



Assignment

Capstone Final Project

This is the capstone project to summarize all the learnings in the 3 modules of the course and to give you practical hands-on experience by working on a major project.

Problem Statement:

In the capstone project for the IoT training, we require you to build a proof of concept of the lighting system of a refrigerator which uses data from the Light Dependent Resistor (LDR) and a push button as well as features of the Bolt IoT cloud. You need to use the light intensity and button state to collect light intensity and button data to decide the state of the fridge:

- a) Door open: Bright and button released
- b) Door closed: Dark and button pressed
- c) Door half open: Dim and button released

In your code, send an e-mail with the current state of the door when the state changes. Additionally, irritate the user once every 10 seconds if the door is open (state a and c). Additionally, you should set the intensity of a single indicator LED which clearly indicates the above three states.

Task 1 : Plan and test hardware

Read the datasheets of the hardware components and plan out your circuit. Make a block diagram or a schematic and test the circuit in parts based on the content covered in the course.

Task 2 : Plan and test software

Do not start coding right away. A good practice is to first break down the challenge into smaller sub-parts. First, figure out all the sub-parts of the problem to be solved and to work out the overall flow on pen and paper. Also figure out a way to create the three cases. Thereafter, implement small tests to keep solving bits and pieces of the overall problem.

Bonus challenge: Machine Learning

Note: This challenge is not graded and is only for those who dare to venture to the land of dragons. This challenge requires you to learn a lot of things outside of the scope of the training but builds up on the topics covered. Each dragon warrior will get a Bolt T-shirt if you successfully complete the bonus challenge covering all the points.

1. Use TensorFlow (or similar) library for Machine Learning by Google to build a classifier which dynamically adapts the thresholds for the LDR for classification of the brightness data (for state a and c) into the above environment categories instead of using hard-coded values. Thus, the system should function for different ambient lighting conditions, independent of the way you are using for triggering the LDR. (Hint: Look into Anomaly Detection for IoT).
2. Document the entire flow on GitHub with a detailed commit history as well as on a detailed blog or on a Hackster.io post
3. Be able to justify original work, with own style of documentation and detailed testing and not copied from another implementation on GitHub for the same.