JULY

Water Security & Resource Efficiency

This month focuses on understanding how water is used, wasted, and managed in everyday environments. Students explore real-world challenges like leakage and water quality through hands-on solutions. They learn how simple systems can improve water efficiency and conservation.

ACTIVITY 1: SMART WATER LEAKAGE ALARM

Materials Needed:

- Water/moisture sensor module (from ATL kit)
- Active buzzer (5V)
- Arduino Nano or Uno
- LED (Red)
- Jumper wires, breadboard
- 9V battery + clip

LEVEL 2 – SENSORS + AUTOMATION

STEP-BY-STEP INSTRUCTIONS:

PART 1: IDENTIFY PROBLEM AREAS

Step 1: Survey the School

Look for 3 possible leakage points, such as:

- Taps that drip
- Pipe joints
- Water coolers
- Roof edges during rain

Note where leaks are frequent or likely.

PART 2: CIRCUIT SETUP

Components Needed

- Arduino Uno
- Breadboard
- 3 LEDs (Red, Yellow, Green)
- 3 resistors (220Ω)
- Jumper wires
- USB cable + laptop

Step 2 : Connect Moisture Sensor

- VCC → 5V
- GND → GND
- Signal (AO) → A0

Step 3: Connect LED

- Long leg (+) → Pin 9 (through 220Ω resistor)
- Short leg (–) → GND

Step 4: Connect Buzzer

- Positive (+) → Pin 8
- Negative (–) → GND

PART 3: UPLOAD CODE

PART 4: CALIBRATION

Step 6: Test the Sensor

- Dip the sensor probes in water

Note the reading in Serial Monitor
LED and buzzer should turn ON

- Dry the sensor

Reading should drop
LED and buzzer should turn OFF
(Adjust threshold value if needed : 400-600 range)

PART 5: REAL WORLD TESTING

Step 7: Place the System

- Keep the sensor near :
  - A leaking tap
  - A damp surface

- Check:
  - Does trigger within **10 seconds?**
  - Is it sensitive enough?

PART 6 : IMPACT CALCULATION

Step 8: Estimate Water Savings

Assume:

- One leaking tap wastes ~10 litres/day

If installed at 10 points:

- 10 × 10 = 100 litres/day saved

Now think:

- Monthly savings = 3000 litres
- Yearly savings = 36,000 litres

PART 7: REFLECTION

Step 9: Think & Improve

- Where else can this be used?
  - Homes
  - Bathrooms
  - Farms
- How can you improve it?
  - Add automatic shut-off valve
  - Send alert to phone
  - Use wireless system

ACTIVITY 2: LOW-COST WATER QUALITY INDICATOR

Materials Needed:

- Turbidity sensor module (from ATL kit)
- Arduino Uno
- RGB LED or 3 single LEDs (Red, Yellow, Green)
- Resistors (220Ω × 3)
- Small transparent containers × 3
- Water samples: clear, muddy, soapy

LEVEL 2 - SENSORES + CONDITIONAL LOGIC

STEP-BY- STEP INSTRUCTIONS:

PART 1 : PREPARE WATER SAMPLE

Components Needed

- 3 transparent containers
- Clean water
- Muddy water (mix soil in water)
- Soapy water

Step 1: Create Samples

- Container 1 → Clean water
- Container 2 → Muddy water
- Container 3 → Soapy water

Ensure all containers are similar in size for fair comparison

PART 2: CIRCUIT SETUP

Components Needed

- Arduino Uno
- Turbidity sensor
- RGB LED OR 3 LEDs (Red, Yellow, Green)
- 220Ω resistors × 3
- Jumper wires
- Breadboard

Step 2: Connect Turbidity Sensor

- VCC → 5V
- GND → GND
- Output → **A0**
-

Step 3: Connect LEDs

- If using 3 LEDs:
- Green LED → Pin 9 (through resistor)
- Yellow LED → Pin 10 (through resistor)
- Red LED → Pin 11 (through resistor)
- All negative legs → GND

PART 3: CONNECT LEDS

PART 4: TESTING

Step 5: Test Each Sample

- Dip the turbidity sensor into each container
- Observe:
  - Sensor value (Serial Monitor)
  - LED color

Record results:

Sample	Sensor Value	LED Output	**Category**
Clean Water			
Muddy Water			
Soapy Water			

PART 5: CALIBRATION

Step 6: Set Baseline

- Use **clean water** as reference
- Note its sensor value
- Adjust thresholds if needed:
  - Clean water should trigger **Green**

DESIGN THINKING MISSION: PROTOTYPE

THE CHALLENGE

Your monthly mission: Build the first working version of your solution.

You have selected your best idea in June.
Now it is time to bring it to life.

This is not about perfection.

This is about making your idea real, even if it is rough, incomplete, or fails.

Focus on:

Testing the core concept
Understanding what works and what doesn't

WHAT TO SUBMIT:

- Prototype (Version 1):

Create a basic working model of your solution
(This can be physical, digital, or a combination)

- Build Evidence:

4–6 photos or a short video of your prototype
Show key parts/components clearly

- Materials & Components List:

List all materials used
Mention any ATL tools/components (if used)

- ·Working Explanation:

Explain: What your prototype is supposed to do
How it works (step-by-step)

- Test Results (Initial):

Test your prototype at least 2 times and document:
What worked
What did not work
Any unexpected outcomes

- Failure Log:

List at least 2 challenges or failures you faced while building:
What went wrong?
Why do you think it happened?

- Next Improvement Plan:

WHAT WILL YOU IMPROVE IN THE NEXT VERSION?

STEP	What to Do	How to Do It	Output
1	Break Idea into Parts	Identify components (sensor, structure, logic, etc.)	Component list
2	Plan Build	Draw rough layout of prototype	Build plan
3	Gather Materials	Collect ATL kit + local materials	Materials list
4	Build Prototype	Start assembling (focus on core function only)	Prototype V1
5	Document Build	Take photos/videos during building	Evidence
6	Test Prototype	Run 2 trials under real/simulated conditions	Test results
7	Record Results	What worked? What failed?	Test log
8	Identify Failures	List at least 2 problems	Failure log
9	Analyze Failures	Why did it fail? (design, material, logic?)	Insights
10	Plan Improvements	What will you fix next?	Improvement plan

JUDGING CRITERIA

- Prototype Completeness (25%) — Does the model demonstrate the core idea?
- Build Effort & Evidence (20%) — Is there clear proof of hands-on work?
- ·Functionality (20%) — Does any part of the solution work as intended?
- Testing & Learning (20%) — Are insights drawn from testing attempts?
- Iteration Thinking (15%) — Is there a clear plan for improvement