Jaden Williams

SNHU – CS499

8/3/25

Milestone Four

These two lines need to be installed on a raspberry Pi (If it is the system used):

pip install pymongo[srv] adafruit-circuitpython-dht

sudo apt install -y mongodb

And once that is ensured, I must make sure that MongoDB is running locally with:

sudo systemctl start mongodb

sudo systemctl enable mongodb

Finally, (this step is technically before the 2nd step) this is a python script needed to run once to set up the database properly:

from pymongo import MongoClient

client = MongoClient("mongodb://localhost:27017/")

db = client["reptile\_env"]

reptiles = db["reptiles"]

# Example species document

reptiles.insert\_one({

"species": "Leopard Gecko",

"temp\_min": 28, # degrees Celsius

"temp\_max": 32,

"humidity\_min": 30, # percentage

"humidity\_max": 40 })

The artifact is a Python-based thermostat control system developed for a Raspberry Pi, originally created during CS350: Embedded Systems. The project was developed in Spring 2025 and is designed to read real-time temperature and humidity data from sensors, control a heating or cooling mechanism using GPIO pins, and display status information on an LCD. It also supports user input via buttons to cycle thermostat modes and adjust temperature setpoints.

I selected this artifact because it showcases my ability to integrate embedded hardware control with database-driven logic, creating a system that is adaptable to multiple real-world scenarios. It demonstrates my skills in:

* Designing modular, object-oriented Python code for hardware interaction.
* Connecting and querying a local MongoDB database.
* Implementing dynamic setpoint control logic based on stored environmental profiles.
* Interfacing user input (GPIO buttons) with system configuration in real time.

Compared to the original, the improved version now allows for automatic profile loading from MongoDB, species cycling via a button press, and targeted environmental control without hardcoded setpoints. This makes the system more scalable, maintainable, and practical for specialized use cases such as reptile enclosures.  
 I met my planned course outcomes for this enhancement. Specifically, I demonstrated the ability to design and evaluate a computing solution that solves a real-world problem (environmental control for reptiles) using algorithmic principles and appropriate computer science practices. I also applied innovative techniques by combining embedded hardware programming with database-driven configuration. My outcome-coverage plans remain the same; this enhancement further reinforces my ability to use tools and techniques to deliver industry-relevant solutions.  
 Through enhancing this artifact, I learned how to integrate a lightweight database into an embedded control loop without introducing excessive latency or complexity. I gained experience structuring MongoDB queries and handling environmental configuration data in a way that is flexible for future expansion.

One challenge was ensuring that the system remained responsive to both sensor readings and user inputs while performing database lookups. Another challenge was designing the system so that it could fail gracefully if the database was unavailable. I addressed this by keeping the hardware control logic separate from the database interface, allowing for fallback defaults. Overall, the enhancement deepened my understanding of how to combine software engineering practices with embedded systems development to create adaptable, data-driven solutions.