12 Month project plan for introduction of energy conscious technology development

**EMPower**

# Intro

This document presents an innovative lesson plan idea for the entire school year, offering a captivating and educational project for each of the eight months. These projects are carefully designed to engage students and provide hands-on learning experiences that encompass a wide range of subjects, from science and technology to art and sustainability. The goal of this lesson plan is to foster creativity, critical thinking, and collaboration among students.

What makes this lesson plan truly unique is its openness and accessibility. We believe in the power of sharing knowledge and resources. Therefore, all project details, including code and materials, are made available on GitHub for anyone to access, use, and expand upon. This not only encourages educators to adopt and adapt these projects but also enables students, parents, and enthusiasts to explore, learn, and contribute to the projects.

Throughout the school year, students will embark on a journey of discovery and exploration, tackling projects that are not only exciting but also align with important educational objectives. By combining the joy of learning with the freedom to innovate, we hope to inspire a new generation of thinkers, problem-solvers, and creators.

Join us in this educational adventure, and let's explore the endless possibilities of learning, making, and sharing. Together, we can empower students and educators to embrace the wonders of knowledge and creativity throughout the school year and beyond.

# 1) Outdoor indoor temperature sensor using basic circuit components, such as resistors, bread boards, and thermistors.

**Introduction:** In this fun and educational project, you'll learn how to create a simple temperature sensor using basic electronic components like resistors, a breadboard, and a thermistor. You'll be able to measure and compare the outdoor and indoor temperatures using your DIY sensor.

**Materials Needed:**

* 1 x Thermistor (temperature sensor)
* 1 x 10k Ohm Resistor
* 1 x Breadboard
* Jumper Wires (male-to-male)
* Pencil and Paper
* Small container or box (to protect the outdoor sensor)

**Safety Precautions:**

* Always have adult supervision when working with electronics.
* Handle components gently to avoid damage.

**Instructions:**

**1. Learn About the Components:**

* Have an adult explain what the components are and how they work:
* Thermistor: A sensor that changes its resistance based on temperature.
* Resistor: A component used to limit the flow of electricity.
* Breadboard: A platform to build and connect electronic circuits.

**2. Build the Circuit:**

* Place the thermistor and the 10k Ohm resistor on the breadboard.
* Connect one end of the resistor to one leg of the thermistor.
* Connect the other end of the resistor to the GND (-) rail on the breadboard.
* Connect the other leg of the thermistor to the 5V (+) rail on the breadboard.
* Connect the center junction of the thermistor and resistor to an analog input pin on a microcontroller (if available). If not, you can use a multimeter to measure the resistance and calculate the temperature.

**3. Protect the Outdoor Sensor:**

* Place the outdoor sensor (the thermistor) in a small container or box with tiny holes for air circulation. Make sure the wires come out of the container.

**4. Write Down Readings:**

* Record the indoor temperature using a regular thermometer.
* Use your DIY sensor to measure the outdoor temperature by placing it outside. Make sure the wires come inside so you can connect them to your circuit.

**5. Observe and Compare:**

* Use a simple code (if using a microcontroller) to read the temperature from your DIY sensor.
* Compare the indoor and outdoor temperatures. Which one is colder or warmer?

**6. Experiment:**

* Try placing the outdoor sensor in different locations and see how the temperature changes.
* Can you predict how the temperature will change during different times of the day?

**Conclusion:** Congratulations! You've created a basic outdoor-indoor temperature sensor using simple components. You've learned how to build a circuit and use a thermistor to measure temperature differences. Keep experimenting and exploring the world of electronics and science!

# 2) Motion sensor using Arduino and basic components

**Introduction:** In this exciting project, you'll learn how to create a motion-activated alarm using an Arduino and a motion sensor. You can use this alarm to detect movement in a specific area, such as your room, and have it trigger an alert.

**Materials Needed:**

* Arduino board (e.g., Arduino Uno)
* Passive Infrared (PIR) Motion Sensor
* Breadboard and jumper wires
* Buzzer or LED
* USB cable to connect Arduino to a computer
* Computer with Arduino IDE installed
* Pencil and Paper

**Safety Precautions:**

* Always have adult supervision when working with electronics.
* Handle components gently to avoid damage.
* Avoid looking directly into the motion sensor when testing.

**Instructions:**

**1. Learn About the Components:**

* Have an adult explain what the components are and how they work:
* Arduino: A small computer used to control electronic devices.
* PIR Motion Sensor: Detects motion by measuring changes in infrared radiation.
* Breadboard: A platform to build and connect electronic circuits.
* Buzzer or LED: These will be used to create the alarm.

**2. Build the Circuit:**

* Connect the PIR motion sensor to the breadboard.
* Connect the VCC (power) and GND (ground) pins of the sensor to the corresponding pins on the Arduino.
* Connect the output pin of the sensor to a digital input pin on the Arduino (e.g., pin 2).
* Connect a buzzer or LED to another digital pin (e.g., pin 3) for the alarm.

**3. Write the Arduino Code:**

* Write a simple Arduino code that checks if motion is detected by the sensor and activates the buzzer or LED when motion is detected.
* You can use the Arduino IDE on your computer to write, compile, and upload the code to your Arduino board.

**4. Test the Motion Sensor:**

* Upload the code to your Arduino.
* Place the motion sensor in an area where it can detect motion (e.g., a doorway or a room).
* When motion is detected, the buzzer or LED should activate.

**5. Customize Your Alarm:**

* Experiment with different settings in the code to adjust the sensitivity of the motion sensor or the duration of the alarm.
* You can also add more components or features, such as a display screen or recording device, to make your alarm even cooler!

**6. Experiment and Have Fun:**

* Test your motion-activated alarm in different locations and see how it performs.
* Try making it more interactive by adding buttons to control the alarm or changing the alarm sound.

# 3) Infrared Sensor

**Introduction:** In this fun project, you'll learn how to create an alarm system using an Arduino and an infrared (IR) sensor. You can use this alarm to detect when an object or person gets too close to the sensor's field of view and trigger an alert.

**Materials Needed:**

* Arduino board (e.g., Arduino Uno)
* Infrared (IR) Sensor Module (e.g., HC-SR501)
* Breadboard and jumper wires
* Buzzer or LED
* USB cable to connect Arduino to a computer
* Computer with Arduino IDE installed
* Pencil and Paper

**Safety Precautions:**

* Always have adult supervision when working with electronics.
* Handle components gently to avoid damage.
* Avoid looking directly into the IR sensor when testing.

**Instructions:**

**1. Learn About the Components:**

* Have an adult explain what the components are and how they work:
* Arduino: A small computer used to control electronic devices.
* Infrared (IR) Sensor: Detects motion by sensing changes in infrared radiation.
* Breadboard: A platform to build and connect electronic circuits.
* Buzzer or LED: These will be used to create the alarm.

**2. Build the Circuit:**

* Connect the IR sensor to the breadboard.
* Connect the VCC (power) and GND (ground) pins of the sensor to the corresponding pins on the Arduino.
* Connect the output pin of the sensor to a digital input pin on the Arduino (e.g., pin 2).
* Connect a buzzer or LED to another digital pin (e.g., pin 3) for the alarm.

**3. Write the Arduino Code:**

* Write a simple Arduino code that checks if an object is detected by the IR sensor and activates the buzzer or LED when motion is detected.
* You can use the Arduino IDE on your computer to write, compile, and upload the code to your Arduino board.

**4. Test the IR Sensor:**

* Upload the code to your Arduino.
* Place the IR sensor in an area where it can detect objects moving in front of it.
* When an object is detected, the buzzer or LED should activate.

**5. Customize Your Alarm:**

* Experiment with different settings in the code to adjust the sensitivity of the IR sensor or the duration of the alarm.
* You can also add more components or features, such as a display screen or a camera, to make your alarm even more advanced!

**6. Experiment and Have Fun:**

* Test your IR sensor alarm in different locations and see how it performs.
* Try making it more interactive by adding buttons to control the alarm or changing the alarm sound.

# 4) Water Sensor integration

**Materials Needed:**

* Arduino board (e.g., Arduino Uno)
* Water level sensor (e.g., a float switch or capacitive sensor)
* Breadboard and jumper wires
* Buzzer or LED
* USB cable to connect Arduino to a computer
* Computer with Arduino IDE installed
* A container with water (to simulate different water levels)
* Pencil and Paper

**Safety Precautions:**

* Always have adult supervision when working with electronics.
* Avoid placing the electronics directly in water to prevent damage.

**Instructions:**

**1. Learn About the Components:**

* Have an adult explain what the components are and how they work:
* Arduino: A small computer used to control electronic devices.
* Water Level Sensor: A sensor used to detect the presence or absence of water.
* Breadboard: A platform to build and connect electronic circuits.
* Buzzer or LED: These will be used to indicate water levels.

**2. Build the Circuit:**

* Connect the water level sensor to the breadboard.
* Connect the sensor's output pin to a digital input pin on the Arduino (e.g., pin 2).
* Connect a buzzer or LED to another digital pin (e.g., pin 3) for indicating water levels.

**3. Write the Arduino Code:**

* Write an Arduino code that reads the sensor's output and activates the buzzer or LED based on the water level.
* You can use the Arduino IDE on your computer to write, compile, and upload the code to your Arduino board.

**4. Test the Water Level Sensor:**

* Upload the code to your Arduino.
* Place the water level sensor in a container with water to simulate different water levels.
* As the water level changes, the buzzer or LED should activate or deactivate accordingly.

**5. Customize Your Water Detector:**

* Experiment with different settings in the code to adjust the sensor's sensitivity or customize the alarm or LED behavior.
* You can also add a display screen to show the water level digitally or connect the Arduino to a computer to log water level data.

**6. Experiment and Have Fun:**

* Test your water level detector in different containers with varying water levels.
* Try using it to monitor a plant's water level or as a simple flood detector.

# 5) Display Integration

**Materials Needed:**

* Arduino board (e.g., Arduino Uno)
* Water level sensor (e.g., a float switch or capacitive sensor)
* Breadboard and jumper wires
* Buzzer or LED
* Liquid Crystal Display (LCD) module (e.g., 16x2 character LCD)
* USB cable to connect Arduino to a computer
* Computer with Arduino IDE installed
* A container with water (to simulate different water levels)
* Pencil and Paper

**Safety Precautions:**

* Always have adult supervision when working with electronics.
* Avoid placing the electronics directly in water to prevent damage.

**Instructions:**

**1. Learn About the Components:**

* Have an adult explain what the components are and how they work:
* Arduino: A small computer used to control electronic devices.
* Water Level Sensor: A sensor used to detect the presence or absence of water.
* Breadboard: A platform to build and connect electronic circuits.
* LCD Module: A display for showing information digitally.
* Buzzer or LED: These will be used for indicating water levels.

**2. Build the Circuit:**

* Connect the water level sensor to the breadboard.
* Connect the sensor's output pin to a digital input pin on the Arduino (e.g., pin 2).
* Connect a buzzer or LED to another digital pin (e.g., pin 3) for indicating water levels.
* Connect the LCD module to the Arduino using jumper wires. Make sure to connect the power and data pins correctly.

**3. Write the Arduino Code:**

* Write an Arduino code that reads the sensor's output, activates the buzzer or LED based on the water level, and displays the water level on the LCD module.
* You can use the Arduino IDE on your computer to write, compile, and upload the code to your Arduino board.

**4. Test the Water Level Sensor and LCD Display:**

* Upload the code to your Arduino.
* Place the water level sensor in a container with water to simulate different water levels.
* As the water level changes, the buzzer or LED should activate or deactivate accordingly.
* The LCD display should show the water level digitally.

**5. Customize Your Water Detector:**

* Experiment with different settings in the code to adjust the sensor's sensitivity, customize the alarm or LED behavior, or change the LCD display format.
* You can also add additional features like a sound alert when the water level is too high or too low.

**6. Experiment and Have Fun:**

* Test your water level detector with various containers and water levels.
* Use it to monitor a plant's water level or as a simple water level indicator for a tank.

# 6) Joystick and Touch Sensor

**Materials Needed:**

* Arduino board (e.g., Arduino Uno)
* Joystick module
* Capacitive touch sensor (e.g., TTP223)
* RGB LED (common cathode)
* Breadboard and jumper wires
* USB cable to connect Arduino to a computer
* Computer with Arduino IDE installed
* Pencil and Paper

**Safety Precautions:**

* Always have adult supervision when working with electronics.
* Handle components gently to avoid damage.
* Avoid exposing the touch sensor to moisture.

# 7) Window servo motor

**Instructions:**

**1. Learn About the Components:**

* Have an adult explain what the components are and how they work:
* Arduino: A small computer used to control electronic devices.
* Joystick Module: A control input device that provides analog values for both X and Y directions.
* Capacitive Touch Sensor: A touch-sensitive input device.
* RGB LED: A multi-color LED with a common cathode.
* Breadboard: A platform to build and connect electronic circuits.

**2. Build the Circuit:**

* Connect the joystick module to the breadboard.
* Connect the X and Y outputs of the joystick to two analog input pins on the Arduino.
* Connect the capacitive touch sensor to a digital input pin on the Arduino.
* Connect the RGB LED to three digital output pins on the Arduino, and make sure to use current-limiting resistors for each color (red, green, blue).

**3. Write the Arduino Code:**

* Write an Arduino code that reads the values from the joystick module and capacitive touch sensor to control the color and brightness of the RGB LED.
* You can use the Arduino IDE on your computer to write, compile, and upload the code to your Arduino board.

**4. Test the Joystick and Touch Sensor:**

* Upload the code to your Arduino.
* Move the joystick to change the LED color (e.g., left for red, right for blue, up for green).
* Touch the capacitive touch sensor to change the LED's brightness.

**5. Customize Your Interactive LED Controller:**

* Experiment with different code settings to adjust the LED color and brightness range, or add more features like color cycling or fade effects.
* You can also use the joystick as a controller for other actions, like playing music or moving a robot.

**6. Experiment and Have Fun:**

* Play around with your interactive LED controller to create different colors and brightness levels.
* Try integrating additional sensors or components to expand your project.

**Materials Needed:**

* Arduino board (e.g., Arduino Uno)
* Servo motor (e.g., SG90)
* Breadboard and jumper wires
* A window or a model window frame
* A lightweight panel (to represent the window)
* USB cable to connect Arduino to a computer
* Computer with Arduino IDE installed
* Pencil and Paper

**Safety Precautions:**

* Always have adult supervision when working with electronics.
* Ensure that the servo motor doesn't get blocked while operating.
* Be cautious when operating near actual windows.

**Instructions:**

**1. Learn About the Components:**

* Have an adult explain what the components are and how they work:
* Arduino: A small computer used to control electronic devices.
* Servo Motor: A motor that can be precisely controlled to move to a specific angle.
* Breadboard: A platform to build and connect electronic circuits.

**2. Build the Circuit:**

* Connect the servo motor to the breadboard.
* Connect the servo motor's signal (control) wire to a digital output pin on the Arduino (e.g., pin 9).
* Connect the servo motor's power (VCC) and ground (GND) wires to the appropriate Arduino pins.

**3. Mount the Servo and Window Panel:**

* Attach the servo motor to the window frame or a stand in such a way that it can move a lightweight panel (representing the window).
* Make sure the servo motor's movement can effectively open and close the window.

**4. Write the Arduino Code:**

* Write an Arduino code that controls the servo motor to open and close the window at desired angles.
* You can use the Arduino IDE on your computer to write, compile, and upload the code to your Arduino board.

**5. Test the Window-Opening System:**

* Upload the code to your Arduino.
* Use the code to open and close the window by specifying the angles.
* Test the system to ensure it works smoothly.

**6. Customize and Automate:**

* Enhance your project by adding automation features. For example, you can use a temperature sensor to automatically open the window when it gets too hot.

**7. Experiment and Have Fun:**

* Explore different angles and timings for opening and closing the window.
* Consider adding sensors for detecting rain or sunlight to further automate the system.

# 8) Solar and Windmill project

**Materials Needed:**

* Small wind turbine or windmill kit
* Small solar panel
* Charge controller (for the solar panel)
* 12V rechargeable battery
* DC-to-DC voltage regulator (e.g., LM7805) if needed
* Small LED or low-power device (e.g., a mini fan or LED light)
* Breadboard and jumper wires
* Arduino board (optional, for monitoring)
* USB cable to connect Arduino to a computer (optional)
* Computer with Arduino IDE installed (optional)
* Pencil and Paper

**Safety Precautions:**

* Always have adult supervision when working with electronics.
* Avoid touching moving parts of the windmill.
* Work in a safe location for wind and solar experiments.

**Instructions:**

**1. Learn About the Components:**

* Have an adult explain what the components are and how they work:
* Wind Turbine: Converts wind energy into rotational energy.
* Solar Panel: Converts sunlight into electrical energy.
* Charge Controller: Manages the charging of the battery from the solar panel.
* Battery: Stores energy for later use.
* DC-to-DC Voltage Regulator: Converts battery voltage to a usable voltage if needed.
* LED or Low-Power Device: Uses the generated energy for lighting or movement.

**2. Assemble the Windmill:**

* Assemble the windmill according to the manufacturer's instructions.
* Ensure it has a generator that produces electrical energy when the blades spin.

**3. Connect the Solar Panel:**

* Connect the solar panel to the charge controller.
* Connect the charge controller to the battery. Make sure to follow the correct polarity.

**4. Connect the Wind Turbine:**

* Connect the wind turbine's generator to the battery or voltage regulator.

**5. Connect the Load:**

* Connect the LED or low-power device to the battery or voltage regulator.

**6. Test and Monitor:**

* Place the windmill and solar panel in a location with sunlight and wind.
* Observe how the battery charges and how the LED or device operates based on available energy.
* You can use an Arduino to monitor and display energy levels and usage if desired.

**7. Customize and Expand:**

* Experiment with different wind speeds and sunlight conditions.
* Consider adding more batteries or devices to the system.

**8. Experiment and Have Fun:**

* Explore the concept of renewable energy by observing how wind and solar power can be harnessed to generate electricity.