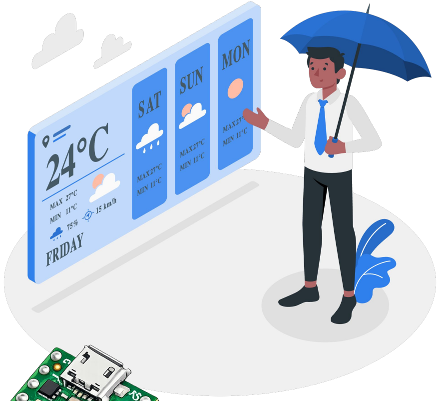
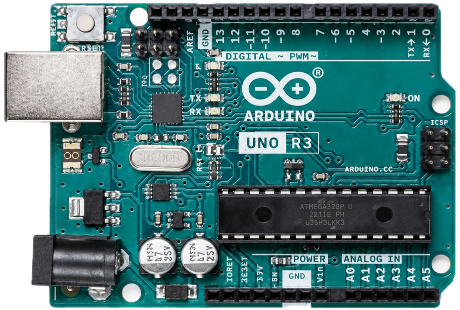
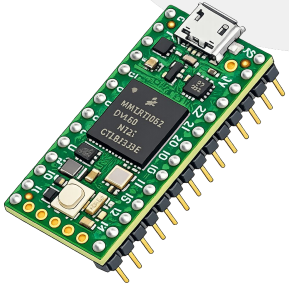
Open Arduino-Based Environmental Sensor Package for COSMOS Nodes:

Design, Integration, and Real-Time Environmental Monitoring

Presenter: Ben Katz







Project Team

Ben Katz
Undergraduate Researcher
Electrical Engineering,
Rutgers SOE

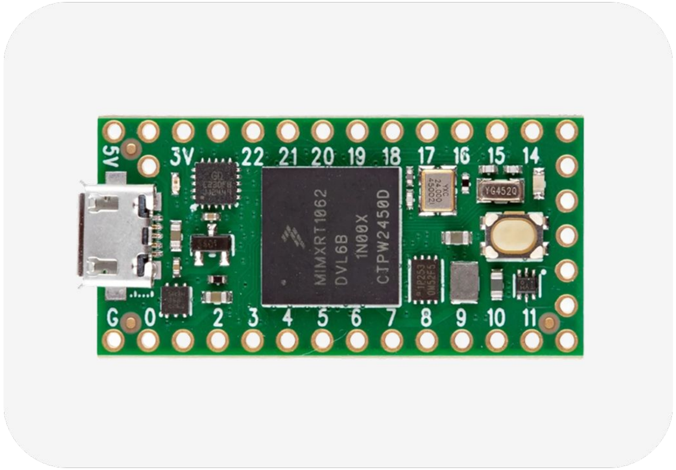
Hardware • Firmware • Data

Jakub Kolodziejski
Advisor
Senior Lab Engineer,
WINLAB Advisor

Mentorship • Integration

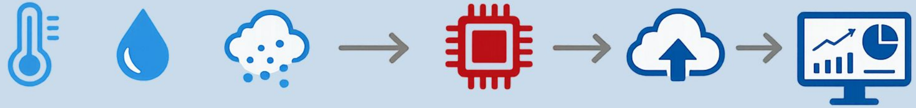
COSMOS Nodes & Microcontrollers

WIRELESS INFORMATION NETWORK LABORATORY



Programmable Research Infrastructure

Sensor Control & Data Collection



Microcontrollers enable environmental monitoring for COSMOS Nodes

Why Monitor the Environment?

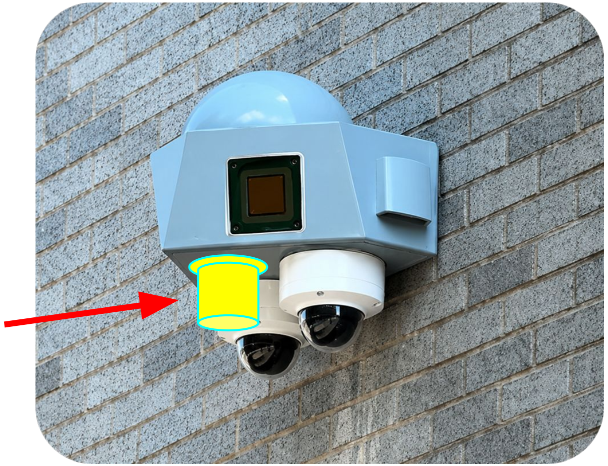
Add Environmental Context
Measure temperature, humidity, air quality, sound, and motion around each node.

Reveal Local Patterns
Correlate environmental conditions with wireless and camera-based activity.

Enable Scalable Monitoring
Low-cost, open-source sensors make deployment across multiple locations practical.

Locations of COSMOS Nodes at Columbia University (In Red)

Project Goals & Proposed Solution



Integrate

Connect multiple environmental sensors to one microcontroller.

Validate

Build and test a reliable working prototype.

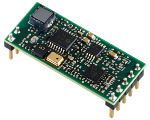
Package

Design a compact enclosure for future deployment.

Sensor Selection

WIRELESS INFORMATION NETWORK LABORATORY

Climate



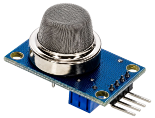
AmbiMate MS4
Temperature, humidity, light, TVOC, eCO_2, motion, and supply voltage

BMP085
Atmospheric pressure and estimated altitude

Particles

SPS30
Dust and airborne particles, including PM1.0, PM2.5, PM4.0, PM10, particle count, and typical particle size

Gases



MQ-7
Carbon monoxide and vehicle exhaust

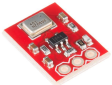
MQ-4
Methane and possible gas leaks

MQ-2
Smoke and combustible gases

MQ-3
Alcohol vapor

MiCS-2714
Nitrogen dioxide and traffic-related pollution

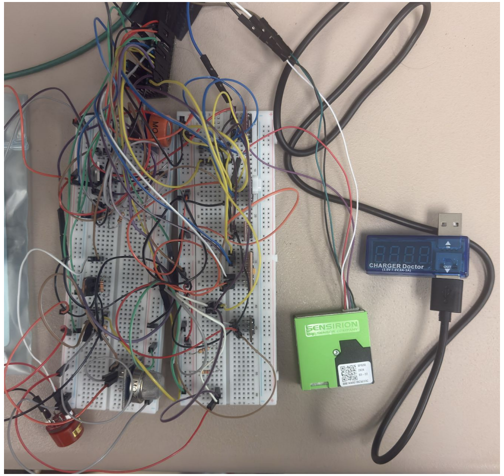
Sound



AmbiMate MS4 + BOB-09868
Ambient noise activity

Result 1:
Multi-Sensor Hardware Integration

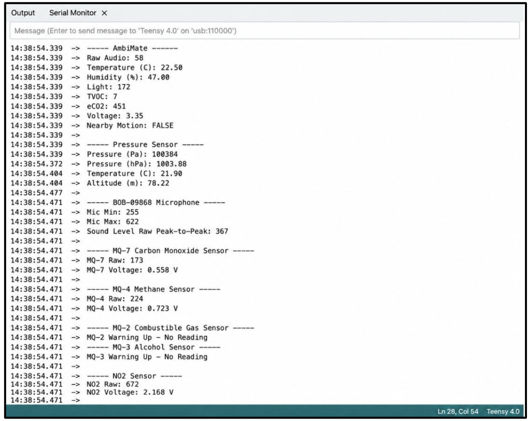
- Connected 10 environmental sensors to a Teensy 4.0
- Integrated communication interfaces
- Added transistor heater-control circuits for the MQ-series sensors
- Used voltage dividers to protect the Teensy’s 3.3 V analog inputs
- Verified simultaneous sensor operation through real-time data acquisition



Hardware Integration

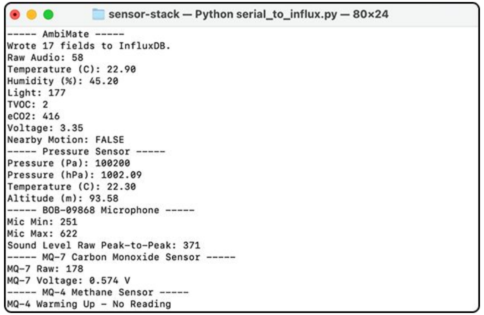
Result 2:

Real-Time Multi-Sensor Data Acquisition



Arduino IDE Output (C++)

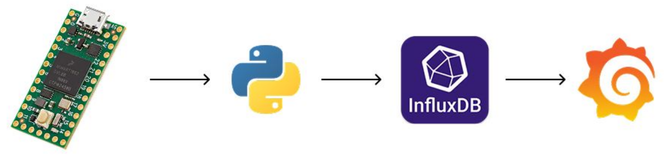
- Collected live measurements from the integrated sensor package
- Recorded temperature, humidity, pressure, light, sound, motion, and gas-sensor outputs
- Displayed MQ-series heater status and measurement cycles
- Verified that multiple sensors could operate within the same program
- Produced structured serial output for downstream data processing



Python Script (.py)

Result 3:
Real-Time Data Pipeline and Visualization

- Parsed live sensor data with Python
- Stored measurements in InfluxDB
- Visualized live data in Grafana
- Verified continuous Teensy-to-dashboard transfer



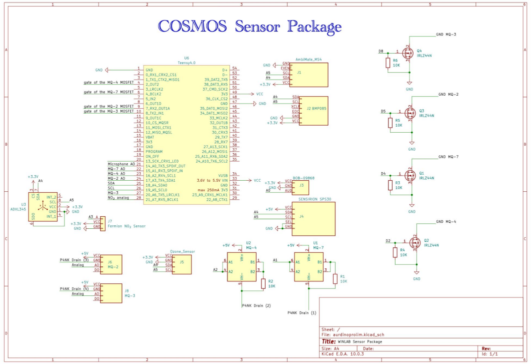
Result 4:
Compact Modular Enclosure Design

Sensor Station Inspiration

- Designed a compact, 3D-printable enclosure
- Added removable mounting and ventilation components
- Optimized airflow and sensor placement
- Reduced internal volume and simplified assembly
- Prepared the package for outdoor deployment

CAD Enclosure

Future Work and Recommendations



KiCad Schematic

Custom Printed Circuit Board Development (PCB)

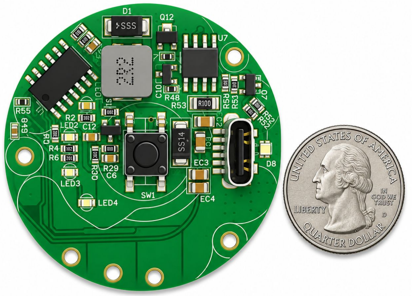
- Consolidate the modular architecture into a single integrated board
- Reduce wiring complexity and improve manufacturability

Sensor Validation and Calibration

- Validate measurements against reference data
- Improve long-term stability and accuracy

Future COSMOS Deployment

- Install and test the package at a Manhattan COSMOS node
- Evaluate real-world sensing performance



Custom PCB Example

QUESTIONS?