VENDINGHEALTH

Nadina Ștreangă, George Oprea, Politehnica University of Timisoara May, 2018

1 Repository

The code and project documentation are contained under the following git repository:
`https://github.com/georgeoprea/VendingHealth`

2 User requirements

1. The system must vend snacks to its user based on a calorie based credit system.
2. The device must be able to read the NFC cards provided with the vending machine.
3. The system must allow users to view their credit count in a desktop application.
4. The system must allow users to see the available snacks and their credit cost in a desktop application.
5. The system should allow users to sign up via a web application.
6. The system should allow users to restock or change the snacks in the machine via a web application.
7. The system should be open for client application extensions (mobile application)
8. The system should be open for extension HW extension (adding an NFC writing module so that more than 2 NFC cards could be used with the application)

3 System overview

The overview of the system is depicted in Figure 1.

The base modules (Push button subsystem, RFID Reader subsystem, Ultrasonic Sensors subsystem, Motors subsystem) are the actuators and the sensors of the system. The RFID Reader is able to read a user's card. The push buttons identify the products and receive input from the user in order to select one of the products. The motors start to spin based on the information received from the RPi interaction module. The ultrasonic sensors detect a product has fallen and sends a response to the motors to stop.

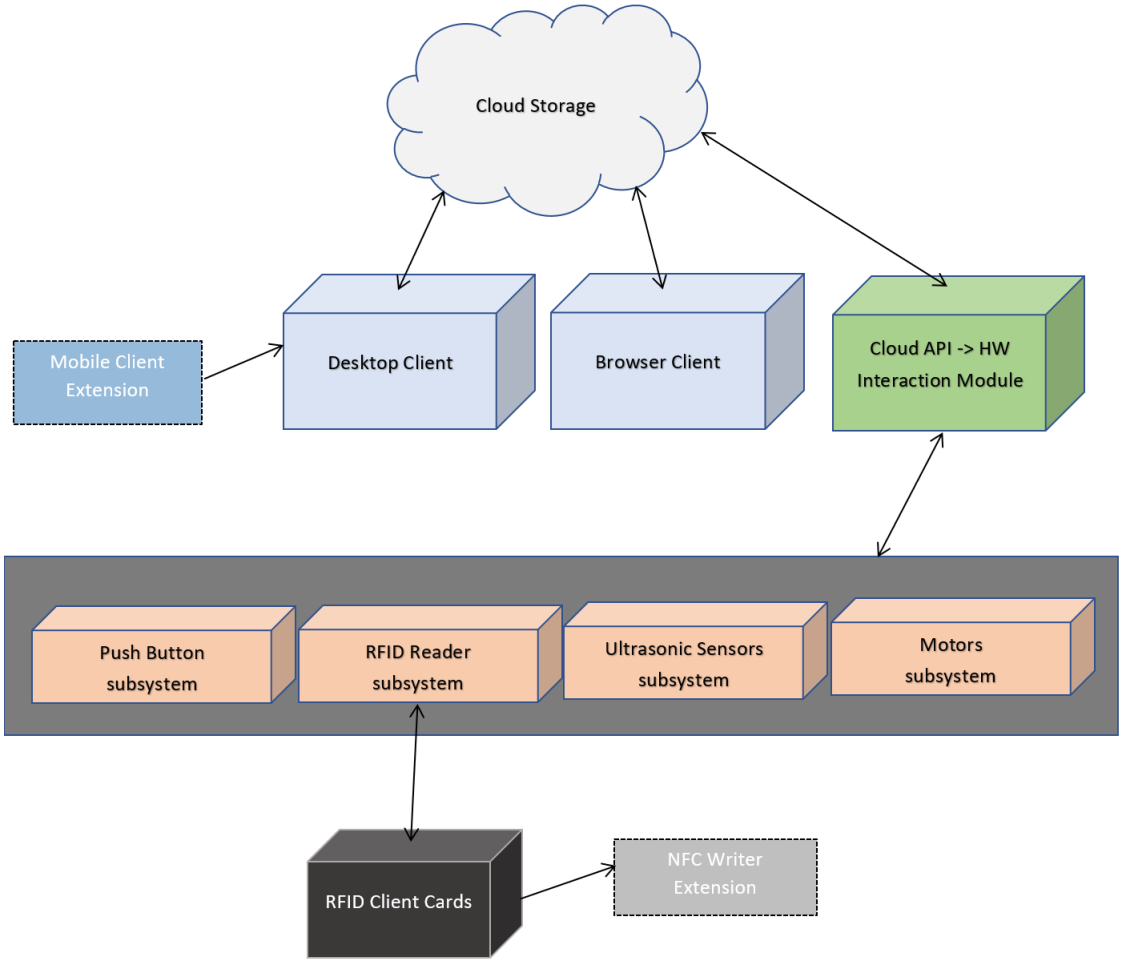

Figure 1: System overview diagram

The HW Interaction Module is a Raspberry Pi that runs a python module. It interacts with the Arduino through a serial interface connection. It also communicates with the Cloud Storage through HTTP requests.

The Cloud Storage Subsystem is a Firebase module that stores user data (credit count, username, password, email), product data (stock, kcal, name, image) and the VendingHealth. client application. It also allows updating users and products.

The Dektop Client provides a UI that allows the user to view his/her credit count and view the products in the vending machine alongside the credits they would conusme.

The Browser Client also provides a UI that connects to the Cloud Storage and allows for an admin to change products in the vending machine and update the cloud storage with the change. It allows a new user to sign up on an RFID card and also provides the means to download the client application.

4 Circuit design

The hardware view of the system and the wiring of the components are depicted in Figure 2.

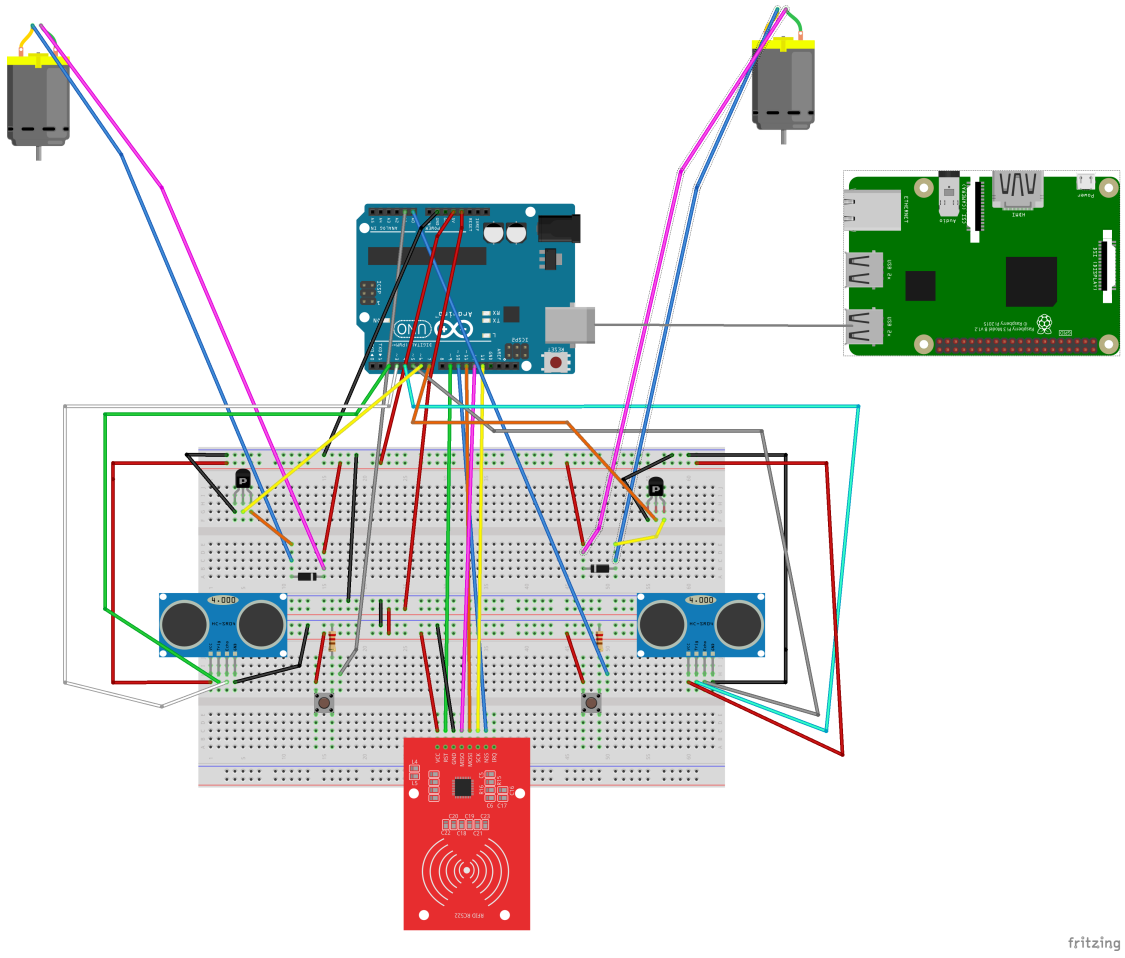

Figure 2: Circuit schematic

Raspberry Pi 3 provides a great interface for connecting to the cloud storage and also to connect to the Arduino. It uses one of its USB ports to connect to the Arduino USB port and receive bits of information. It communicates with the cloud through the internet using HTTP GET/PATCH requests.

Arduino Uno provides power to its components using the 3.3V and 5V outputs. It also connects all sensors and actuators in the same environment and uses both its analog and digital pins to send impulses to all components.

The two HC-SR04 Ultrasonic Sensors use 2x2 of the Arduino's digital outputs for the Trig and Echo pins. The Trig pin triggers a sonic signal and then the Echo pin will transmit the received signal which represents the number of miliseconds the sound wave that was outputted traveled. The sensors are powered at 5V.

RFID MFRC522 is paired with two dedicated RFID cards from which it reads the tag numbers. The RFID reader is powered at 3.3V. It uses the SPI protocol to communicate with the

Arduino. Ports 9 to 13 from Arduino are linked to the MISO, MOSI, SCK, IRQ, RST, NSS pins of the RFID Reader.

The two DC motors are powered at 5 V. They are connected to a diode and a PN2222 transistor. The diode assures that no reverse current will flow from the motor to the Arduino. The transistor allows gaining control of the motor rotation. The transistors are connected to pins 6 and 7, respectively.

The two push buttons are connected to the A0 and A1 pins of the Arduino and are powered at 3.3V.

5 Software design

Provide a walktrough of the most important components/modules/entities or concepts you've implemented in the software section.

The software components and data flow directions are depicted in Figure 3. Each of these will be presented in the following subsections.

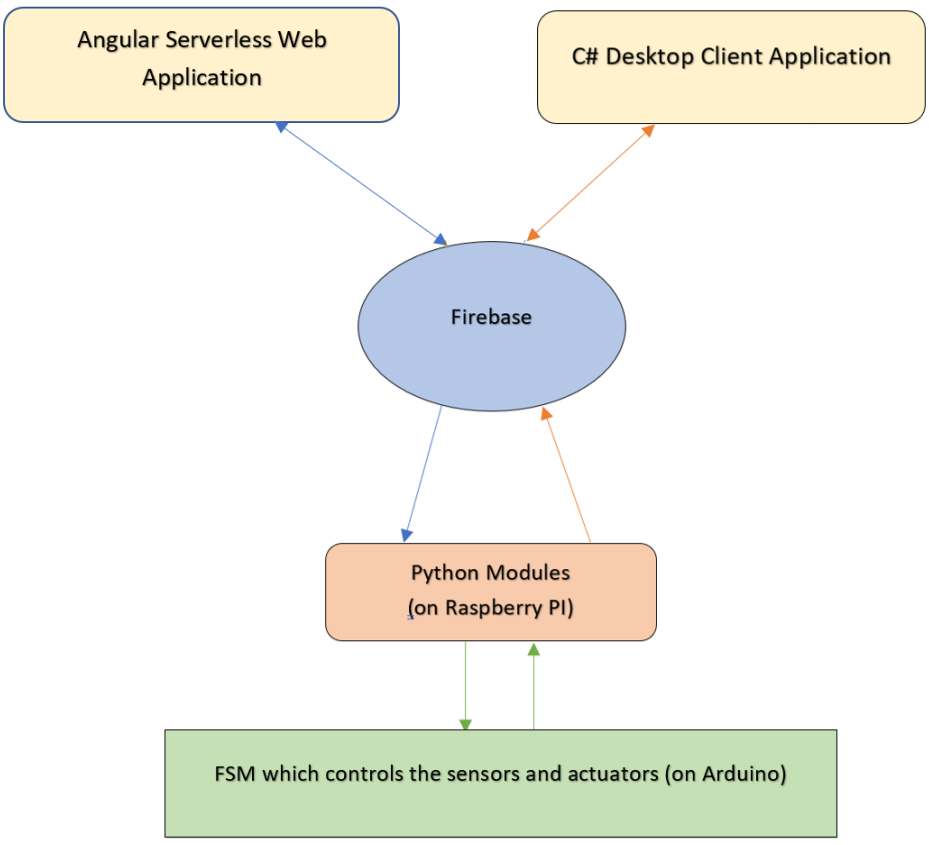

Figure 3: Software entities involved

5.1 Arduino Modules

5.2 Python Modules

You should not concentrate on providing line level descriptions, but rather class/script/module level explanations. Short code comments are strongly advised. Third party libraries should also have a brief description and a reference link.

write_firebase_sensor.py: it retrieves the temperature and humidity tuple from DHT-22 sensor over the one-wire interface and then pushes it to Firebase every 1 second.

Adafruit_DHT22 library: it provides a quick implementation of one-wire interface communication with the sensor.

5.3 Firebase

Describe the purpose of the service and present the specifics you are using.

Firebase is a PaaS (Platform as a Service) which means it offers developers to a quick list of functionalities supported by a traditional backend. Realtime Database simplifies storing and synchronising data between different devices in realtime using a noSQL database.

The following code section shows the initialization of the Firebase project.

```
# root of the project
FIREBASE_ROOT = 'https://vendinghealth-alpha.firebaseio.com'
# init Firebase Database instance
firebase = firebase.FirebaseApplication(FIREBASE_ROOT, None)
```

5.4 Desktop Client Application

Explain the responsibility of some key entities contained in your Android project.

5.5 Browser Application

Give a brief description of the main parts contained by the JavaScript module.

6 Results and further work

The current version of the project supports the following functionalities:

- client implementations for retrieving product and user information stored in Firebase Database (Desktop and Web)

- storing new products in the Firebase Database
- allowing an admin to modify products in the Firebase Database
- allowing users to create an account for the application and register a certain card to it
- allowing users to download the application from the browser application

The following list of extensions and improvements was identified to be supported in the future:

- include a mobile version of the desktop application
- extend the desktop application to Unix users
- allow extending the number of cards that can be used with the machine
- improve application security (encrypt sensitive user information)
- allow more vending machines in the Firebase database by creating a new node for each one

7 References

1. Draw IO [last seen: May 2018], https://www.draw.io/
2. Fritzing [last seen: May 2018], http://fritzing.org/
3. Firebase Database [last seen: May 2018], https://firebase.google.com/docs/database/
4. RFID library [last seen: May 2018], https://github.com/miguelbalboa/rfid