*Microprocessor Systems*

**Pets Smart Feeding device**

Amelia Munteanu, Maria Teodor

**Content:**

1. User Requirements

1. System overview
2. Hardware design
3. Software design
4. Python modules
5. Repository
6. Results and further work
7. References
8. **User requirements**

SmartFeeding device must succeed in dispensing food at user request or automated, based on a schedule and a distance range. The system must ensure it provides the proper amount of food every time and it should also provide permanently access.

The access to the system should be provided via an easy-to-use Web interface. It should be open for extension, eg. adding an ultrasonic sensor.

1. **System Overview**

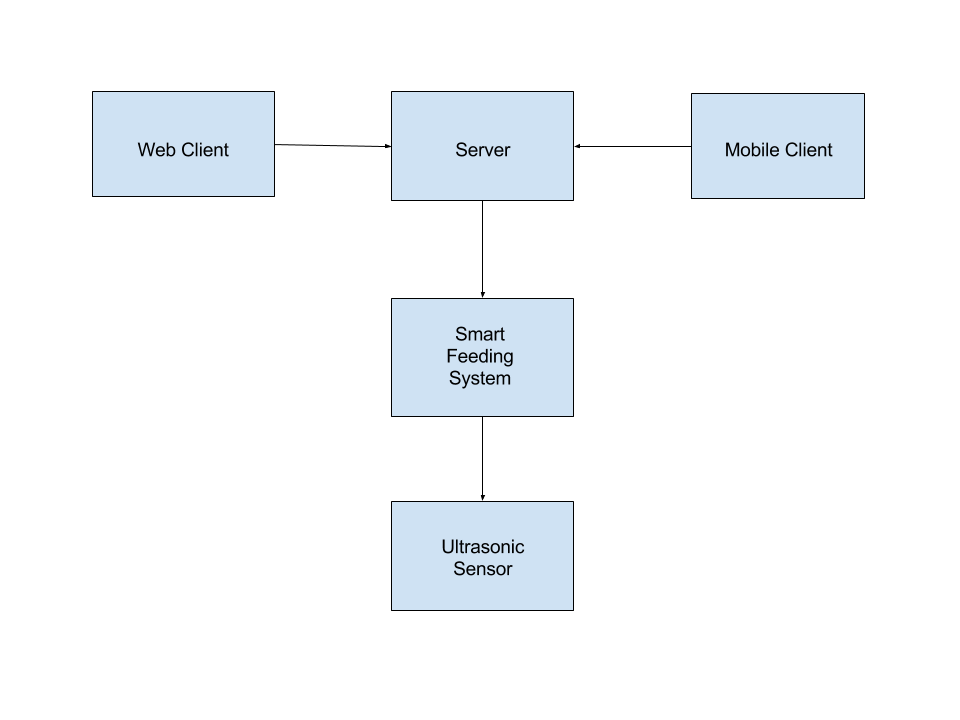


Figure 1: System Overview

The server created using Flask and RESTful web service has the purpose of controlling the Smart Feeding System by rotating the stepper motor attached to a portioned food container (**Figure 2**) such that it performs 1/8 from a full rotation whenever it receives a request from the client.

Web client can send a request to the server by accessing a page within the Web browser.

Mobile client can also send a request to the server by accessing a page within the Web browser running on a mobile device.

The ultrasonic sensor is an extra feature of the Smart Feeding Device, which enables the option of automatic feeding, the mechanism of dropping food being acted whenever the sensor detects an obstacle within a previously established distance range and after a certain period of time has passed since the previous dropping.

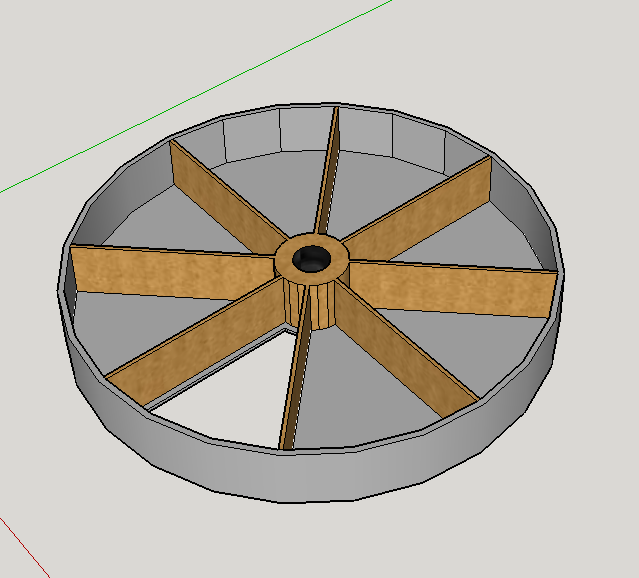


Figure 2: Portioned food container

1. **Hardware design**

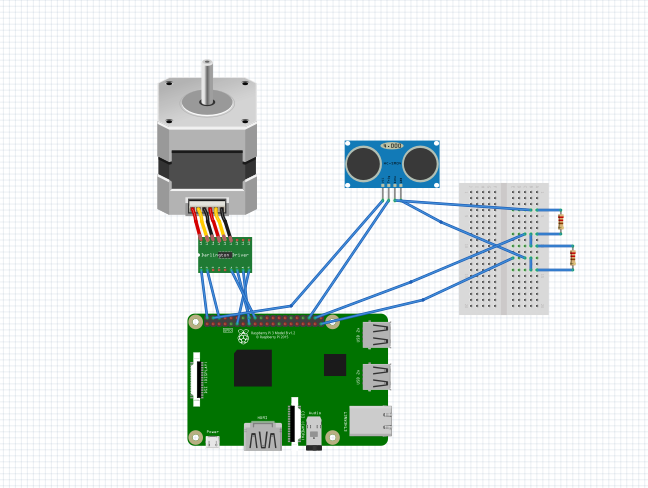


Figure 3: Hardware Diagram

We use **Raspberry Pi 3** as it suits best for what we have thought about our SmartFeeding prototype, it is the most versatile and has a widest range of capabilities. It was chosen especially for its Wi-Fi ability as we host a server and communication over the Internet is needed.

As it can be observed in **Figure 3** a **28BYJ-48 Unipolar Stepper Motor** is used together with a **ULN2003 driver board**. The motor connects to the controller board with a pre-supplied connector. There are 6 pins that have to be connected to the Pi header (P1):

* Inp1 (P1-11)
* Inp2 (P1-15)
* Inp3 (P1-16)
* Inp4 (P1-18)
* 5V (P1-02)
* GN D(P1-06)

The input voltage for the motor is 5V which we can power from the Raspberry Pi. The stepper motor is used to rotate the portioned container in order to drop a precise amount of food. We do this by providing a sequence of ‘high’ and ‘low’ levels to each of the 4 inputs using half stepping mode.

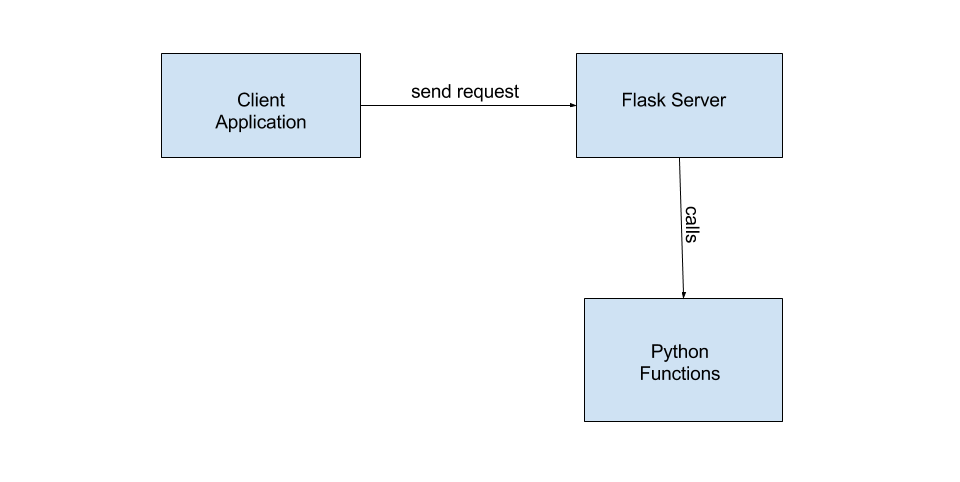
Half Stepping is a combination of single and dual phase stepping, alternating between one coil and two coils being turned on. In half stepping one cycle takes 8 steps -despite full stepping which takes 4 steps- thus doubling the resolution of the step motor resulting in smoother rotations. Our container is divided in 8 partitions but only one partition is rotated at a time so in order to achieve that a range of 64 cycles are used.

1 revolution= 8 cycles; gear ratio= 1/64 => 1 revolution = 512 cycles

1/8(meaning 1 partition) rotation => 512/8 = 64 cycles.

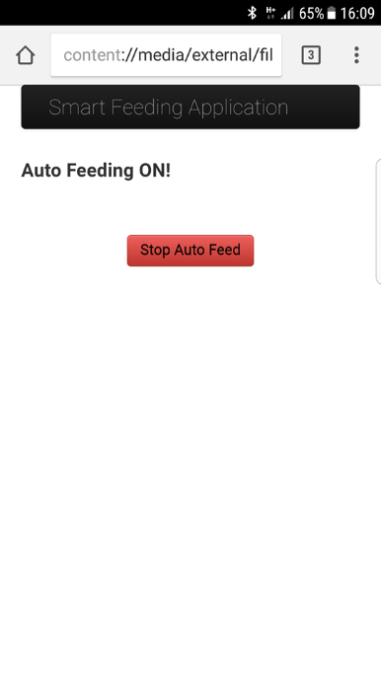
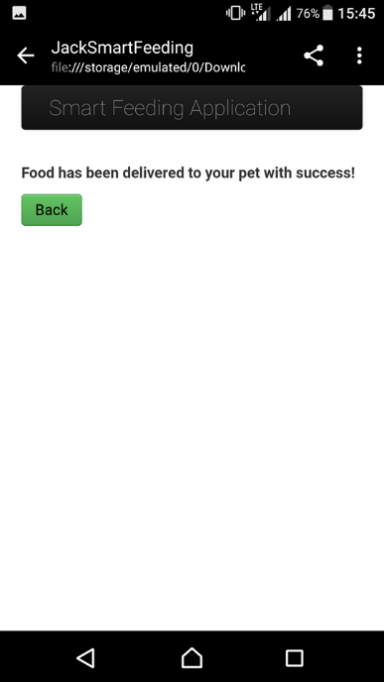
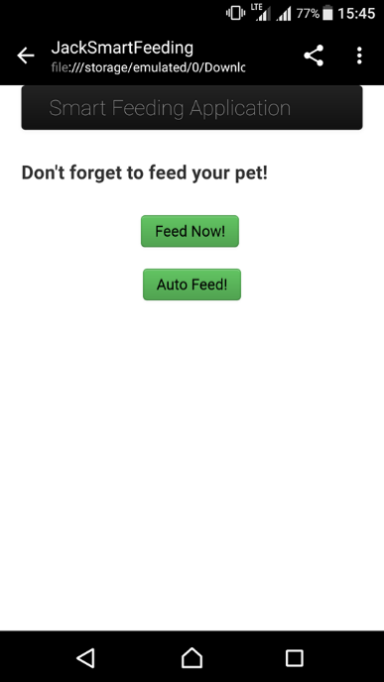
In our circuit a HC-SR04 Ultrasonic sensor is used. Connected to the Raspberry Pi through the breadboard, it uses a 5V (P1-04) Vcc, GND (P1-09), Trig (P1-22), Echo (P1-24). This sensor is used in automated feed mode on, to feed the pet based on a distance range from the dispenser and also based on a schedule.

1. **Software design**

 The client application is represented by three html pages. The first one, index.html, is the page being displayed in the browser whenever a client navigates to the address of the Raspberry PI board, followed by the port on which our server is listening. It has two buttons, corresponding to the two types of request a client can send to the server: deliver food to his pet with one simple click (“Feed Now!” button), enable automatic food delivery whenever the pet gets close to the sensor (“Auto Feed!” button).

feedNow.html is the page where the client is redirected after “Feed Now!” button was pressed. It contains a “Back” button, which sends the client back to the main page.

Screen captures with the 3 html pages can be seen below:



After pressing the “Auto Feed!” button, the client will be redirected to autoFeed.html, a page which contains a “Stop” button, which will end the process started by the previous action.

When “Feed Now!” button is pressed, the python function feed() is called, because @app.route decorator was used in order to match the redirect URL with our function.

The function feed() simply starts the stepper motor, which will make 1/8 from a full rotation, thus the container will drop one portion of food.

When “Auto Feed!” button is pressed, the python function autoFeed() is called in a similar manner, but this time it will run in another process, waiting for the sensor to detect the presence of the dog near the device. If the dog is placed in front of the device, at least 20 cm near to it, the function feed() will be called, dropping another portion of food. This will only happen if there already passed at least 4 hours since the previous time when the dog was fed.

For creating the server we used Flask, a micro-framework which provides tools, libraries and technologies for building web-based applications.

1. **Repository**

The project history, schematics, diagrams and codebase are contained under the following git repository:

**https://github.com/MariaTeodor18/Travian**

1. **Results and further work**

The submitted version of the project offers the following functionalities:

* Dispensing food to your pet remotely, with a single button click
* Enabling auto feeding, a feature which will feed your dog whenever it is near the device (between two food dispenses, a time period of at least 4 hours should pass)

For the next iteration, the following improvements are planned:

* Improving web interface for the client
* Making our local server accessible over the internet

1. **References**
2. Project Repository, Web:
3. Google Drawings, Web: https://docs.google.com/drawings/create/
4. Fritzing, Web: http://fritzing.org/
5. Flask, Web: http://flask.pocoo.org/
6. 28BYJ-48 Stepper Motor Datasheet, Web: http://robocraft.ru/files/datasheet/28BYJ-48.pdf
7. HC-SR04 Ultrasonic Sensor Datasheet, Web: http://www.micropik.com/PDF/HCSR04.pdf