



NSF Engineering Research Center

DIY Environmental Sensor

Overview

Students enjoy seeing concepts they are learning in class applied to real life. I believe that students feel a greater sense of ownership when they build or design a project related to their learning, rather than follow a teacher's step-by-step instructions.

Students will build an environmental sensor (hence "DIY" or Do It Yourself) using an Arduino MKR microcontroller and a Nicla Sense Env board. After assembling the circuit and programming the sensor, they will engage in guided scientific inquiry where they will use the sensor to analyze their home or school environment. The project culminates in a class presentation.

Key Search Words

High School Science, Environmental Science, Chemistry, Biology, Engineering, Project-Based Learning, Microcontroller, Programming, Air Particulates

Learning Objectives

At the end of this lesson, students will be able to:

1. **Use a Microcontroller** to interface with a computer and take measurements.
2. **Code** in C++ to tell the sensor what to look for in the environment.
3. **Design an experiment** to answer questions about how human/pet activity affects the environment.
4. **Present and show** scientific findings in a presentation.

Curriculum Alignment

This is an interdisciplinary lesson. I will cite standards in Biology and Earth/Environmental Science (there are no state standards for Engineering). This lesson plan is intended for high school students of any age, though perhaps 7th-8th grade students with prior coding experience could also be successful with this project.

LS.Bio.10.1 Construct an explanation to illustrate how varying environmental conditions may result in: changes in the number of individuals of a species, the emergence of new species over time, or the extinction of other species.

ESS.EES.4.4 Construct an explanation to predict how potential future changes in abiotic factors could impact biodiversity and species distribution.

ESS.EES.5.3 Construct an argument to evaluate the ways that human activities influence atmospheric composition.

Classroom time required

This project will require between 2-6 school days (between 2-10 hours), depending on the number and complexity of your students' experimental questions.

Day 1: Assembling hardware, writing software (plug, program, and use the microcontroller and sensor) (1-2 hours)

Days 2-4: Experimental Design (write and test experiment) (1-5 hours)

Days 5-6 (Optional): Data processing and planning presentation (1-3 hours)

Materials & Technology

- Arduino MKR WIFI 1010 ([User manual](#) / [Amazon](#) link) – \$40
- Nicla Sense Env ([User manual](#) / [Amazon](#) link / [Arduino website](#) link) – \$40
- Breadboard ([Amazon](#) link – pack of 6) – \$7 (optional)
- Set of wires ([Amazon](#) link – pack of 840 (good for 6 breadboards)) – \$10 (optional)
- Students need a laptop to connect to and power the Arduino MKR. The kind of computer dictates which cable they need! The Arduino MKR has a micro-USB connection.
 - Most PCs use USB-cable connection (Micro-USB to USB-cable 2-pack [linked here](#)) - \$5
 - Most Macs require a Type C connection (Micro-USB to Type C cable 2-pack [linked here](#)) - \$5
- They need to download [Arduino software](#) – Free
- I recommend students work in pairs but they can work in groups of 3, 4, or 5. Each group requires 1 laptop, 1 Arduino MKR WIFI, 1 Nicla Sense Env, and 1 cable. If students will add a feature to the Arduino, each group may also want 1 breadboard and a handful of wires.

Safety

Arduinos use a low 3.3 Volt (V) input, which is not harmful to humans. It is possible to harm the Arduino or possibly the USB port of your computer if you misplug wires, such as connecting a 3.3V source directly to ground. For this lab there is no need to experiment with more complex circuitry, just be cognizant that there is a possibility that a computer driver or the Arduino itself could be damaged if students plug wires arbitrarily.

Teacher Preparation for Activity

Teachers will need to have the materials purchased and ready. Assuming you haven't used Arduinos before, I *highly recommend* connecting the Nicla Sense Env with the Arduino MKR WIFI 1010 yourself. Copy + paste the code and upload it onto your Arduino. It is hard to answer student questions unless you yourself have familiarity with Arduinos.

The teacher should decide beforehand if they would like the students to only use the technology at school, or if they can take the technology home.

Student Preparation for Activity

Instruct students to download the Arduino software before beginning the project.

This project involves guided inquiry, where the teacher has some idea of what the students will do, but the students will ultimately decide what particulates they want to measure and where/how they will measure them.

Students should first write a series of questions involving particulate matter in the environment. Here are a handful of sample questions. You may provide 1-2 examples to prompt them.

1. How do temperature and humidity vary from day-to-night in my home?
2. How do levels of Volatile Organic Compounds (VOCs) vary around my home and with human activity? For example – will I detect more VOCs when cooking?
3. Can I predict human presence in the room by measuring Carbon Dioxide (CO₂) levels?
4. Since Nitrogen Dioxide (NO₂) is produced from burning fuel, can I detect oven or grill use by measuring NO₂?
5. Can I detect Ozone (O₃) from using air purifiers, hair dryers, or from the environment?
6. How do air purifiers or humidifiers affect these levels in my home?

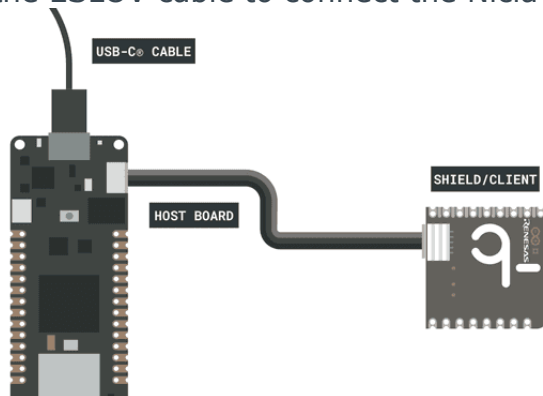
If students will be taking the sensor home for home testing, it is important that they discuss this with their parents beforehand. If a sensor is moved over the course of data collection, the data may not be reliable. It is worth mentioning that the sensor could potentially detect unsafe air quality, such as ozone levels above 0.070 parts per million.

Procedure

Day 1: Assembling Hardware, Writing Software

1. Sit students in groups. At each group ensure there is the set of materials needed (computer, MKR WIFI, Nicla Sense, cable (breadboard and wires optional))
2. If students use their own laptop, ensure they have downloaded the Arduino IDE. Otherwise, ensure each school computer has downloaded Arduino IDE.

3. Use the ESLOV cable to connect the Nicla Sense and MKR WIFI boards.



From <https://docs.arduino.cc/tutorials/nicla-sense-env/user-manual/>

4. Copy + paste the "Blink" example sketch on the above user manual link to verify that the board is connected. A correctly working code blinks the orange LED indicator on and off on the Nicla Sense. (The "Blink" code is also displayed in the Appendices section.)
5. Copy + paste (or type) the "Temperature and Humidity" code shown on the user manual link. (This code is also in the Appendices section.)
6. Test out using this code in at least two different environmental conditions.

Days 2-4: Experimental Design

Write testable hypotheses

1. Students should now understand the capabilities of the Nicla Sense. Given their initial questions from before, students should formulate testable hypotheses. A well-meaning question like "What happens to CO₂ levels in the day?" is not very specific. Better to ask "How do CO₂ levels vary in a classroom during the day as occupancy changes?"

Here are some considerations for writing testable hypotheses:

- a. The question involves *measuring* a quantity.
- b. The question is *specific* to a particular situation.
- c. The question involves how an *independent variable* affects a *measurement*.

Groups should write 1-5 hypotheses (depending on class time)

Experimental Design

2. Now that students have specific questions, they will design an experiment to test their interests.

As much as you can, try to engage the student to design the experiment as much as possible, giving them scaffolded tips along the way.

Questions students ought to consider when designing how, when, and where to use the Nicla Sense Env:

- a. What variables should be controlled?
- b. How many repeated measurements are needed?
- c. How can we fairly compare different environments?

Let's look at an example. Suppose students want to test the aforementioned question: "How do CO₂ levels vary in a classroom during the day as occupancy changes?". The group could use the Nicla Sense in a classroom. They would want to set up conditions where the occupancy changes, such as 30 minutes before class, during a full class of students, and 30 minutes afterwards. They can use the sensor to measure CO₂ levels, sampling every 10 seconds. They can test this over the course of 3 different classes, ideally at the same time of day (i.e. in the morning).

As the students test, they should iterate and redesign the experiment as needed. For example, perhaps they need to sample every 30 seconds if the 10 second sampling rate data is noisy (fluctuates a lot). Or perhaps they need to change the sampling location to move the sensor from floor height to desk height. Perhaps students want to also test for CO₂ levels in a big library and compare it with results from a small classroom.

Indoor Air Quality Testing

3. Copy + paste (or type) the "Indoor Air Quality" code shown in the user manual link. (This code is also displayed in the Appendices section.)

Practice using this code in an *indoor* environment. Students should try a variety of environmental conditions. They can vary things like the number of people in the room, the size of the room, and the activities conducted in that room (eating, talking, or over a locker with lots of passing students, etc).

Outdoor Air Quality Testing

4. Copy + paste (or type) the "Outdoor Air Quality" code shown in the user manual link. (This code is also displayed in the Appendices section.)

Practice using this code in an *outdoor* environment. Students should try a variety of environmental conditions, such as sampling at evening or daytime, on a hot or cold or rainy day, or in the presence of people or pets.

Days 5-6 (Optional): Data Processing and Planning Presentation

Data Processing

1. The data output might be noisy. The levels of CO₂ or humidity or VOCs may vary quite a lot, especially if sampling over short time intervals.

Ask the students about making a moving average or filtering their data. For example, if they sampled every 10 seconds, perhaps you could bin the data into 6-sample averages and then graph the average measurement every minute.

Students could also filter the data by discarding data points with significant deviations from adjacent measurements. Students would want to be upfront about disclosing this filtering during their presentation.

Anomaly Detector

2. If time permits, students can use the sensor to detect anomalies. For example, ozone O₃ is considered unhealthy for prolonged exposure at levels above 0.070 ppm (parts per million). They could write in the code to show a print statement warning, or add a red LED warning indicator, if the ozone exceeds this level. The same could be done for excessively high temperature, CO₂, or NO₂ levels.

Planning Presentation

3. I believe it is more meaningful in this project for students to be graded on presentation feedback rather than taking a quiz. A sample presentation rubric is in the appendices section.

Students can spend some time in class (or work as homework) to prepare a slideshow. Their presentation should discuss their process, hypotheses, and experiment. Furthermore, they should show graphs displaying their results, make scientific inferences from their results, and discuss error analysis showing the limitations and future improvements on their experiment.

Differentiation

Although it empowers students to do as much as they can themselves, the project can be modified to take less time and make it simpler. For example, rather than having students type the code, you can have the code already saved and provided as a download, either on your website or from the [Nicla Sense Env User Manual](#). You could also not do the Data Processing, or limit the measurements (i.e. only test for CO₂ levels).

On the other hand, more advanced students could do the full activity. If students have experience coding, you could have them modify the code and provide extra features, such as incorporating an LED light or a speaker to activate when temperatures exceed 30° C.

Assessment/Check for Understanding

1. The presentation can be graded with a rubric. A sample rubric is posted in the appendices section.
2. The teacher should monitor student progress during the working class periods. Teachers should help ensure the students make insightful, specific, measurable questions at the outset of this project.

Required resources

- The [Nicla Sense Env User Manual](#) has sample codes for students to use to allow the Nicla Sense to measure. It may be more practical for a teacher to provide these codes (also listed in appendices) on their school website, but they can also direct students to use the code here.

Author comments

I initially taught Engineering with zero experience. I hadn't taken an Engineering course, used Arduinos, nor programmed with C++. I encourage teachers to dive headfirst! You will surprise yourself how quickly you learn. If you follow the user manual, it is easy and straightforward to use this technology. I think your students will also love this exciting opportunity.

Sources

Bagur, J. (2025, November 24). *Nicla Sense Env User Manual*. Arduino Docs.
<https://docs.arduino.cc/tutorials/nicla-sense-env/user-manual/>

Appendices

Page 1 – “Blink” code:

```
/**
  Blink Example on Nicla Sense Env
  Name: nicla_sense_env_blink.ino
  Purpose: This sketch demonstrates how to blink the onboard
  orange LED of the Nicla Sense Env board.

  @author Arduino Product Experience Team
  @version 1.0 31/05/24
*/

// Include the NiclaSenseEnv library
#include "NiclaSenseEnv.h"

// Global device object for the Nicla Sense Env board
NiclaSenseEnv device;

/**
  Toggles the onboard orange LED between on and off states.
  @param led Reference to OrangeLED object controlling the LED.
*/
void toggleLED(OrangeLED& led) {
  // Turn on the LED to full brightness for one second
  led.setBrightness(63);
  delay(1000);

  // Turn off the LED for one second
  led.setBrightness(0);
  delay(1000);
}

void setup() {
  // Initialize serial communication at 115200 bits per second.
  Serial.begin(115200);

  // Wait for Serial to be ready with a timeout of 5 seconds
  for(auto start = millis(); !Serial && millis() - start < 5000;);

  if (device.begin()) {
    // Initialize the onboard orange LED
    auto orangeLED = device.orangeLED();
  }
}

void loop() {
  // Retrieve the orange LED object
  OrangeLED orangeLED = device.orangeLED();

  // Continuously toggle the orange LED on and off
  toggleLED(orangeLED);
}
```

Page 2 – “Temperature and Humidity” code:

```
/**
 * Temperature and Humidity Sensor Example for Nicla Sense Env
 * Name: nicla_sense_env_temp_humidity_example.ino
 * Purpose: This sketch demonstrates how to read temperature and humidity from
 * the HS4001 sensor on the Nicla Sense Env using the Arduino_NiclaSenseEnv library API.
 *
 * @author Arduino Product Experience Team
 * @version 1.0 31/05/24
 */

// Include the NiclaSenseEnv library
#include "NiclaSenseEnv.h"

// Global device object for Nicla Sense Env
NiclaSenseEnv device;

/**
 * Displays temperature and humidity data from the HS4001 sensor.
 * @param sensor Reference to TemperatureHumiditySensor object controlling the sensor.
 */
void displaySensorData(TemperatureHumiditySensor& sensor) {
    if (sensor.enabled()) {
        float temperature = sensor.temperature();
        if (isnan(temperature)) {
            Serial.println("- Temperature: N/A");
        } else {
            Serial.print("- Temperature: ");
            Serial.print(temperature, 2);
            Serial.println(" °C");
        }
        Serial.print("- Relative humidity: ");
        Serial.print(sensor.humidity(), 2);
        Serial.println(" %");
        Serial.println("");
    } else {
        Serial.println("- Temperature sensor is disabled!");
    }
}

void setup() {
    // Initialize serial communication and wait up to 2.5 seconds for a connection
    Serial.begin(115200);
    for (auto startNow = millis() + 2500; !Serial && millis() < startNow; delay(500));

    if (device.begin()) {
        Serial.println("- Device is connected!");
    } else {
        Serial.println("- Device could not be found. Please double check the wiring!");
    }
}

void loop() {
    // Read data from the HS4001 sensor
    // Wait for 2.5 seconds before reading again
    auto temperatureSensor = device.temperatureHumiditySensor();
    displaySensorData(temperatureSensor);
    delay(2500);
}
```

Page 3 – “Indoor Air Quality” Code:

```
/**
  Indoor Air Quality Sensor Example for Nicla Sense Env
  Name: nicla_sense_env_indoor_air_quality_example.ino
  Purpose: This sketch demonstrates how to read air quality data from the
  ZMOD4410 sensor on the Nicla Sense Env using the Arduino_NiclaSenseEnv library API.

  @author Arduino Product Experience Team
  @version 1.0 31/05/24
*/

// Include the NiclaSenseEnv library
#include "NiclaSenseEnv.h"

// Global device object for Nicla Sense Env
NiclaSenseEnv device;

/**
  Displays air quality data from the ZMOD4410 sensor.
  @param sensor Reference to IndoorAirQualitySensor object controlling the sensor.
*/
void displaySensorData(IndoorAirQualitySensor& sensor) {
  if (sensor.enabled()) {
    Serial.print("- Indoor air quality value: ");
    Serial.println(sensor.airQuality());
    Serial.print("- CO2 (ppm): ");
    Serial.println(sensor.CO2());
    Serial.print("- TVOC (mg/m3): ");
    Serial.println(sensor.TVOC());
    Serial.print("- Ethanol (ppm): ");
    Serial.println(sensor.ethanol());
    Serial.println("");
  } else {
    Serial.println("- Indoor air quality sensor is disabled!");
  }
}

void setup() {
  // Initialize serial communication and wait up to 2.5 seconds for a connection
  Serial.begin(115200);
  for (auto startNow = millis() + 2500; !Serial && millis() < startNow; delay(500));

  if (device.begin()) {
    Serial.println("- Device is connected!");
    auto airQualitySensor = device.indoorAirQualitySensor();

    // Set the sensor mode to indoor air quality
    airQualitySensor.setMode(IndoorAirQualitySensorMode::indoorAirQuality);

    // The ZMOD4410 can take a sample every 3 seconds in IAQ mode and requires 60 warm-up
samples,
    // meaning the sensor will take about 3 minutes to fully warm-up before accurate readings can
    // be obtained. In this example, we allow 5 seconds for the sensor to start delivering data.
    delay(5000);
  } else {
    Serial.println("- Device could not be found. Please double-check the wiring!");
  }
}

void loop() {
  // Read data from the ZMOD4410 sensor every 5 seconds
  auto airQualitySensor = device.indoorAirQualitySensor();
  displaySensorData(airQualitySensor);
  delay(5000);
}
```

Page 4 – “Outdoor Air Quality” Code:

```
/**
 Outdoor Air Quality Sensor Example for Nicla Sense Env
 Name: nicla_sense_env_outdoor_air_quality_example.ino
 Purpose: This sketch demonstrates how to read air quality data from the
 ZMOD4510 sensor on the Nicla Sense Env using the Arduino_NiclaSenseEnv library API.

 @author Arduino Product Experience Team
 @version 1.0 31/05/24
 */

// Include the NiclaSenseEnv library
#include "NiclaSenseEnv.h"

// Global device object for Nicla Sense Env
NiclaSenseEnv device;

/**
 Displays air quality data from the ZMOD4510 sensor.
 @param sensor Reference to OutdoorAirQualitySensor object controlling the sensor.
 */
void displaySensorData(OutdoorAirQualitySensor& sensor) {
    if (sensor.enabled()) {
        Serial.print("- Outdoor air quality value: ");
        Serial.println(sensor.airQualityIndex());
        Serial.print("- NO2 (ppb): ");
        Serial.println(sensor.NO2());
        Serial.print("- O3 (ppb): ");
        Serial.println(sensor.O3());
        Serial.println("");
    } else {
        Serial.println("- Outdoor air quality sensor is disabled!");
    }
}

void setup() {
    // Initialize serial communication and wait up to 2.5 seconds for a connection
    Serial.begin(115200);
    for (auto startNow = millis() + 2500; !Serial && millis() < startNow; delay(500));

    if (device.begin()) {
        Serial.println("- Device is connected!");
        auto outdoorAirQualitySensor = device.outdoorAirQualitySensor();

        // Enable the outdoor air quality sensor
        outdoorAirQualitySensor.setMode(OutdoorAirQualitySensorMode::outdoorAirQuality);
        outdoorAirQualitySensor.setEnabled(true);

        // The ZMOD4510 takes a sample every 6 seconds in NO2/O3 mode and requires 50 warm-up
samples,
        // meaning the sensor will take about 5 minutes to fully warm-up before accurate readings
data.
        // can be obtained. In this example, we allow 6 seconds for the sensor to start delivering
        delay(6000);
    } else {
        Serial.println("- Device could not be found. Please double-check the wiring!");
    }
}

void loop() {
    // Read data from the ZMOD4510 sensor every 5 seconds
    auto outdoorAirQualitySensor = device.outdoorAirQualitySensor();
    displaySensorData(outdoorAirQualitySensor);
    delay(6000);
}
```

Sample Presentation Rubric:

	Pts	Excellent	Solid	Fair	Poor
4-6 minutes	4	Presentation between 4-6 minutes (Q&A not included) (4)	Presentation over or under time by <60 seconds (3)	Presentation over or under time by 1-2 minutes (2)	Presentation over or under time by >2 minutes (1) or >3 minutes (0)
Working Sensor	6	Group measures and shows at least 4 different quantities (6)	Group measures and show at least 2 quantities (4)	Group measures and show at least 1 quantity (2)	Group is not able to measure any environmental factors (0)
Science / Theory	5	Group explains clearly what compounds they measured, how they are found or produced in the environment, and how they affect human/animal health (5)	Group mostly explains what compounds they measured and how they are found or produced in the environment (4)	Group discusses which compounds were measured, though it isn't clear what factors make them more or less present in the environment (2)	Group does not discuss how the measured quantities are affected by the environment (0)
Circuit / Code	5	Group shows the code they used, how they plugged in the circuit, & explains how the code tells the sensor to measure (5) Bonus: modifies code or circuit to add additional feature (6)	Group shows the code they used and how they plugged in the circuit (4-3)	Group shows the code they used or how they plugged in the circuit (2-1)	Group does not show the code they used, nor how they plugged in the circuit (0)
Presentation of results	5	Sensor results shown with clear, well organized series of graphs; data discussed with thoughtful analysis; some statistics done for clarity (5)	Sensor results shown with graphs; data discussed and analyzed (4-3)	Sensor results shown with a graph (2)	Sensor results not discussed, or no graph is present (0)
Interactive / Storytelling	5	Group interacts with and asks questions to audience; tells engaging story that connects data to real life (5)	Group interacts with audience; clearly connects the data to the environment (4-3)	Group interacts with audience (2-1)	Presentation is directed at audience with no opportunity for audience to speak (0)

Total: 30 points