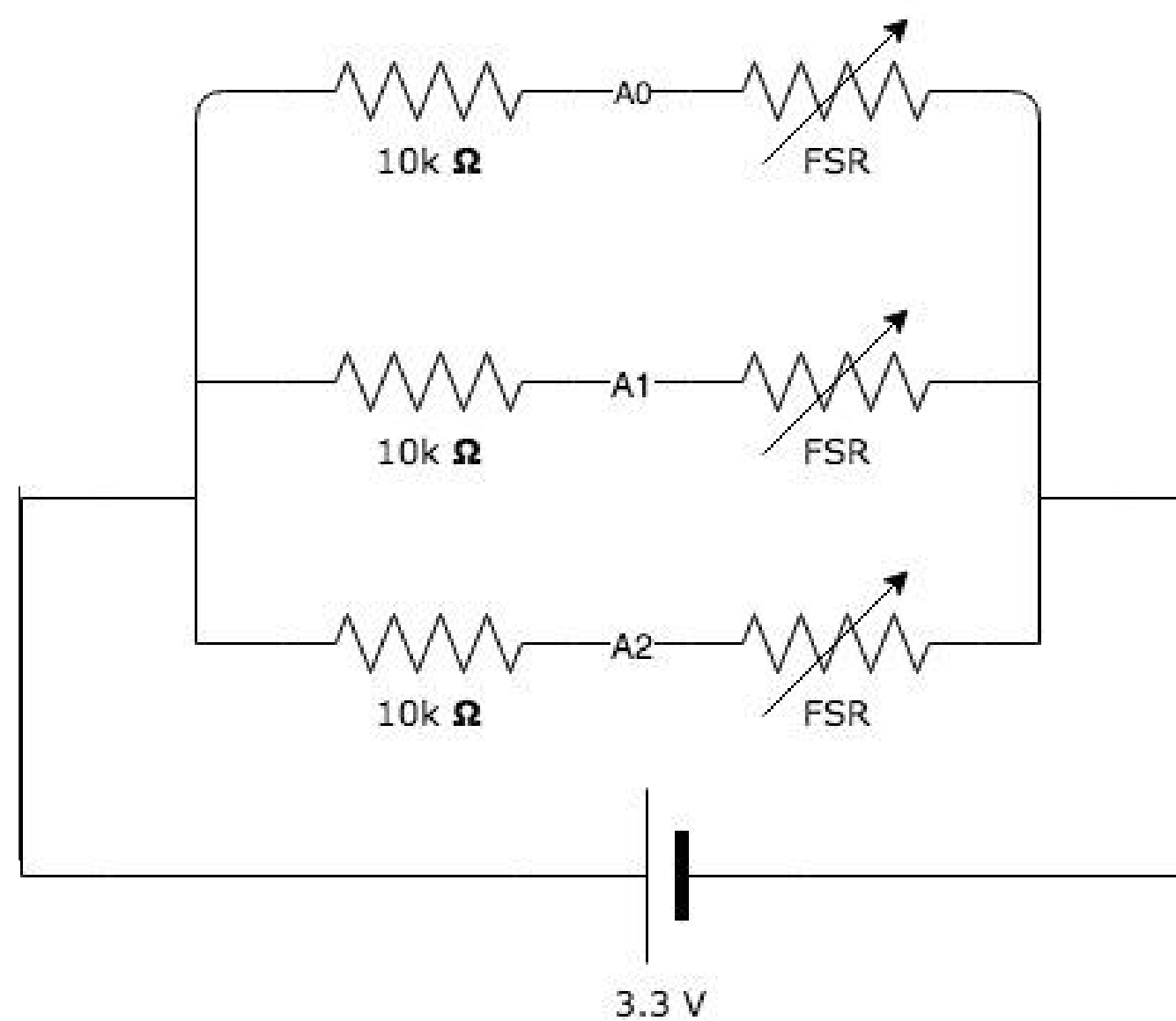


Recipe Ready

Force Sensitive Resistors

Force sensitive resistors (FSR) are an essential part to our IOT devices, as they are the source of input for our system. These resistors will sense any weight on the top of their sensitive padding, and would change its resistance based on how much pressure was applied, ranging from infinite resistance when no pressure was applied, to 200 ohms when max pressure is applied.

In order to read from these FSRs, we incorporated an Arduino Uno into our IOT. We designed our circuit so that the three FSRs are connected in parallel; they share the 3.3 volt power source from the Arduino, while connected to the same ground pin. Each FSR also outputs their reading to a particular analog read pin on the Arduino.



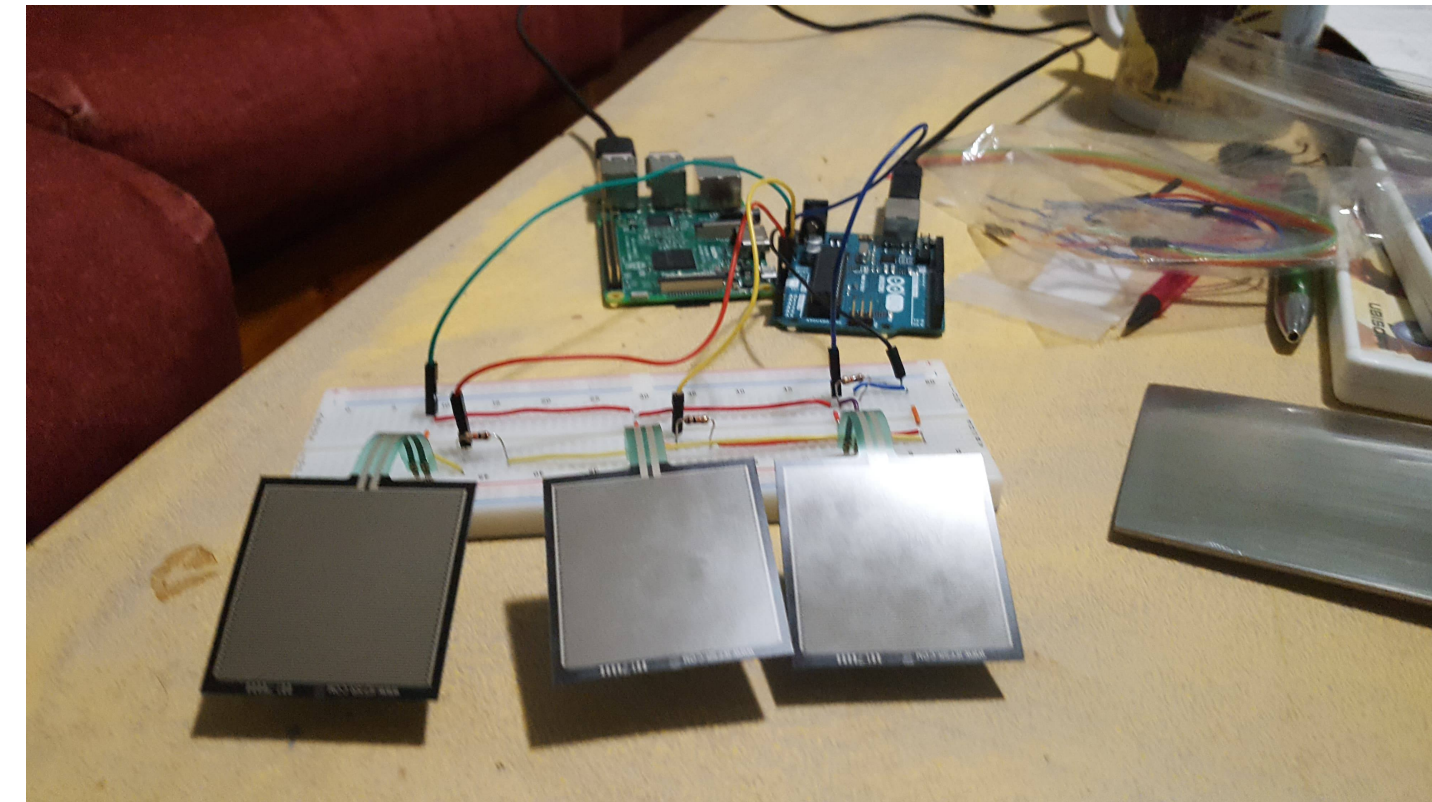
Arduino Program

The **Arduino program** identifies the presence of each ingredient by reading pressure input from each FSR, detects changes in ingredient availability, and relays the availability to the Raspberry Pi if changed. This is accomplished by sampling the pressure of each FSR at a specified rate. A pressure of 0 from the FSR signifies a lack of the specified ingredient, while any pressure represents the presence of the ingredient. If a change in the availability is detected, a serial write is performed to inform the Raspberry Pi of the updated list of available ingredients. No serial write is executed if there is no change in ingredient availability.

Raspberry Pi

The **Raspberry Pi** serves as a bridge between the Arduino and the webserver. It collects data coming in over the serial wire and then makes HTTP GET requests to the webserver, supplying it with the user ID of the device as well as whether or not each sensor is activated or not. In order to prevent the webserver from being overloaded with requests, or from getting improper requests, it forwards data if and only if the new data present on the wire from the Arduino differs from the last data to come in over the wire.

Project



Data Forwarding Programme

The Python **Programme** makes use of two libraries. The first is PySerial, which abstracts away the task of reconstructing the bits coming over wire into a human-presentable form. The second is the requests library which handles the low-level mechanisms of sending an HTTP request. Instead, all that needs to be supplied is the type, parameters and destination of the request... as well as a timeout so that one does not wait too long for a response.

The last bit of technology that enables the programme to run as intended is the Cron system utility. Since the Pi would be deployed in an environment where it would seldom interact with the user, it is imperative that the programme runs every time the system boots. By adding the appropriate line to a Crontab, the utility knows to begin execution of the Python script as soon as the system reboots and the cron daemon comes online.

The Cloud

We are using a **Cloud** based server farm run by a company called Digital Ocean. This allows us to continuously update our code base so that our IoT devices will send information to. The server runs a very minimal configuration that can be scaled horizontally to accommodate growth. We are also able to continuously integrate new features!

This server runs Ubuntu with packages for Node.js and MongoDB pre-configured. Node.js is a server-side JavaScript variant that runs on Google's V8 JavaScript engine. MongoDB is a NoSQL database that handles any data type. Node.js has a highly active open-source community that gives us access to "node modules" (libraries) to perform any number of tasks such as provide API endpoints with "express" or connect to MongoDB through "mongoose". This combination of technology runs our server and handles all data from the sensors through HTTP POST requests. This server also holds the front-end code to display the web app to our users. This is written in HTML and JavaScript, with styling provided by a W3schools CSS template. This is accessed through any web browser.

All of this combined allows us to run a highly scalable and feature-driven product to provide ongoing value to our users.



Team 7

