens. When the fridge is open however, the crash sensor is no longer pressed which triggers a timer. As the time increases with the fridge door open, a warning light goes from green, to yellow, to red. When the light turns red, an alarm sounds reminding you to close the fridge. The timer is reset when the fridge is closed and the crash sensor is being pressed.

To start, set-up the crash sensor at Pin1 and the LED at Pin 2.

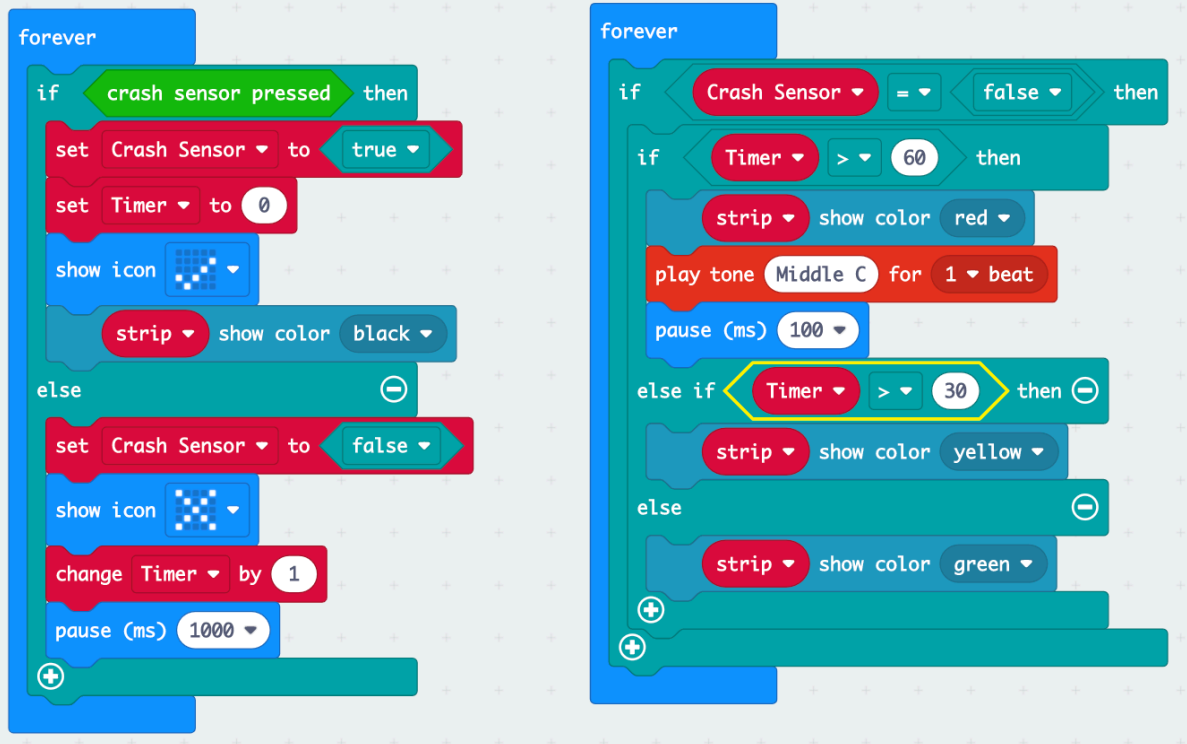


Start by creating two variables, one for the timer and one for the crash sensor.

Create a conditional statement inside a forever loop that sets-up the crash sensor. In this code, if the crash sensor is pressed the timer resets, a check mark is shown and the LED shows black (off).

Additionally, it sets the variable crash sensor to true. If the crash sensor is not being pressed (else), the timer starts to increase by one every second, an X is shown and the crash sensor is set to false.

What happens when the crash sensor is set to false is addressed in the second forever loop. When the crash sensor is not being pressed (set to false) an if else statement is used to indicate what occurs depending on how long the timer has been running for. If the timer has run for more than 60 seconds, the LED turns red and plays an alarm sound. If the timer has not yet reached 60 seconds but has run for more than 30 seconds, the LED turns yellow. If the timer has not yet reached 30 seconds, the LED turns green which is the default as soon as the crash sensor is no longer being pressed.



Here is the final set-up:

