PET CARE AMI 2016

Designed by:

Federico Parodi
Simone Impagnatiello
Francesco Gramazio
Filippo Caldera

PET CARE: the system is designed to satisfy our pets' primary needs

- Feed need:

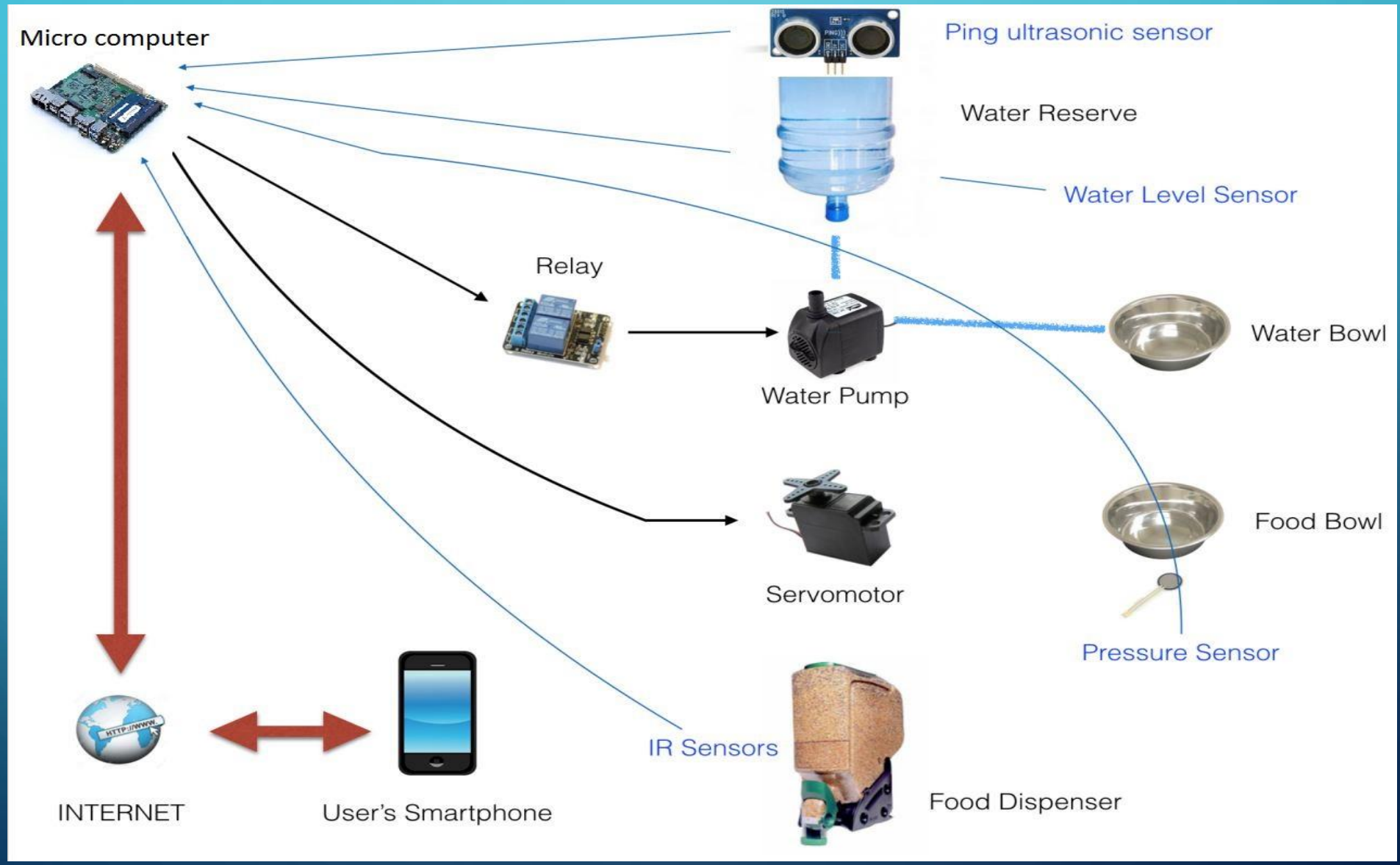
  It provides food and water, trought dispensers

- Physical need:

  the pet can play with an automatic toy which provides fun and physical activity

Our system is programmable by the user who can schedule daily events that allows the system to look after to your pet

HARDWARE ARCHITECTURE



WATER MANAGMENT

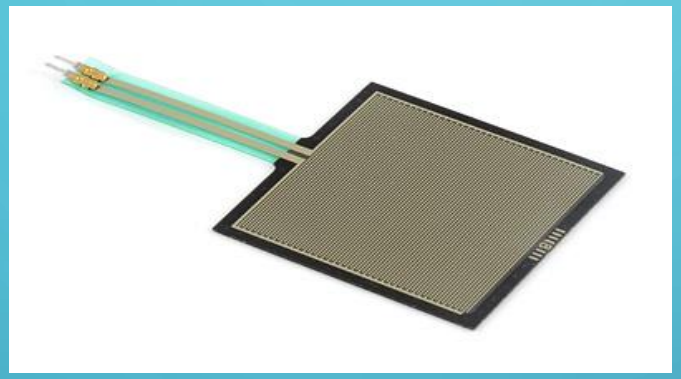
Water level control is programmed to keep the water bowl full anytime.

To do so, we exploited the water itself to create shortcircuit and detect if the water has reached full level

When water level goes down two wires are disconnected, allerting Raspberry to activate water pump

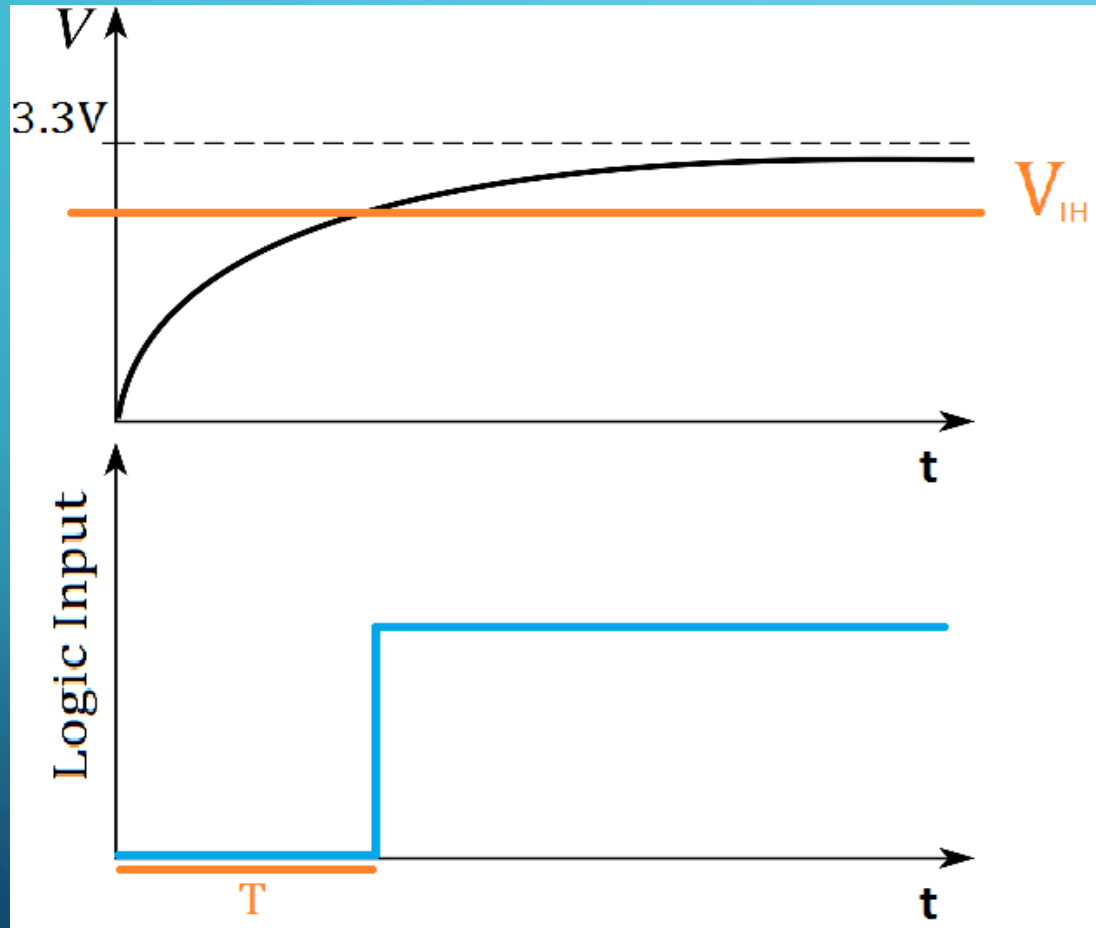
FOOD DEPLOYMENT SYSTEM

- Food deployment system uses two component:
- **Force Resistance Sensor:** measuring how many food has been released



- **Servo motor:** let drop food from food dispenser to the bowl

FORCE RESISTANCE SENSOR: HOW TO GET VALUE



Since : t=RC

where R is variable depending on weight and C is a constant, we can get the weight value measuring T

FSR AND SERVO WORKING TOGETHER

When is time to deploy food, FSR mesures the amount of food in the bowl, then if the level is too low allows Servo to add more.

At every single servo action FSR measures the new weigth and requestes more servo actions or stops it if food available in the bowl is enough already.

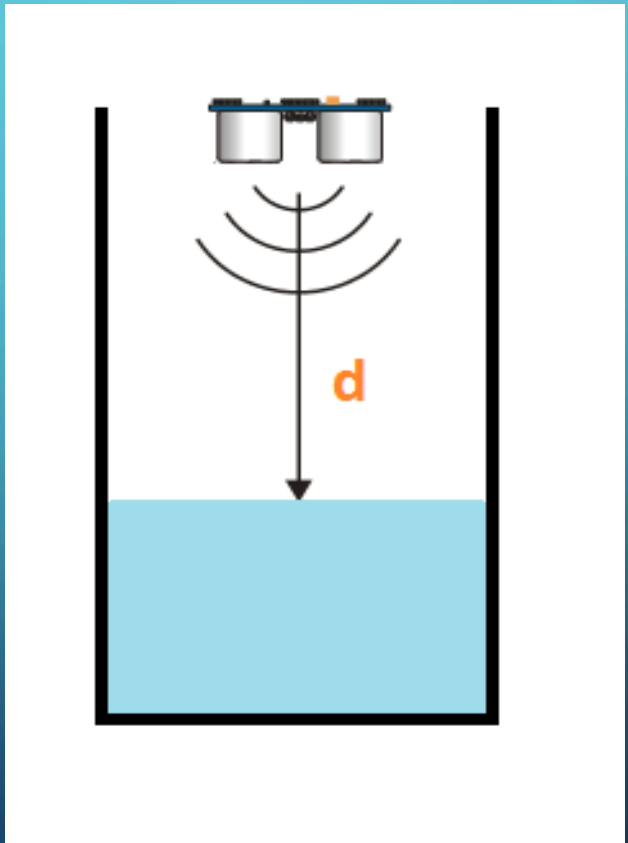
Software is also looking for issues:

- in case there are too many food deployment but there is no weight increasement then system stops servo.
- In case FSR is deteching no weight on it then servo is stopped .

DISPENSERS LEVELS CONTROLS

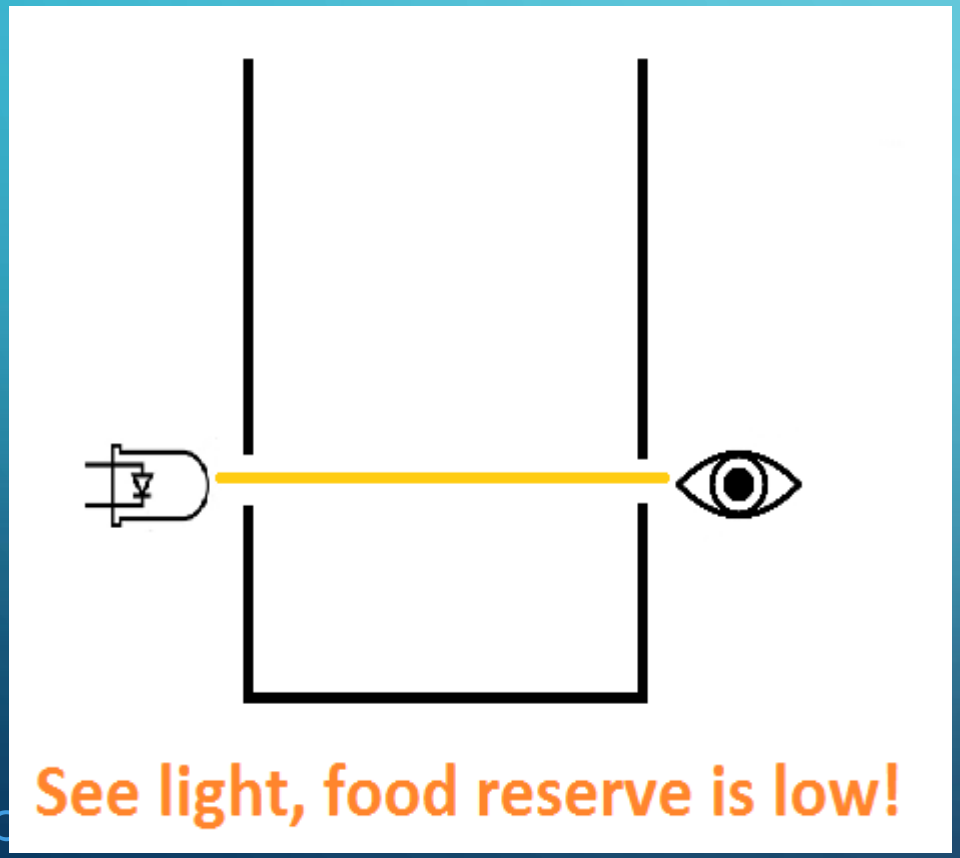
Two dispensers, two different level control ideas:

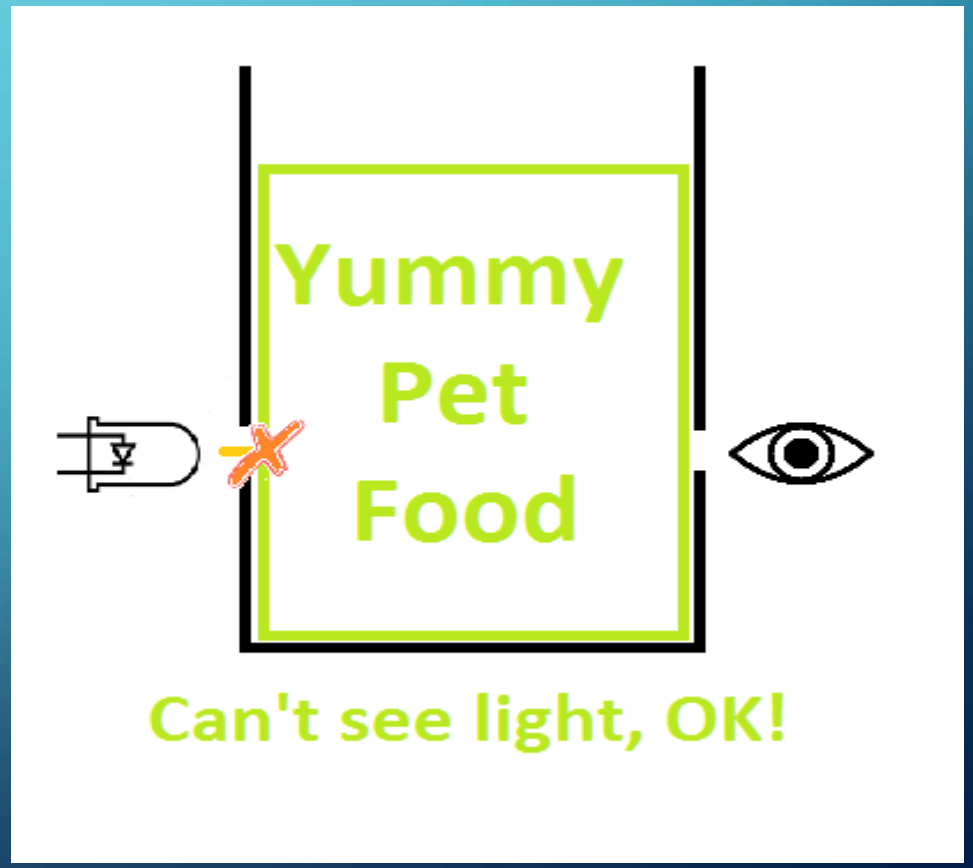
Water level is controlled using an ultrasonic sensor measuring the water level.



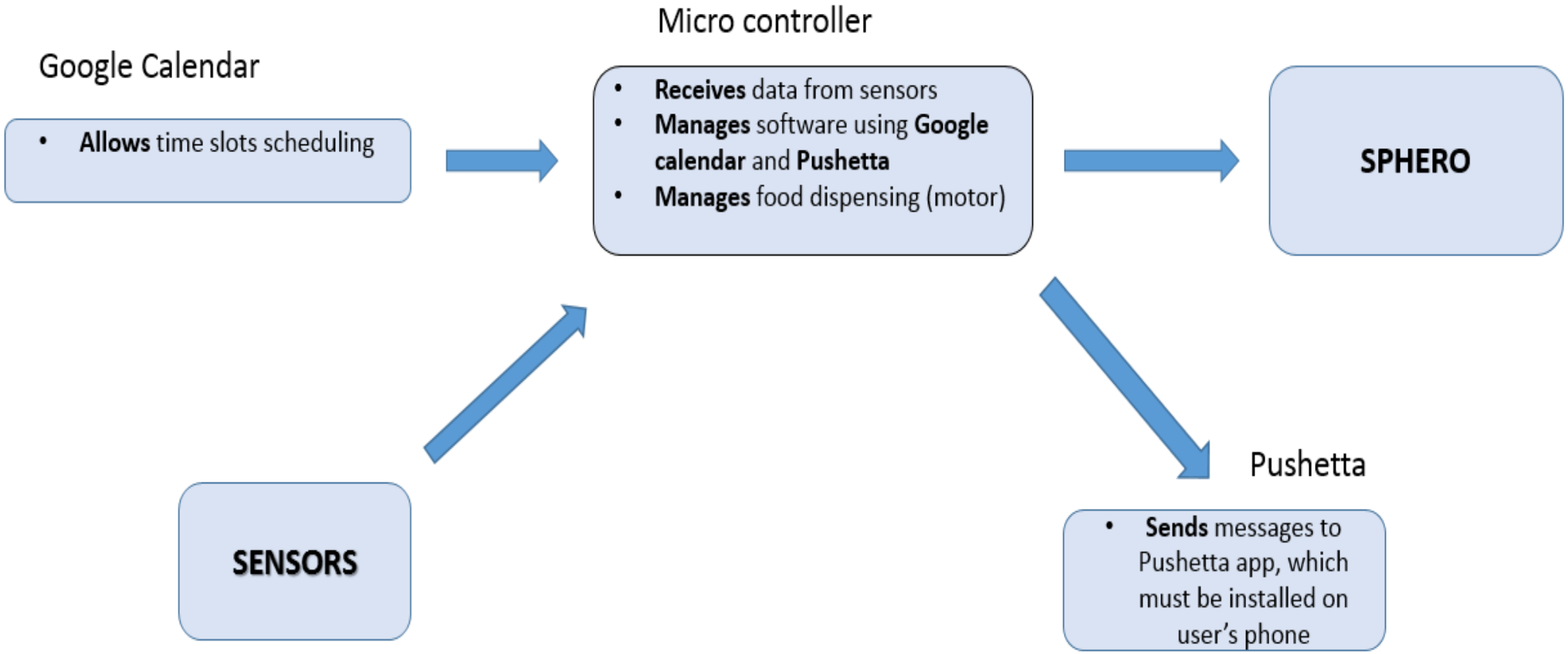
DISPENSERS LEVELS CONTROLS

Food level is controlled using a LED and a fotoresistor





SOFTWARE ARCHITECTURE



SOFTWARE ARCHITECTURE

- GOOGLE CALENDAR
- PUSHETTA
- SPHERO

The user must have a Google account and must create a dedicated calendar.

On the dedicated calendar the user sets a maximum three events per day.

Raspberry searches the events in your calendar between 6am and 11pm.

It also stores them into a FIFO list temporally ordered.

Google Calendar satisfies the AmI feature of **INTERACTING**, because it allows the user to schedule daily meal time slots.

ALGORITHM

The system works in polling mode, and compares the current time with the time of next events.

Every tasks in the routine requires a computational time

For example we set 12am as time to dispense food but...

...at 11.59:

Sphero (6min)

Refilling water (2min)

Message (1min)

→ Food is dispensed at 12.10

Pushetta is a instantaneous notification application.

It needs to create an account which must be created on Pushetta website, and then the user should create a private channel where the system sends info.

In order to connect the system with the user personal channel, you need to communicate to Raspberry your Pushetta channel's ID.

The notification implemented are about:

- the critical amount of food and water
- the activation of Sphero

Pushetta satisfies the AmI feature of **INTERACTING**, because it interacts with the user through notification.

Sphero is a programmable ball.

We use Sphero to allow the pet to play even when the owner is not at home.

It is activated when the pet stays in the bed for too long, and it moves randomly for some minutes.

ALGORITHM

Sphero detects collisions and let the ball to change direction and it stops after some minutes presetted by software designer.

Sphero requires a pairing routine between its own Bluetooth and rPi Bluetooth module

Sphero satisfies the AmI feature of **ACTING**, because the system itself activates automatically the ball, thanks to a pressure sensor placed under the pet's bed.

That's all Folks!