AUGUST

Renewable Energy Innovation

This month focuses on exploring clean and renewable sources of energy. Students build and test systems like solar and passive cooling models to understand energy efficiency. They learn how innovative solutions can reduce dependence on non-renewable resources.

ACTIVITY 1: MINI SOLAR WATER HEATING SYSTEM

Materials Needed:

• Solar Energy Experiment Kit (from ATL kit) OR: Small solar panel (5V 500mA)
• Small submersible water pump (DC 5V)
• Clear tubing (~50cm)
• Plastic container × 2 (one painted black)
• LM35 or DS18B20 waterproof temperature probe
• Arduino Uno + LCD display
• Aluminium foil

LEVEL 2 — MECHANICAL + ELECTRONICS

STEP-BY-STEP INSTRUCTIONS:

PART 1: PREPARE THE SYSTEM

Components Needed

- 2 plastic containers
- Aluminium foil
- Black paint or black paper
- Small solar panel (5V)
- Submersible water pump (5V)
- Plastic tubing (~50 cm)
- DS18B20 temperature sensor
- Arduino Uno
- LCD display (I2C)
- Jumper wires

Step 1: Prepare Containers

- **Container A (Heater):**
  - Paint it black or line with black paper (absorbs heat)
- **Container B (Receiver):**
  - Line with aluminium foil (reflects heat)

Step 2: Fill Water

- Fill **Container A (black)** with water
- Place it in **direct sunlight**

PART 2: SETUP WATER CIRCULATION

Step 3: Connect Pump

- Attach tubing:
  - From Container A → Container B
  - From Container B → back to Container A
- Connect pump to solar panel:
  - When sunlight hits → pump runs → water circulates

(No Arduino needed for pump at this stage)

PART 3: TEMPERATURE MONITORING

Step 4: Insert Temperature Sensor

- Place DS18B20 probe inside water (Container A)
- Ensure probe is fully submerged

Step 5: Connect DS18B20 to Arduino

- Red → 5V
- Black → GND
- Yellow (Data) → Digital Pin **2**
- Add a **4.7kΩ** resistor between Data and VCC (pull-up resistor)

Step 6: Connect LCD (I2C)

- VCC → 5V
- GND → GND
- SDA → A4
- SCL → A5

PART 4: UPLOAD CODE

PART 5: TESTING

Step 7: Record Temperature

- Record:
  - **Starting temperature**
  - Temperature after **20 minutes**
- Ensure:
  - System is under direct sunlight
  - Pump is circulating water continuously

PART 6: ANALYSIS

Step 8: Calculate Heating Effect

- Temperature increase = Final temp – Initial temp
- Observe:
  - Is water heating faster with circulation?
  - What role does black surface play?

PART 7: REAL-WORLD APPLICATION

Step 9: Estimate Daily Usage

Assume:

- System heats ~5 litres per cycle
- Runs 4 cycles/day

Total = **20 litres/day**

Now think:

- Is this enough for a family of 4?
- How can capacity be increased?

PART 8: EXTENSION (ADVANCED CONTROL)

Step 10: Add Efficiency Control

- Add a relay module between pump and power
- Modify logic:
  - Pump runs only when temperature difference > 5°C

This improves:

- Energy efficiency
- System performance

VIDEO LINK 1: HOW SOLAR WATER HEATER WORKS?

VIDEO LINK 2: ARDUINO DS18B20 TEMPERATURE SENSOR TUTORIAL

VIDEO LINK 3: ARDUINO - DRIVING DC MOTOR (WATER PUMP) WITH RELAY

ACTIVITY 2: PASSIVE COOLING WINDOW SYSTEM WITH AUTO-CONTROL

Materials Needed:

Materials Needed:

- Cardboard box (A3 or larger)
- Wooden skewers or popsicle sticks × 8
- Thin cardboard strips for louvers (8cm × 3cm) × 8
- Servo motor (SG90) × 1
- LM35 temperature sensor
- Arduino Uno
- Jumper wires
- Hot glue gun

Level 2 – Mechanical + Servo Automation

STEP-BY-STEP INSTRUCTIONS:

PART 1: BUILD THE ROOM MODEL

Components Needed

- Cardboard box (shoebox size recommended)
- Cutter/scissors
- Skewers or sticks (for pivots)
- Cardboard strips (for louvers)
- Thread or thin wire

Step 1: Create the Room

- Take a cardboard box and seal all sides
- Cut a **window opening (15 cm × 10 cm)** on one side

Step 2: Make Louvers

- Cut **4 equal rectangular strips** (~15 cm × 2–3 cm)
- Insert a skewer through each strip horizontally (center line)
- Fix both ends of skewers into the window frame so each strip can rotate

Step 3: Link Louvers Together

- Tie threads/wires from each louver to a **single central point**
- Connect this point to the **servo motor arm**
- When servo rotates → all louvers rotate together

PART 2: ELECTRONICS SETUP

Components Needed

- Arduino Uno
- Servo motor (SG90 or similar)
- LM35 temperature sensor
- Jumper wires

Step 4: Connect LM35 Sensor

- VCC → 5V
- GND → GND
- Output → A0

Step 5: Connect Servo Motor

- Red → 5V
- Brown/Black → GND
- Yellow/Orange (Signal) → Pin 9

PART 3: UPLOAD CODE

PART 4: TESTING THE SYSTEM

Step 7: Simulate Heat

- Place a **lamp or light source** near/inside the box
- Observe:
  - Temperature rising (Serial Monitor)
  - Servo rotating louvers

Step 8: Observe Behavior

- Above **30˚C** → Louvers open (airflow allowed)
- Below **28˚C** → Louvers close

PART 5: COMPARISON EXPERIMENT

Step 9: Compare Two Setups

- Box A → With smart louvers
- Box B → Fully sealed

Heat both using the same lamp.
Record:

- Temperature after 10–15 minutes

Step 10: Analyze

- Which box is cooler?
- Calculate temperature difference:
  - Example: 35˚C vs 31˚C → **4˚C reduction**

VIDEO LINK 1: CONTROL SERVO MOTOR WITH ARDUINO

DESIGN THINKING MISSION: REFINE

THE CHALLENGE:

Your monthly mission: Improve your prototype based on testing, failures, and feedback.
In July, you built your first version.
Now your goal is to make it better, stronger, and more reliable.
This stage is about:

- Fixing what did not work
- Improving performance
- Making your solution closer to real-world use

WHAT TO SUBMIT:

Improved Prototype (Version 2):

Build an updated version of your solution with clear improvements

Version Comparison (V1 vs V2):

Show the difference between your July and August prototypes:
What changes did you make?
Why were these changes needed?

Build Evidence:
4–6 photos or video of the improved prototype
Highlight new or modified components

Performance Testing

Test your improved prototype at least 3 times and record:

- What improved compared to Version 1
- What still does not work

User Feedback (Second Round):
Show your improved solution to at least 2 users
What did they understand better?
What issues still remain?

Improvement Log:
List at least 3 specific improvements, such as:

- Better stability
- Better accuracy
- Easier usability
- Reduced cost

Limitations Identified:
What are the remaining challenges that still need to be solved?

Judging Criteria:

- Quality of Improvement (30%) — Are the changes meaningful and clearly justified?
- Performance Enhancement (20%) — Does Version 2 work better than Version 1?
- Testing Depth (15%) — Are improvements supported by testing evidence?
- User Feedback Integration (15%) — Has feedback influenced the improvements?
- Problem-Solution Alignment (20%) — Does the solution remain focused on the original problem?

Step	What to Do	How to Do it	Output
1	Review V1 Issues	Look at failure log and test results	Problem list
2	Identify Improvement	Decide 3 changes (design, efficiency, usability)	Improvement plan
3	Modify Design	Update structure/components	Updated design
4	Build Version 2	Implement improvements	Prototype V2
5	Document Changes	Capture photos showing differences	Evidence
6	Compare V1 vs V2	What changed and why	Comparison table
7	Test Again	Run 3 tests under similar conditions	Test results
8	Collect User Feedback	Show to 2 users, ask for input	Feedback notes
9	Evaluate Improvement	Did performance improve?	Analysis
10	Identify Remaining Gaps	What still needs fixing?	Limitations
11	Review V1 Issues	Look at failure log and test results	Problem list