Saunf Dispenser Developer Documentation

Developer and Designer: Sama Shah / 12/08/2023

Table of Contents

Product Overview

The saunf dispenser is an IoT-enabled vending machine that dispenses fennel seeds ("saunf") as a refreshing post-meal mouth freshener popular in Indian culture due to its rich antioxidants. The machine allows users to dispense saunf by hovering their hand over with a photoresistor sensor after paying $0.25 via a mobile app. Users are also shown an OLED display with messages and instructions, and an indicative RGB LED.

Problem

Saunf is a traditional Indian mouth freshener that is given out in a bowl at every authentic Indian restaurant. I love Saunf and I often munch on these fennel seeds in my free time by buying it from the grocery store. However this process gets messy since the packages sold are always in a plastic bag that you cut open once with lots of room for spills, and at restaurants people are just spooning saunf into their hands which can get unsanitary.

Solution

I decided to make a saunf vending machine type dispenser so that I can easily access saunf in an efficient and maintained way. This vending machine concept can also decentralize the consumption of saunf to people who have never tried it or those who want it whenever they want whenever they are out, just like how people use vending machines.

Hardware Components

- Particle Argon
- OLED display
- DHT20 temperature sensor
- Photoresistor
- RGB LED
- Servo

Software Libraries

- Blynk: For IoT capabilities and mobile app integration
    - Blynk documentation: https://docs.blynk.io/en/
- DHT20: For interfacing with the temperature sensor
- SparkFunMicroOLED: For controlling the OLED display

Important Variables

- payVal: Stores on/off state of the "Pay" button in Blynk app
- rgbGreen: Tracks whether payment is completed and dispenser is activated
- payment_revenue: Total revenue from payments made

Code Structure

SECTION 1: INITIALIZING LIBRARIES AND GLOBAL VARIABLES

- Where all the libraries are included for DHT20 sensor, OLED, Photoresistor
- Blynk configuration defined
- Global variables are defined:
    - Servo servo; Servo's object and pin is defined. Library is already included in Particle
    - RGB data pins
    - payment_revenue goes through initial state
    - The photoresistor threshold for the dark light in my environment - this varies per person's environment.
    - payVal is initialized which is the variable that stores the state of the virtual button on Blynk

SECTION 2: function definitions

- The change RGB function is a clean and easy way to analog change the RGB color output whenever needed later in the code
- My Handler handles the integration response of the initial state

SECTION 3: setup function()

- Sets up pinModes of the rgb pins
- Begins the DHT20 sensor, OLED, authorizes Blynk app, Servo
- Subscribes to Initial State

SECTION 4: BLYNK_WRITE function

- The Blynk function that allows the app to control the Argon
- payVal is set to the state of the pay button based on user interaction on the mobile interface
- When the button is clicked, the payVal value is 1. Whenever this happens,
    - Particle.publish sends the payment data whenever the pay button is clicked. 25 cents is added each time to the revenue total displayed and published on initial state
    - After payment, light turns green, the rgbGreen state is TRUE whenever the payment is completed, aka payVal = 1 (button is clicked)
    - OLED displays an "activated" message, and also messages based on the temperature of the environment at that given time
- If button is not clicked, or payVal = 0, then the LED color should display red meaning the rgbGreen value is FALSE, once again tracking the payVal state with the rgbGreen state

SECTION 5: The loop function is the core logic that loops on and on. This is where the actual Servo dispensing actions take place.

- Blynk is run continuously
- DHT sensor status is being read and stored in a variable for monitoring and debugging purposes, but it is not necessary. Usually it would be used in a conditional statement to check if it is functioning properly
- Temp stores the DHT temperature in fahrenheit
    - Optional serial debugging for if you want to see the exact temperature
- The photoresistor's light reading from 0-4095 is stored in the variable lightReading, which can be serial monitored. This is useful for determining proper thresholds based on the environment the vending machine dispenser is in. This is because the light values are relative so the light value when a user hovers their hand over the sensor varies in order for it to work in every environment

SECTION 6: Main logic of loop function

- Inside the loop function, there are a series of conditional statements that determines when the dispenser is active when it is locked, based on payments and light detection

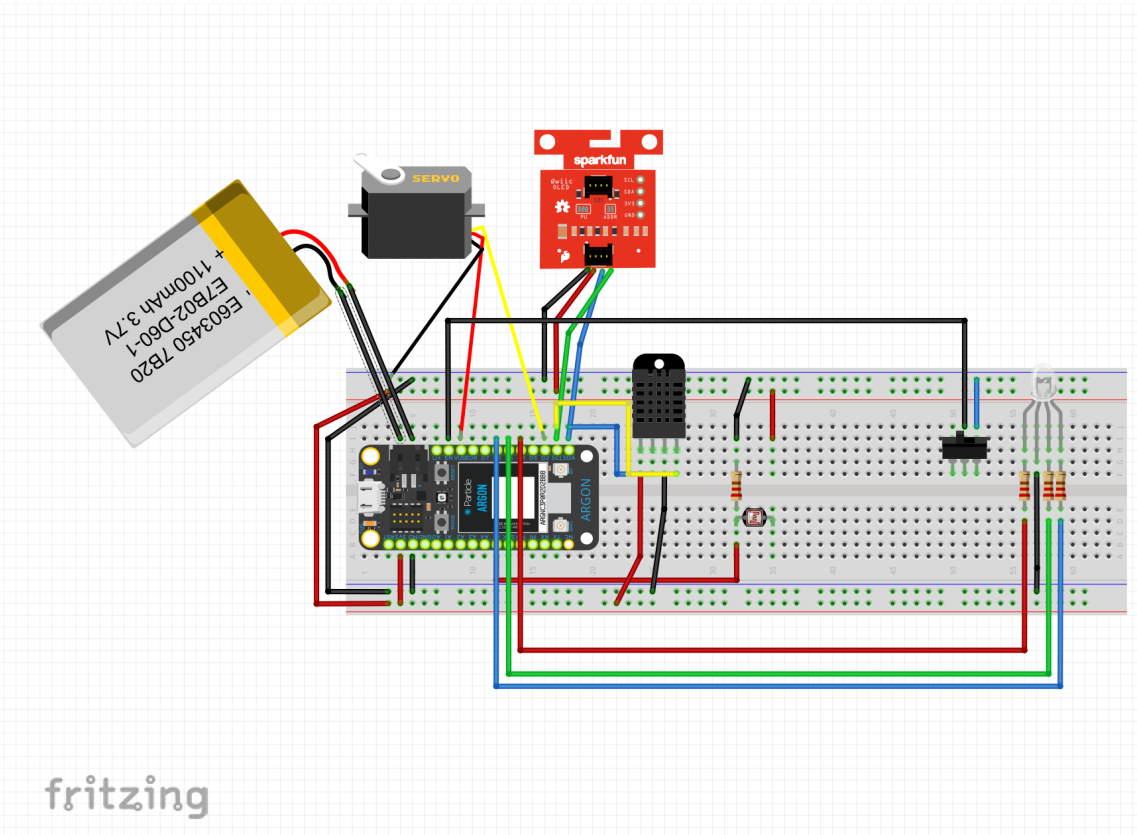
1. If the light reading detects "dark", meaning the hand is over the sensor, AND the pay button is clicked (implying the LED color aka rgbGreen is TRUE based on the Blynk function), then the servo will move to dispense the saunf
2. Else if the pay button was clicked and the sensor does not detect darkness, then this is considered an idle stage where the servo does not move yet but the rgb led indicates that the user can dispense by turning green and changing the state of rgbGreen to TRUE again
3. This Else if acts like a lock, so when the sensor detects darkness, aka a hand, but the payment has not gone through (payVal =0 AND rgbGreen = FALSE, the servo will not move and the OLED screen will tell the user to pay first in order to dispense
4. Else basically means payVal =0, therefore resets the rgbGreen state to FALSE, communicating to the program that there is no payment so any other functions can proceed as needed

End of code!

Future Improvements

- Send low balance alerts when saunf needs refilling
- Allow dynamic control over the light threshold to account for changes in light intensity throughout the day
- Add music/sound effects when dispensing is done, or voice narration of the OLED messages

Fritzing



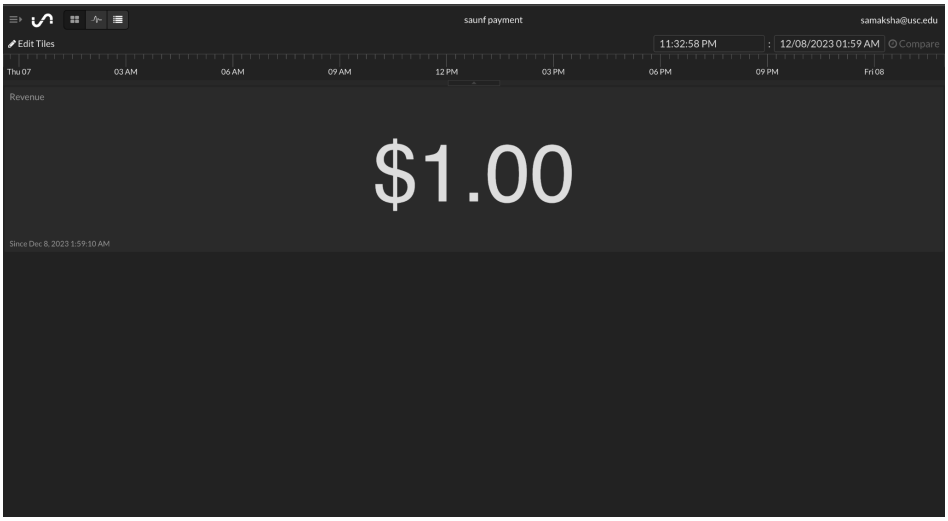
Component	Purpose
Photoresistor	To detect hand motion based on relative light changes
DHT20 Temperature and Humidity Sensor	To output kind messages about the weather specific to the temperature in the user's environment
RGB LED	To indicate whether the vending machine is locked or active for dispensing
Micro OLED Display	To output messages for communication like instructions and kind messages
Servo	To move the bowl of saunf in a position to dispense through the funnel
LiPo Battery	To allow the microprocessor to function wirelessly

Power Switch	To turn off the circuit environment completely to preserve battery when not in use

← Blynk Interface

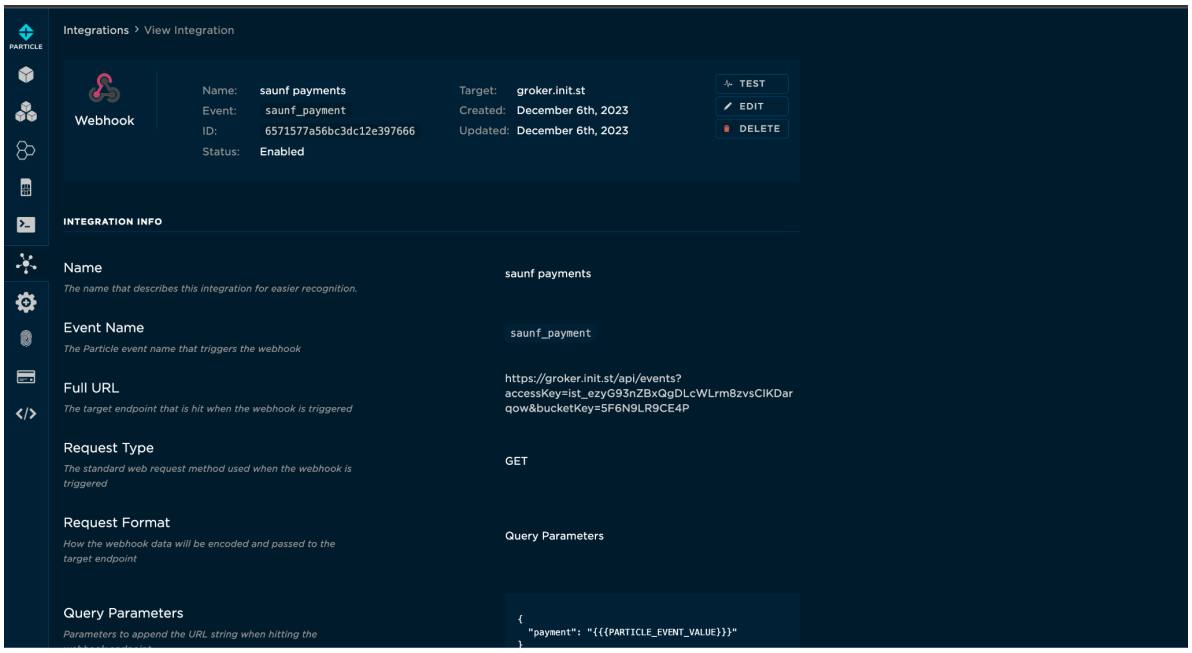
What the Blynk user interface looks like. You would click the latch button that says “Pay $0.25” to pay and be able to dispense food.

Initial State



Event Name	Details
saunf_payment	Detects when user has “payed” on the Blynk mobile interface - when the pay button is “on” (true/1). $0.25 is added to this revenue total everytime someone “pays”.

Particle Webhook Integration for Initial State:



PROJECT DEMO

PHOTOS