



Final Report

Aaditya Sharma, Bryce Pollak

April 29, 2025

WIRELESS SENSOR NODE G3

Introduction

The objective of this project was to design and implement a fully autonomous, solar-powered Wireless Sensor Node (WSN) capable of monitoring environmental conditions and transmitting data wirelessly to a remote base station. The core functionality centers around temperature and smoke detection, enabling early-stage fire monitoring in remote areas such as forests or fields. A key constraint of the system was that it must operate independently of external power or computing resources—relying solely on a solar energy input and a rechargeable battery for power, and using Bluetooth communication to transmit data.

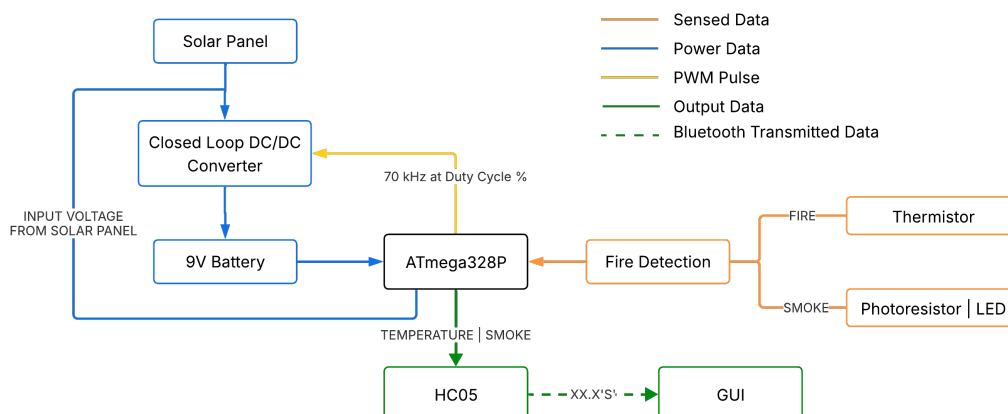
At the core of the design is the ATmega328P microcontroller, which handles all control, sensing, and communication tasks. To meet power autonomy requirements, a closed-loop DC/DC boost converter was developed to step up the 4–5V input from a simulated solar panel to a voltage suitable for charging a 9V battery at over 80% efficiency. The converter is regulated through a proportional-integral (PI) control algorithm implemented in software, allowing the system to maintain high power conversion efficiency. The battery powers the entire system, and the design ensures operation for over 6.5 days in power-saving mode—well beyond the minimum 5-day requirement.

To detect fire-related conditions, the node integrates a thermistor-based temperature sensor as well as an added custom-built smoke detection module using a photoresistor and LED. These sensors feed data to the microcontroller, which then transmits meaningful updates using an HC-05 Bluetooth module. A custom graphical user interface (GUI) on the receiving end displays the sensor readings in real-time and provides automated email alerts when danger thresholds are crossed.

In addition to meeting, if not exceeding, all core design specifications, several enhancements were implemented to improve system reliability, responsiveness, and deployability. These include optimized Bluetooth transmission, event-driven data handling to conserve power, and a custom 3D-printed enclosure for field use. The project demonstrates an effective integration of power electronics, embedded systems, sensor interfacing, GUI development, and wireless communication, resulting in a robust prototype suitable for real-world remote sensing applications.

High Level Design

The wireless sensor node (WSN) is built around the ATmega328P microcontroller, which serves as the central microcontroller responsible for sensing, control, communication, and power regulation. Every subsystem within the WSN is either powered by or communicates directly with the ATmega328P, making it the core of the entire design.



The system is divided into three key components: power, sensing, and transmission. Each component has been optimized to work cohesively to ensure reliable performance in a low-power, remote environment, such as a forest deployment scenario. The block diagram illustrates the data and power flow between components, clearly delineating the roles of input sources, processing units, and outputs.

Power

The WSN is designed to operate independently of external infrastructure, relying entirely on a solar panel and a 9V rechargeable battery. The solar panel provides an input voltage in the range of 0–5V, which is insufficient on its own to directly power the system or charge the battery. To address this, the voltage is passed through a closed-loop DC/DC boost converter, which is regulated using a PI (proportional-integral) control algorithm implemented on the ATmega328P.

The microcontroller samples the input voltage from the solar panel and calculates the appropriate PWM duty cycle to drive the DC/DC converter. The feedback loop ensures that the output voltage remains consistent, enabling efficient and safe charging of the 9V battery. The battery acts as the primary power supply for the entire circuit. The ATmega328P draws power through a voltage regulator to ensure stable operation, even as the battery voltage fluctuates during charge and discharge cycles.

Sensing

Environmental monitoring is carried out using a thermistor and a custom smoke detection circuit. For temperature sensing, we use the Steinhart approximation, an accurate method for calculating temperature from thermistor resistance. Our implementation is specifically optimized around 50°C—a temperature that exceeds typical forest conditions and may indicate an incoming fire. This optimization helps in early fire detection while avoiding false positives from normal environmental fluctuations.

Smoke detection is inspired by photointerrupter modules but implemented in a more accessible, low-power format using a photoresistor and an LED. The ATmega328P first records a baseline brightness level under clean air conditions. When smoke is present between the LED and photoresistor, the amount of light reaching the sensor decreases significantly. This drop in brightness is used as an indicator of smoke, triggering a detection event when it crosses a defined threshold. This complementary sensor enhances fire detection capability by combining thermal variation and smoke cues.

Transmission

To wirelessly transmit sensor data, the system uses an HC-05 Bluetooth module, which communicates over UART using the SoftwareSerial library. The system sends formatted data to a paired receiver whenever there's a significant change in environmental conditions.

To optimize power and bandwidth, data transmission is event-driven

- Temperature data is sent only if the change exceeds $\pm 0.5^{\circ}\text{C}$
- Smoke detection triggers immediate transmission

The transmitted data follows a format

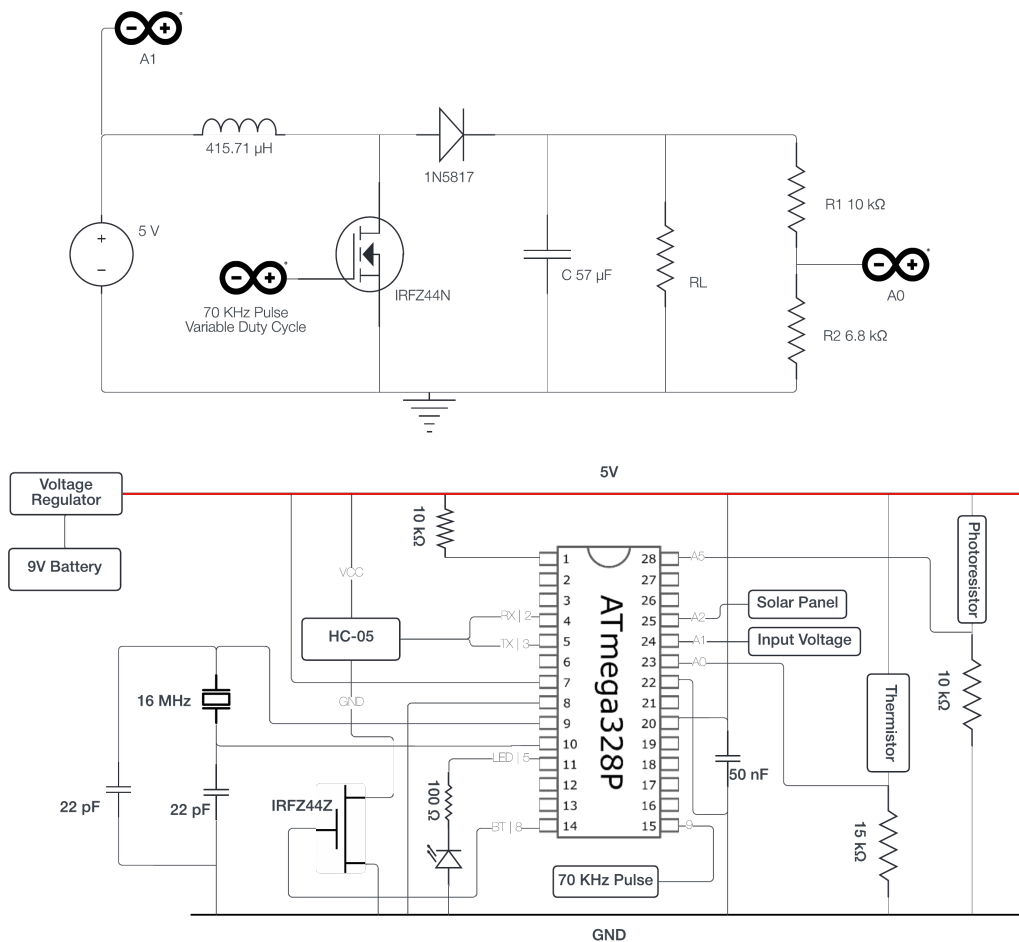
- Regular temperature readings are sent as XX.X
- If smoke is detected, the message becomes XX.XS, where S denotes smoke presence

On the receiver end, a custom-built GUI parses and displays the data in real time. It features a clean interface for temperature monitoring and dynamically updates in the presence of hazardous conditions. Importantly, the GUI

also includes an email-based alert system that notifies users remotely of potential fire events, thereby extending the safety benefits of the WSN even further.

This high-level design balances reliability, energy efficiency, and situational awareness, providing a strong foundation for remote environmental monitoring and alert systems. The tight integration between sensing, power management, and wireless transmission—centered around the ATmega328P—ensures that the system remains compact, responsive, and autonomous.

Detailed Design and Verification



Circuit Schematic

Closed Loop DC/DC Converter

Circuit Breakdown

- Solar Panel
 - Provides the input voltage (5V) for the circuit
- ATmega328P Pin 9
 - Provides a PWM (Pulse Width Modulated) gate signal to control the MOSFET

- Controls the duty cycle (D) of the switching MOSFET, which directly determines the boosted output voltage
- Inductor
 - Stores and releases energy to boost voltage
 - When the MOSFET is ON, current flows through the inductor, and energy is stored in its magnetic field
 - When the MOSFET is OFF, the inductor releases energy to the output capacitor and load, increasing the output voltage
- MOSFET - IRFZ44
 - Acts as a switch that alternates between conducting (ON) and blocking (OFF) states, controlling the inductor's energy storage and release
 - MOSFET ON: Current flows through the inductor, storing energy
 - MOSFET OFF: The inductor releases energy to the output through the diode
 - MAX $R_{DS(ON)} = 17.6 \text{ m}\Omega$
- Diode - 1N5817 Schottky Diode
 - Allows current to flow to the output capacitor and load when the MOSFET is OFF, preventing reverse current
 - MOSFET ON: Diode is reverse biased (blocks current)
 - MOSFET OFF: Inductor releases energy, and the diode conducts, supplying power to the output
 - Voltage Drop = 0.45 V (Under 1.0A)
- Capacitor
 - Smooths output voltage by filtering ripples
 - During MOSFET ON: The capacitor supplies current to the load
 - During MOSFET OFF: The capacitor stores excess energy from the inductor
- EBL 9V Rechargeable Ni-MH Battery 280 mAh

Parameters

- Input Voltage: 5V
- Switching Frequency: 70 kHz
- Desired Output Voltage: 10.5V
- Desired Output Current: 150 mA
- Output Voltage Ripple Factor: 1%

Calculations

Output Resistor

$$R = \frac{V_{out}}{I} = \frac{10.5}{150 \times 10^{-3}} = 66.66\Omega$$

For pulse for gate switching, switching period

$$T_s = \frac{1}{F_s} = \frac{1}{70k \text{ Hz}} = 12.29 \mu s$$

Duty Cycle

$$D = 1 - \frac{V_{in}}{V_{out}} = 1 - \frac{5}{10.5} = 1 - 0.4762 = 0.5238 = 52.38 \%$$

Inductance

$$L = \frac{V_{in}}{\Delta i} D T_s = \frac{5}{90 \times 10^{-3}} \times 0.5238 \times \frac{1}{70 \times 10^3} = 415.71 \mu H$$

Capacitance

$$C = \frac{V_{out}}{R \Delta v} \times D T_s = \frac{10.5}{66.66 \times 0.10} \times 0.5238 \times \frac{1}{70 \times 10^3} = 11.78 \mu F$$

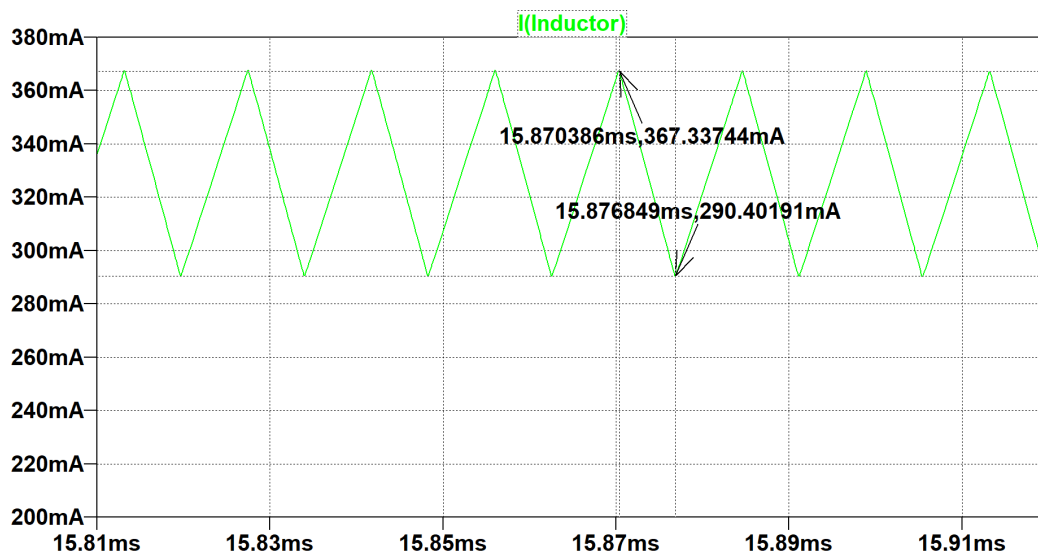
Used the specified 1% voltage ripple, calculated R, and the formula from slide 14 of converter slides. However, a capacitance of 11.78 μF might not be sufficient to reduce output voltage ripple as 1% voltage ripple is our maximum. Thus, we upped our capacitance by a magnitude of ~5 to be **57 μF** .

To build our inductor, we have $A_L = 6.9 \mu H$ Therefore $L = N^2 A_L$

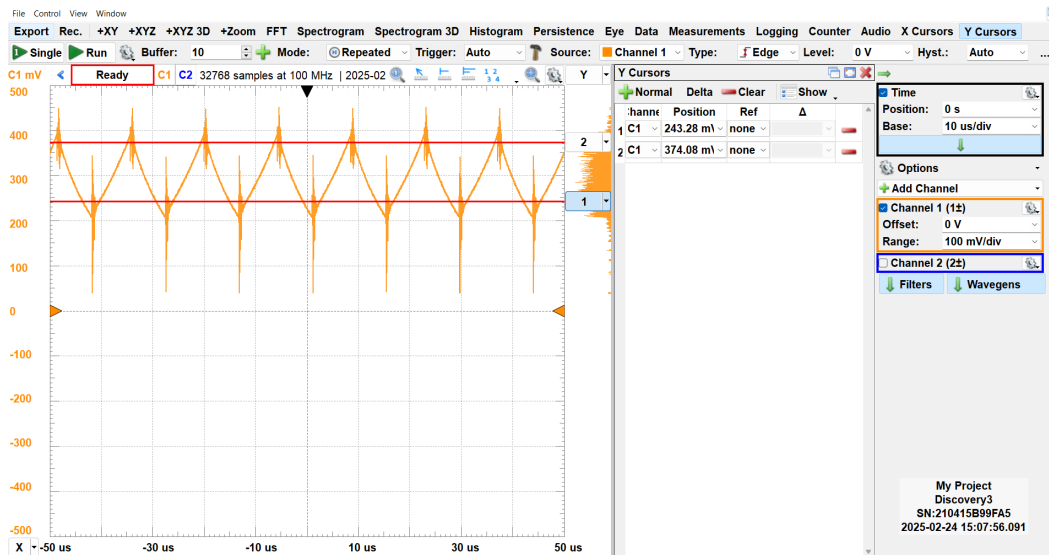
With our Inductance of 415.71 μH we need 7.76 turns, however since we needed higher EMF absorption to reduce inductor ripple we built a dual-core inductor which effectively doubles A_L so our new required number of turns is 5.49. Since we can't have fractional turns, we round it up to 6 coils and end up with an inductance of 496.8. Doubling our inductor core helps us reduce flux by manipulating N and I to prevent saturation. Reducing N reduces flux which in turn prevents saturation.



High ripple current ratios allow for smaller inductors but result in larger peak currents, increasing electromagnetic interference (EMI) and potentially stressing components. Maintaining continuous conduction mode (CCM) under such conditions also requires higher load currents. On the other hand, low ripple current ratios require larger inductors, which store more energy. If the load is suddenly disconnected, this energy can cause a voltage spike across the output capacitor, risking damage to the system. To balance size, stability, and performance, a ripple current ratio of 30–40% is generally optimal. Fine-tuning can also be achieved by adjusting the switching frequency.

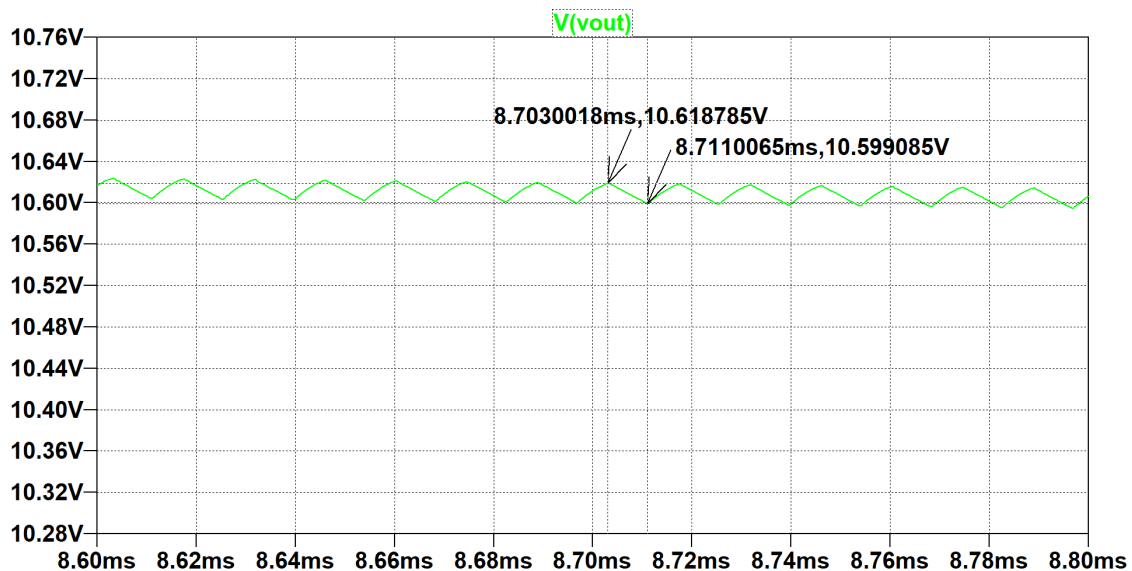


Simulated Inductor Current



Measured Inductor Current

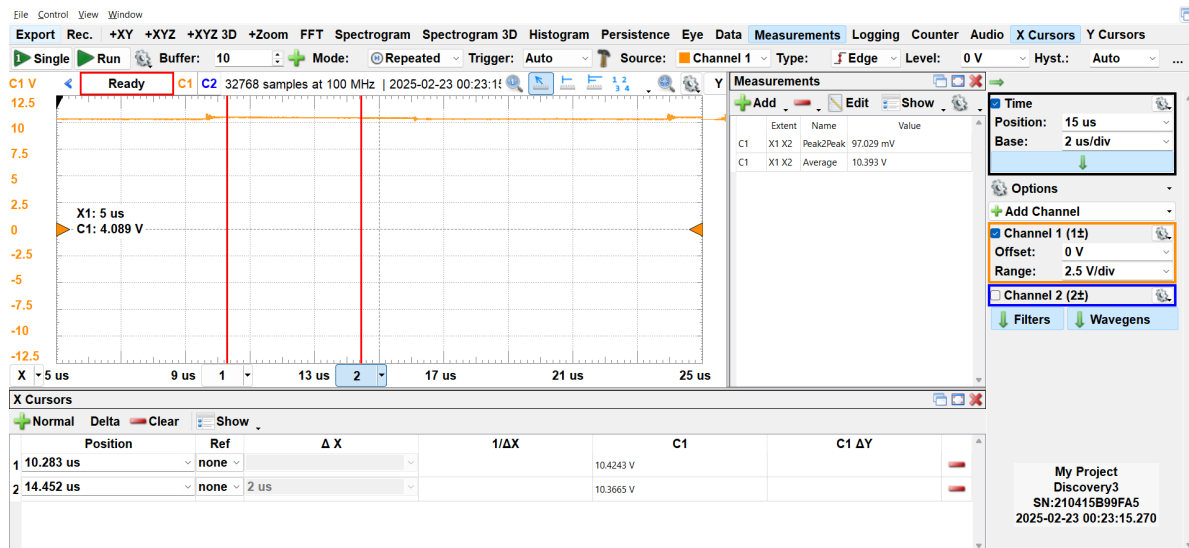
Cursors were used to measure the simulated inductor current ripple at 76.94 mA with an average current of 328.87 mA, yielding a predicted output current of approximately 164.44 mA—consistent with the expected 150 mA when accounting for minor voltage increases in the model. In the physical circuit, the peak-to-peak ripple measured 130.8 mA, with an average inductor current of 308.68 mA. This shows a higher ripple and slightly lower average current than the simulation, primarily due to real-world effects like core losses, which are not present in simulations. The ripple percentage was 23.39% in simulation versus 42.37% in practice—still within an acceptable range. Both the simulated and measured inductor current waveforms were triangular in shape, indicating minimal inductor saturation and confirming proper converter operation. The simulated output voltage oscillated between



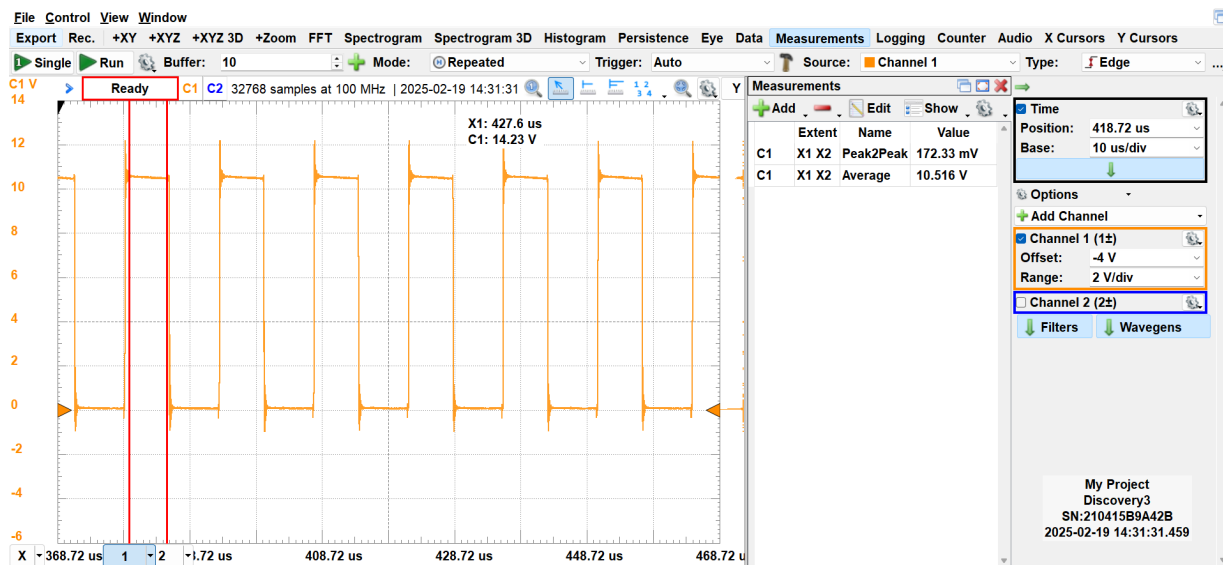
Simulated Output Voltage

10.619 V and 10.599 V, resulting in a 20 mV peak-to-peak ripple and an average voltage of 10.61 V—slightly above

the target range of 10–10.5 V. In the real circuit, the measured ripple was 97.03 mV, about five times higher than the simulation. Despite this, it remains below the 1% (100 mV) ripple threshold, meeting the design requirement. The measured average output voltage was 10.393 V, showing a 200 mV drop from the simulation and still within the desired range for safe and effective battery charging.



Measured Output Voltage



Drain Source Voltage (Added a 10 Ω resistor to reduce ringing)

Parameter	Simulated	Measured
Output Voltage Ripple (Peak to Peak)	0.019 V	0.097 V
Average Output Voltage	10.61 V	10.393 V
Inductor Current Ripple	76.94 mA	130.8 mA
Average Inductor Current	328.87 mA	308.68 mA
Inductor Current Ripple %	23.4%	42.37%
Drain-Source ON Voltage (Peak to Peak)	-	172.33 mV
Average ON Drain-Source Voltage	-	10.516 mV

Closed loop DC/DC converter

Our closed loop DC/DC converter dynamically adjusts the PWM duty cycle to regulate the output voltage using a proportional-integral (PI) control algorithm. The closed-loop adjustment ensures that V_{out} stabilizes at the desired level.

- Voltage Divider Circuit – Scales down V_{out} to a level measurable by the Arduino's ADC.
- Analog-to-Digital Converter (ADC) – Reads the scaled output voltage and converts it into a digital value.
- Error Calculation – Compares the measured V_{out} to the desired reference voltage (V_{ref}).
- Proportional-Integral Control Algorithm – Adjusts the PWM duty cycle dynamically based on both the instantaneous error and the accumulated past error for smoother regulation.
- PWM Output (MOSFET Gate Control) – Modulates the switching duty cycle to regulate V_{out}
- Boost Converter Circuit – Uses an inductor, diode, capacitor, and MOSFET to step up the input voltage.

How it Works

- Voltage Measurement
 - The Arduino reads V_{out} through the ADC.
 - This voltage is scaled up through a voltage divider to get V_{out}

$$V_{A0} = V_{out} \times \left(\frac{R_2}{R_1} + R_2 \right) \quad \rightarrow \quad V_{out} = V_{A0} \times \left(\frac{R_1 + R_2}{R_2} \right)$$

$$V_{out} \approx V_{A0} \times 2.471$$

- The voltage divider ensures that V_{out} is within the 0-5V range for accurate ADC readings.
- Error Calculation
 - The system calculates the voltage error
 - $\text{Error} = V_{ref} - V_{out}$
 - If V_{out} is **too low**, the duty cycle **increases** to provide more energy to the inductor
 - If V_{out} is **too high**, the duty cycle **decreases** to reduce energy transfer
- PI Controller-Based Duty Cycle Adjustment
 - Proportional Term ($K_p \times \text{Error}$)

- Provides an immediate response to correct deviations.
- Integral Term ($K_i \times \sum \text{Error}$)
 - Gradually adjusts the duty cycle to eliminate steady-state errors and reduce overshoot.
- The final duty cycle is computed as
 - $\text{Duty Cycle} = K_p \times \text{Error} + K_i \times \sum \text{Error}$
- Duty Cycle Constraints:
 - The updated PWM duty cycle is limited within a safe operating range to prevent overcompensation

Duty Cycle Variation

In the context of our boost converter, the duty cycle (D) represents the fraction of time the MOSFET switch is ON during each PWM cycle. It is defined as

$$D = \frac{T_{on}}{T_{on} + T_{off}}$$

In a boost converter, adjusting the duty cycle directly controls the output voltage (V_{out}). A boost converter works by storing energy in an inductor when the MOSFET is ON and then releasing that energy to the output when the MOSFET is OFF. The output voltage is determined by

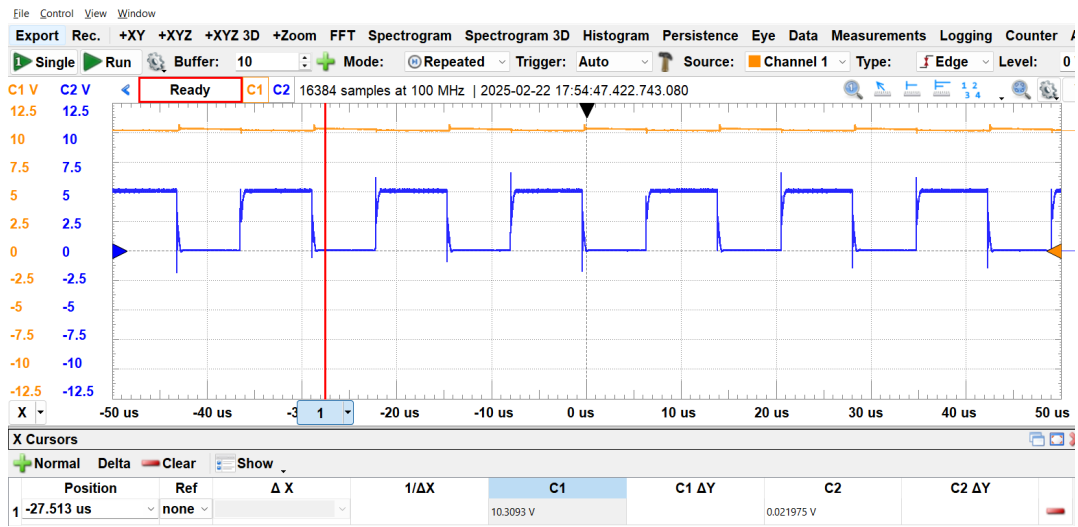
$$V_{out} = \frac{V_{in}}{1 - D}$$

From this equation, we can see that we can vary the Duty Cycle to maintain a constant V_{out} with changes in V_{in} .

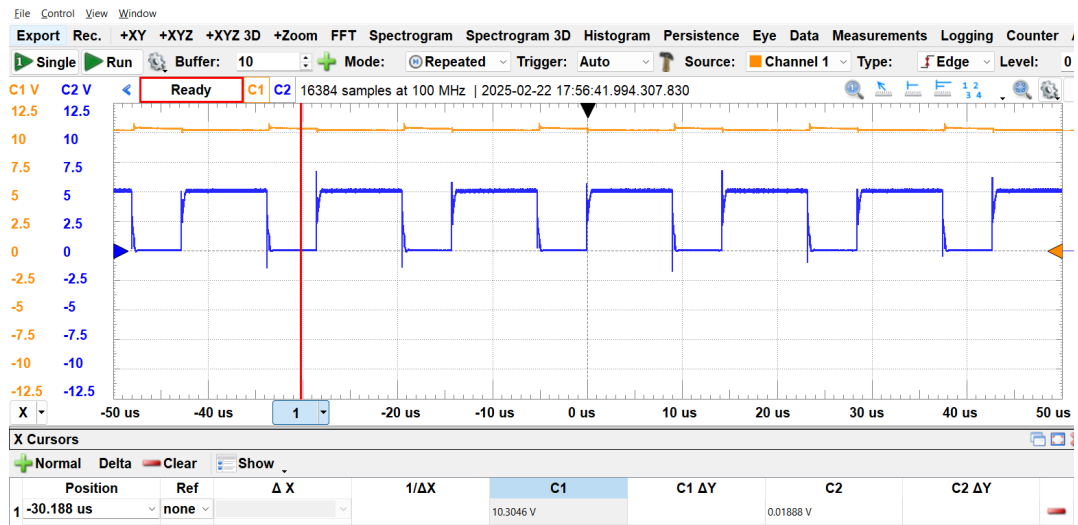
Change in Duty Cycle %	Change in V_{out}
Increase	Increases
Decrease	Decreases

Our experimental results clearly demonstrate the robustness and accuracy of our closed-loop DC-DC boost converter in regulating the output voltage V_{out} to the desired set-point. The waveforms for 5V, 4V, and even 3V input voltages provide conclusive proof that the system maintains voltage regulation irrespective of input variations. In the following waveforms, the target voltage is set at 10.3V.

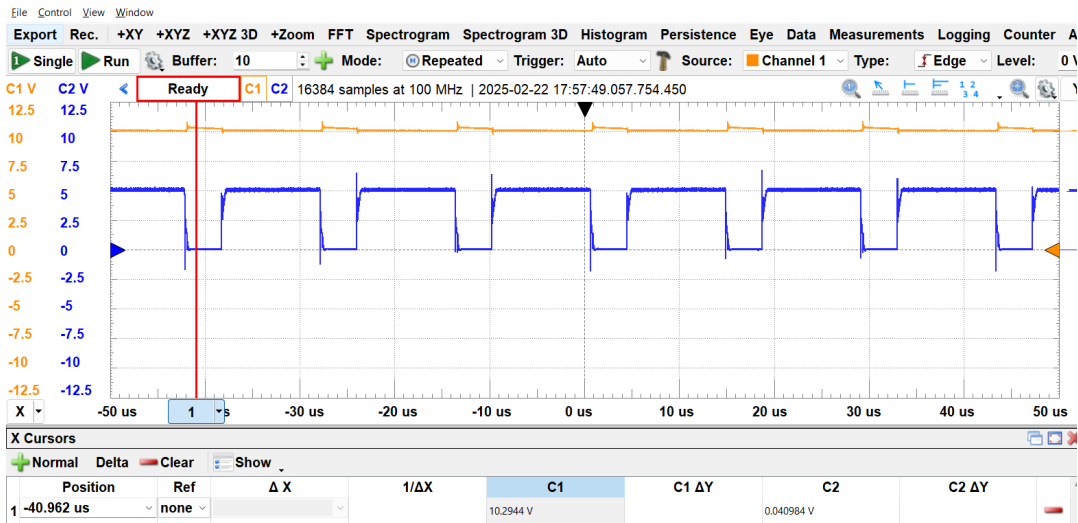
- Consistent Output Voltage (10.3V) Across Different Input Voltages
 - The captured waveforms clearly show that V_{out} stabilises at 10.3V, irrespective of input voltage variations.
 - Even when the input voltage is as low as 3V, the system effectively regulates the output, which exceeds the assignment requirement (4V-5V range).
 - This demonstrates excellent efficiency and adaptability, ensuring that the converter can function reliably even in suboptimal conditions.
- Stable Switching Waveforms with Proper Duty Cycle Adjustments
 - The blue waveform represents the switching signal at the MOSFET gate. The switching pattern remains consistent and stable, confirming the proper operation of the PI-controlled PWM adjustment.



Output Voltage at 5V Input

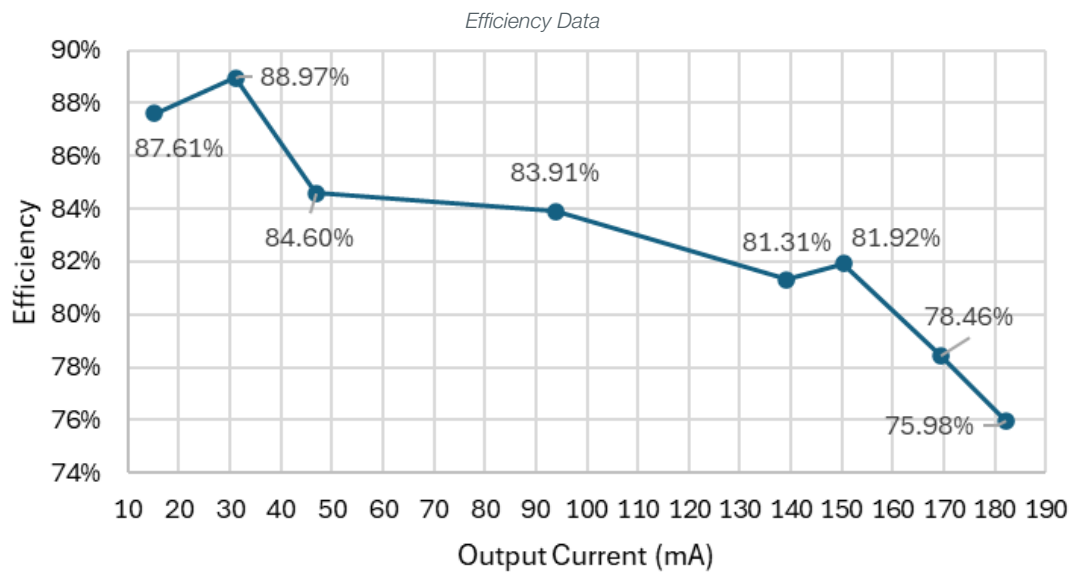


Output Voltage at 4V Input



Output Voltage at 3V Input

Load Resistance (Ω)	Vin (V)	Iin (A)	Input Power (W)	Vout (V)	Iout (A)	Output Power (W)	Efficiency
680	5.0103	0.035413	0.1774	10.28	0.015118	0.15542	87.610%
330	5.0072	0.07174	0.35933	10.271	0.031124	0.31968	88.966%
220	5.0012	0.11385	0.56959	10.296	0.0468	0.48189	84.603%
110	4.9874	0.23131	1.153	10.316	0.093782	0.96752	83.913%
73.33	4.9598	0.35283	1.7482	10.209	0.13922	1.4214	81.306%
68	4.959	0.3783	1.8759	10.222	0.15032	1.5367	81.918%
60	4.9567	0.44304	2.194	10.162	0.16937	1.7214	78.459%
56	4.9451	0.49496	2.4442	10.197	0.18209	1.8571	75.980%



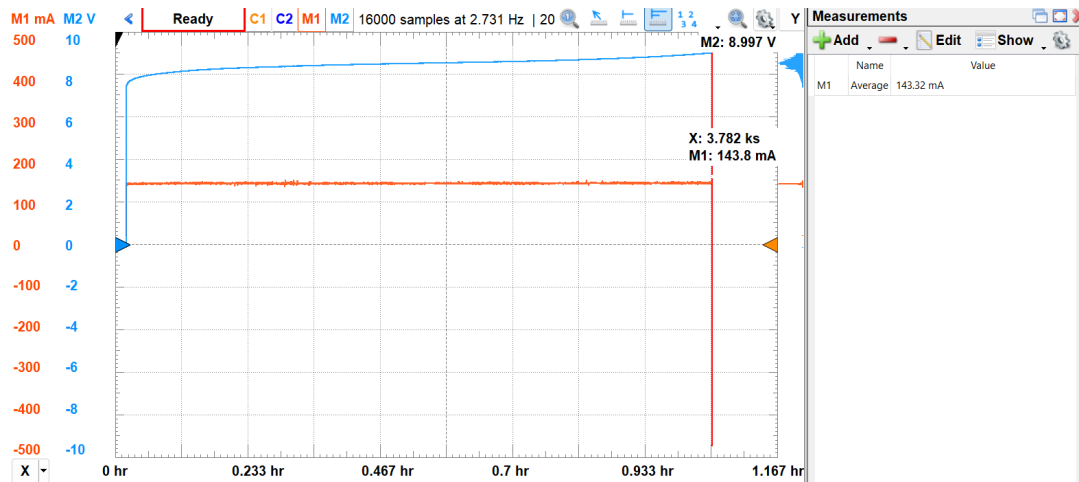
Efficiency across Output Currents

Efficiency Curve

Using the exponential model for efficiency $\eta = -1.637 \times e^{(11.281 \times I_{out})} + 89.394$

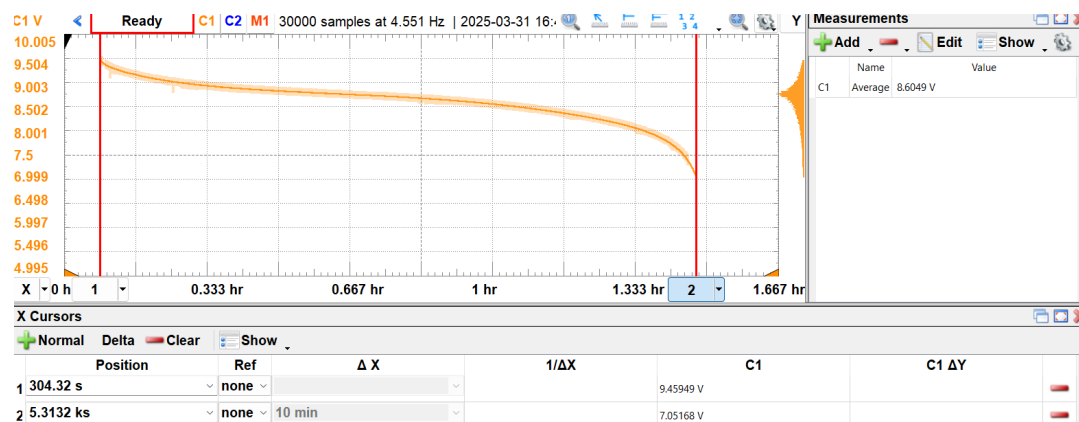
We can determine that the efficiency of our system at our chosen battery charging voltage of **120 mA - 140 mA** results in efficiency between **81.5% and 83.5%**, exceeding design requirements. This results in a charging time of **1.25 - 1.1 hours**, giving us a good balance between efficiency and charging time - making the most of limited access to power.

9V Battery



Battery Charging Waveform

$$\text{SUPPLIED INTO BATTERY} \rightarrow 144.32 \text{ mA} \times 1.061 \text{ Hrs} = 153.12 \text{ mAh}$$



Battery Discharging Waveform

$$I = \frac{8.605}{80} = 107.6 \text{ mA}$$

$$\text{DRAWN OUT OF BATTERY} \rightarrow 107.6 \text{ mA} \times 1.4 \text{ Hrs} = 150.64 \text{ mAh}$$

The battery was fully charged, then discharged across an 80Ω resistor until its voltage dropped to 7 V over 1.4 hours, with an average voltage of 8.605 V used to compute current. For charging, it was connected to the DC/DC boost converter with a 10Ω resistor to regulate current via PWM control, targeting 125 mA into the battery. Charging lasted 1.051 hours, reaching 9V, with an average current of 143.32 mA and total charge of 153.12 mAh.

Both discharging and charging waveforms are evidence of the system drawing and supplying 150 mAh out of and into the battery.

Fire Detection

Temperature Detection

Temperature is measured using a thermistor configured in a voltage divider circuit with a 15kΩ fixed resistor. To convert thermistor resistance into a temperature value, we used the Steinhart approximation. This model is preferred over linear approximations because it offers significantly greater accuracy across a wider range of temperatures. Specifically, we applied the simplified Beta formula, which is suitable for thermistors with a known Beta value. The equation used is

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta} \ln\left(\frac{R}{R_0}\right)$$

Where

- T is the absolute temperature in Kelvin
- T₀ is the nominal temperature (25° C)
- R is the thermistor resistance
- R₀ is the resistance at the nominal temperature (11.2 kΩ)
- β is the thermistors' Beta coefficient

These constants were derived from the manufacturer datasheet for our specific thermistor model. The temperature sensing algorithm was refined to improve accuracy in the early fire detection range of 40°C–60°C. This required carefully selecting the appropriate constants for the Steinhart Beta approximation, which are as follows

- Series Resistor = 15 kΩ
- Nominal Resistance = 4.5 kΩ
- Nominal Temperature = 50° C
- Beta Coefficient = 3950

These constants were selected based on the Murata NCP15XW153 thermistor's datasheet. While the default reference point for many thermistor systems is 25°C, this project required a shift in the accuracy window toward elevated temperatures, which are indicative of the early stages of fire or overheating. According to the datasheet, the thermistor's resistance is approximately 4.5kΩ at 50°C, making it a suitable reference point for nominal resistance.

Choosing 50°C as the nominal temperature balances the desired detection range and the thermistor's response curve. This ensures the Steinhart approximation is most accurate in the 40°C–60°C window, which is critical for detecting potential fire events before they escalate to dangerous levels. Lower reference temperatures (e.g., 25°C) would reduce accuracy in this region due to the logarithmic nature of the equation and thermistor response curve.

By aligning the nominal parameters with the application's target detection range, the system is better equipped to deliver fast and reliable temperature data that can be used to trigger alerts or responses when early signs of excessive heat are detected. Temperature sensing was verified using the thermostat to confirm accurate readings.



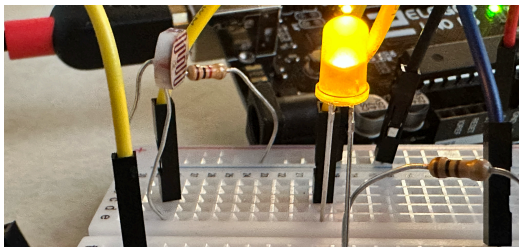
$78^{\circ}\text{F} \approx 25.5^{\circ}\text{C}$

```
Temperature: 25.75 °C
Temperature: 25.75 °C
Temperature: 25.84 °C
Temperature: 25.84 °C
```

Temperature Reading Verification

Smoke Detection

Inspired from photointerrupter modules used in smoke detection, we decided to build our own smoke detection sensor using a photoresistor and LED to work in tandem with the thermistor.



Custom Smoke Detection Sensor

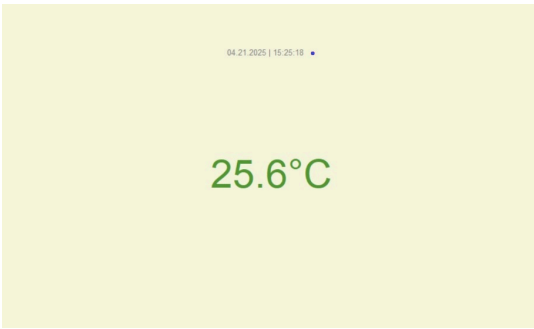
First we measured a base brightness value on the photoresistor. Then we measured the dip in observed brightness with smoke between the LED and photoresistor and established a smoke detection threshold. Whenever there is smoke in the room, the light from the LED is effectively dimmed and the WSN registers this as smoke detection. When smoke is detected, the character 'S' is pinged over Bluetooth along with the temperature data to alert the user of smoke, which can be another sign of an oncoming or spreading forest fire.

Bluetooth Transmission

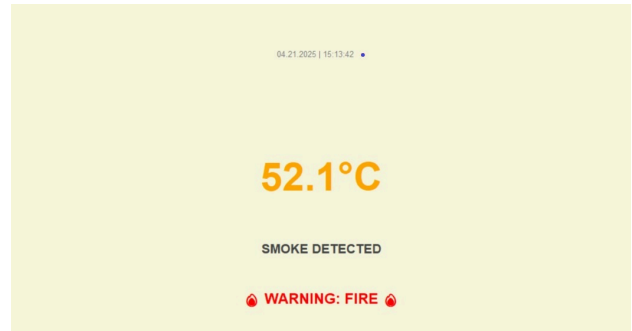
To transmit sensor data wirelessly, we used the HC-05 Bluetooth module. This module supports serial communication using the UART protocol. Data is transmitted over Bluetooth using the Software Serial library, as indicated by the project description.

GUI

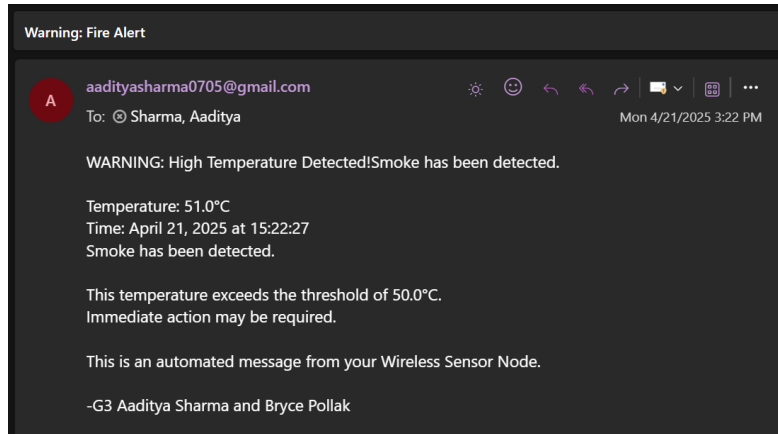
A custom-built Graphical User Interface (GUI) was developed to provide real-time visualization and alerting for the Wireless Sensor Node (WSN). Built using Python's tkinter framework, the GUI reads temperature and smoke data via a Bluetooth serial connection using the PySerial library. It displays the temperature in large, dynamically color-coded text and includes a live timestamp and connection status indicator (blue for connected, red for disconnected). When the temperature exceeds 50°C , the GUI issues a bold, flashing fire warning; if smoke is detected, a separate alert is triggered. One of the standout features is the automated email notification system, which sends safety alerts with sensor readings and timestamps when hazardous conditions are first detected. The interface is visually intuitive and functionally robust—making it a critical component in transforming raw sensor data into actionable insights for early fire detection.



Normal Temperature



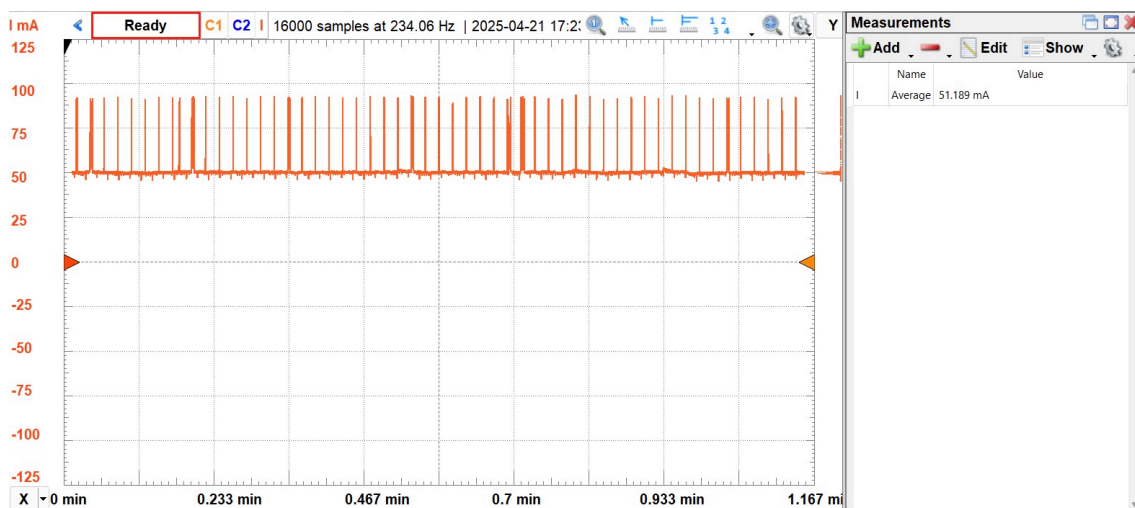
Over Threshold Temperature and Smoke Detected (Simulated with Small Flame)



Automated Emergency Email Alert

Power Efficiency

Power efficiency was a central focus in the development of our Wireless Sensor Node (WSN), as the system is designed to operate in remote or low-maintenance environments where battery longevity is critical. To ensure the WSN could run continuously for at least five days on a single full charge, we targeted a maximum average current



Current Draw Waveform on Standard Mode

draw of 1.25 mA, given our 150 mAh battery capacity. Achieving this goal required careful implementation of deep sleep modes, selective peripheral activation, and adaptive sampling intervals, all aimed at minimizing energy consumption while maintaining core sensing and communication functionality.

Runtime duration is calculated by

$$\text{Runtime in days} = \frac{\frac{150mAh}{I_{draw}}}{24}$$

Where I_{draw} is the current draw out of the battery.

The above waveform shows the current draw out of the battery in active mode. At this stage no power saving strategies were implemented. This gives a good foundation to start off and compare how power saving strategies affect power consumption.

With this level of current draw, the system would only operate for 3 hours, extremely far away from the 5 day runtime goal.

Power Saving Strategies

Aside from stripping the Arduino UNO down to a Standalone ATmega circuit, the following are the power saving strategies we implemented to maximise efficiency. No libraries are used, all strategies are implemented by directly writing to the ATmega registers. The system shifts into power saving mode whenever voltage from the solar panel drops below a threshold voltage, which is set to be 3.5V. The power saving strategies are in order of how much they increase power efficiency.

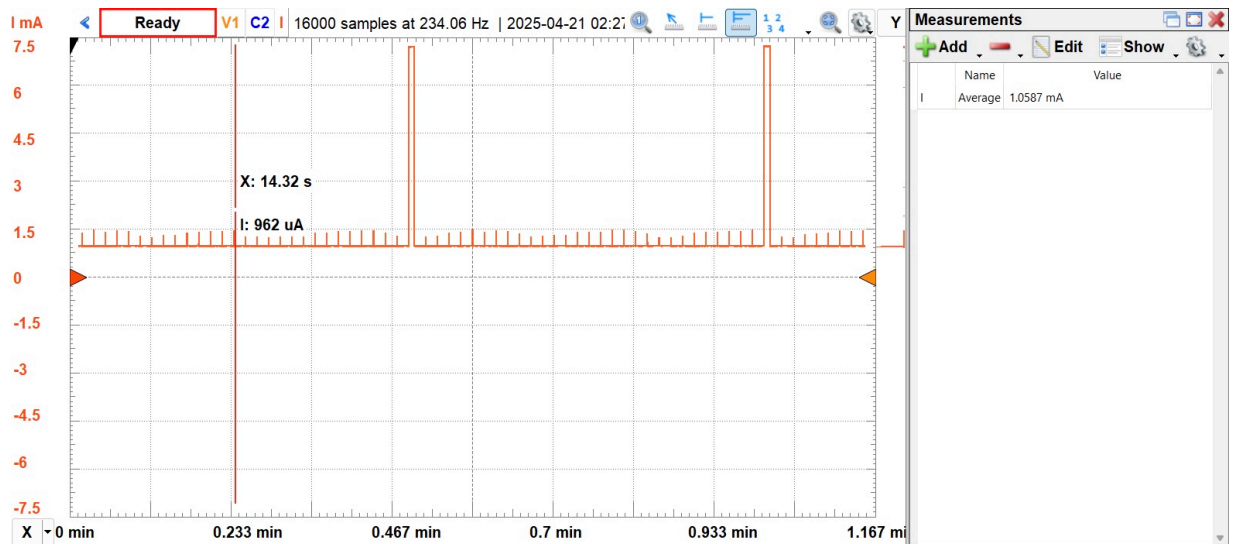
- Bluetooth Power Control
- ADC Power Management
- Sleep Mode Implementation
- Watchdog Timer As Wake Source
- PWM Optimization
- Sleep-to-Active Ratio Optimization

These combined techniques create a highly power-efficient implementation for the ATmega328P that extends battery life significantly, particularly important for solar-powered applications where energy conservation during low-light conditions is critical.

By combining these techniques, the system achieves remarkable power efficiency

- High-power components like Bluetooth are only activated when needed and disabled immediately after
- The ADC is only powered during readings
- The CPU sleeps in ultra-low power mode most of the time
- The WDT provides low-power, periodic wake-ups
- The system dynamically adapts its power usage based on available power

These methods collectively reduce power consumption by several orders of magnitude during low-power conditions, extending battery life significantly while maintaining essential functions.



Current Draw Waveform on Power Saving Mode

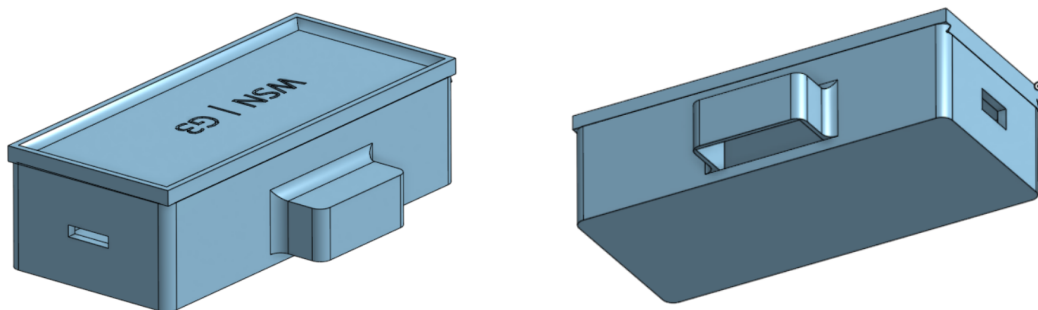
This waveform demonstrates the effectiveness of our power-saving **strategies**. In power-saving mode, the system's average current draw drops to just 1.05 mA, enabling a runtime of nearly six days on a single 150 mAh battery charge—already exceeding our design requirements.

Moreover, the waveform was captured under dynamic conditions, with temperature fluctuations simulated using a flame and ice. While the system is specified to transmit data every 30 seconds, our logic ensures that transmissions only occur when the temperature deviates by more than 0.75°C. As a result, the system remains in standby mode for the majority of the time, avoiding unnecessary transmission when there is no meaningful data to send. This reduces the average current draw closer to 0.962 mA, extending the estimated runtime to over six and a half days.

Custom Deployable Enclosure

To achieve deployability and usability, we designed, modeled, and 3D-printed a custom enclosure for our Wireless Sensor Node (WSN). This protective case enables deployment in a variety of environments while maintaining reliable access to temperature and smoke detection data.

The enclosure features cutouts for the HC-05 Bluetooth module to ensure unobstructed bluetooth communication. It also includes passthrough slots for the solar panel's power cables. Additionally it has an extended open pocket for the thermistor, allowing accurate temperature readings while also serving as a smoke collection chamber to allow for effective smoke detection.



Validation of the Overall Project

Feature	Implementation	Specification
Inductor Current	Triangle Shaped Acceptable Current Ripple Non-Saturating	Meets Specifications
DC/DC Closed Loop Converter	Converter outputs desired voltage accurately from 3V to 5V PI Controller Implemented	Exceed Specifications
Converter Efficiency	83% - 85% Efficient at battery charging current	Exceed Specifications
Battery Charge and Discharge	Waveforms show 150 mAh supplied into and draw out of the battery	Meets Specifications
Temperature Detection	Steinhart Approximation used for accurate temperature detection Optimized at 50°C Verified with commercial thermometer	Exceed Specifications
Power Saving Mode and Efficiency	Switches to Power Saving Mode when input voltage drops below threshold Current draw falls below 1.25 mA Runs for 6.5 days on a full charge	Exceed Specifications
Data Transmission and Display	Transmits data through HC-05 successfully Custom GUI reads and displays received data	Exceed Specifications

The Wireless Sensor Node (WSN) project was successfully implemented and validated through extensive measurements and system-level testing. All core deliverables specified in the project requirements were met or exceeded, and several additional features were integrated to enhance functionality, responsiveness, and deployability. Our system performs reliably under real-world constraints and more than achieves its design objectives.

Beyond the core requirements, we implemented several enhancements that significantly improved the WSN - from efficiency and user interface to deployability.

Extra Feature	Implementation
PID Control System for Efficient Charging	Proportional-Integral control for closed loop DC/DC converter duty cycle
Custom Smoke Detection System	Custom smoke sensor using photoresistor and LED
Optimised Bluetooth Transmission	Only transmit meaningful data - if temperature differential exceeds threshold or smoke is detected
Custom Built GUI with Warning Email Notifications	Warning email notifications with current temperature, timestamp and smoke detection when danger threshold is crossed
Custom Deployable Enclosure for WSN	Custom 3D modelled and printed enclosure

The final prototype integrates all hardware and software components into a cohesive, autonomous system. It successfully performs temperature and smoke detection, executes closed-loop voltage regulation, and communicates sensor data to a responsive user interface. The system also exhibits strong energy efficiency and longevity on battery power alone, making it a viable candidate for remote deployment in forest monitoring and other environmental sensing applications.

No shortcomings were observed. All critical functionality was validated through waveform and power measurements, and GUI feedback. As such, the project not only meets all outlined specifications but also surpasses them through thoughtful feature expansion and robust design.

Validation

This project provided a comprehensive, hands-on experience in designing and building an autonomous wireless sensor node—integrating concepts from power efficiency, embedded systems, sensing, and wireless communication into a cohesive and functional prototype. Beyond achieving all core specifications, the project challenged us to think creatively, troubleshoot systematically, and push our understanding of both hardware and software design.

One of the most valuable takeaways was a deepened understanding of power efficiency. Before this project, our exposure to DC/DC converters and feedback regulation was largely theoretical. Designing a closed-loop boost converter from scratch, implementing proportional-integral control, and validating system behavior through real measurements solidified these concepts in a practical context. We also gained insight into efficiency optimization, component selection, and power loss mitigation—key considerations in any energy-constrained embedded system.

On the embedded systems side, the project reinforced our skills in low-power microcontroller programming and peripheral interfacing. Working with a standalone ATmega328P forced us to think carefully about power draw, sleep modes, and efficient sensor polling. Implementing software-based UART communication, condition-driven data transmission, and GUI integration further developed our understanding of real-time system constraints.

Equally important were the engineering process skills we developed. From debugging erratic waveforms to identifying issues in our circuit, we learned the value of methodical troubleshooting and test-driven iteration. This experience has emphasized the importance of precision in circuit construction, thorough documentation, and hardware protection—lessons that are essential for any future hardware development project.

If we were to approach this project again, we would place greater emphasis on component protection and validation before integration. Several delays resulted from damaged parts due to overlooked overvoltage conditions or incorrect wiring during early tests. Implementing safeguards such as overvoltage protection diodes, current-limiting resistors, and more rigorous pre-power checks would have reduced delays considerably.

In hindsight, this project was more than an academic exercise—it was a realistic simulation of a professional engineering workflow. It demanded coordination, critical thinking, and the ability to turn design goals into a reliable product. These are the kinds of challenges we expect to encounter in industry, and this project has given us a solid foundation to face them with confidence.

Appendix

Code: <https://github.com/theGeekyBoi/IDP-WSN-G3>

Validation Video: <http://tiny.cc/WSNG3>

Authorship Page

- Introduction - Written by Bryce Pollak | Refined by Aaditya Sharma
 - High Level Design - Written by Aaditya Sharma
 - Detailed Design and Verification - Written by Aaditya Sharma and Bryce Pollak
 - Validation of the Overall Project - Written by Aaditya Sharma
 - Conclusion - Written by Bryce Pollak | Refined by Aaditya Sharma
-