Rural Makers Project - Technical documentation & future steps

What we want to achieve

We need to send useful data to the API so that we can measure water consumption, and prevent leaks or excessive water consumption.
The electronic setup should be easy to mount, in a waterproof case, and also not too expensive.

For now, 3 possible ways to get water data:

1. Take a picture of the counter every day (or few days), send it to the cloud.
2. Measure the water flow directly, send data to the cloud.
3. Combination of both

The first phase of the project took place in Anceu, in August 2021. The goals were:

- Defining problems and possible solutions.
- Proof of Concept and prototype of one solution.

For the proof of concept we chose to implement solution 1, because it is more feasible to do in 1 week.

Hardware Prototypes / Possible Solutions

ESP32Cam

This is an ESP32 board with integrated camera and Wi-Fi, for less than 10€. The board would be mounted in front of the counter, in the counter box, so the camera faces the numbers of the counter.

What the code does:

- Connect to Wi-Fi
- Take photo
- Send photo to API
- Go to deep sleep mode

The deep sleep mode is useful to considerably reduce battery usage.

This is probably the cheapest solution to implement, and also the one which draws less power (i.e. longer battery life).

However, the quality of the picture may not be good enough. And the Wi-Fi reception is not super strong, but can be balanced by adding an external antenna directly on the ESP32Cam board.

The battery life may be good enough for a few months of use, but the ideal solution would be to have a solar panel.

A few useful docs:

- https://dronebotworkshop.com/esp32-cam-intro/
- https://randomnerdtutorials.com/esp32-cam-post-image-photo-server/
- https://randomnerdtutorials.com/esp32-cam-ov2640-camera-settings/

The code we wrote for the ESP32Cam can be found here:
https://github.com/agustinjch/rural-makers/tree/main/code/ESP32Cam

Raspberry Pi Zero W + Rpi Camera

The Raspberry Zero W can be bought for about 5-10€, same for the camera. The combination would cost around 15€.

Here's how it works (programming):

- Raspberry Pi boots and connects to Wi-Fi
- A Python script automatically starts
- The script takes a picture and sends it to the API
- For now, the script waits for a predetermined amount of time

This is also a cheap solution and the quality of images is pretty good.

However, the Raspberry Pi doesn't have a deep sleep mode, so the battery life would be just a few hours/days. This can be compensated with solar panels.

A solution for deep sleep mode would be to shutdown the Raspberry Pi whenever it has sent a photo, and wake it up (the Raspberry Pi can be woken up directly from the GPIOs) with another super cheap Arduino-like board, which has a deep sleep mode. The circuit would be a bit more complicated but feasible.

To setup the Raspberry Pi:

- Install Raspberry Pi OS
- Add the Python script in the home directory
- Create a systemd service to start script on boot (tuto here)

The code we wrote for the Raspberry Pi can be found here:
https://github.com/agustinjch/rural-makers/tree/main/code/rpi

Water flow sensor + ESP32

The ESP32 (without camera) is a small Arduino-like board with integrated Wi-Fi, for less than 10€.

A water flow sensor could be added just before the official counter, to get a different measure of water flow. The Arduino/ESP32 needs to constantly read data from the sensor to measure actual water usage, and send it to the API.

The main problem with this approach is that we measure relative (water flow) and compute absolute data (total water usage) from it. This involves an error that drifts over time and makes the measure completely unusable.
So, this approach would also require one of the camera setups previously detailed. The camera setup would help us get absolute water usage every x amount of time, and the water flow sensor would allow us to immediately detect a leak or excessive water usage over a very short period of time (minutes, hours).

A few useful docs:

- https://makersportal.com/blog/water-metering-with-the-wawico-usb-kit-and-raspberry-pi
- https://create.arduino.cc/projecthub/SAROSH_AHMAD/water-flow-rate-and-volume-measurement-using-arduino-in-2020-5377df
- https://www.seeedstudio.com/blog/2020/05/11/how-to-use-water-flow-sensor-with-arduino/

Server API and admin dashboard

Flask API

In order to be able to receive the data from the cameras (both raspberry pi and ESP32) a python server has to be set up. This server is a simple Flask server that will capture the images, upload them to Amazon S3 and send the data to another system that we'll talk about later.

The python flask server we wrote can be found in https://github.com/agustinjch/rural-makers/tree/main/code/flask-server . To be able to use it, we assume python and flask basic knowledge, but in general terms, it'd be install python3 environment with the requirements.txt is needed.

When the flask server is up and running, the next step of the system is a meteor.js admin site with API to get data from the python flask server.

Meteor.js API and Admin

The code we wrote for the meteor server can be found in https://github.com/agustinjch/rural-makers/tree/main/code/admin-server . To be able to use it,

we assume knowledge of meteor.js www.meteor.com , but in the repository, you can find a quick install guide.

The meteor.js admin will be the interface to use from the water community users, and they'll be able to "translate" the pictures into data, and the system itself will add that to their database and show it as graphs.

Future work & challenges

This first phase (1 week) was mainly used to define the problem, find possible solutions, and implement a proof of concept/early prototype.

The **second phase** is about building a prototype that can be installed and used in real life conditions for an extended period of time.

Here are a few points that need to be developed:

- water/humidity proof case for the hardware. Galicia is a very humid and rainy region. Humidity is very bad for hardware components and can wear them off very fast.
- Mounting the device on the water counter. You can find many different types of water counters. For each kind of counter, you'd need to design an easy pluggable support. 3D printing can be of great help for this.
- Battery. Each solution will need batteries to work. Some solutions are going to draw more current than others, but overall all solutions would require a setup with battery + solar panel. The best scenario is if we don't need to touch anything after setting up the counter.

Regarding the API and server side, the points to be developed are:

- Improve the admin features, selecting just needed data from the whole database
- Add alarms to send communications to the owners of a given water gauge, or to the members of the water association
- Develop a communications system. Email, whatsapp API, telegram API and SMS

The **third phase** would be the scaling of the prototype:

- Making it work for 10 counters, then 100.
- Defining an easy way to install the devices on each counter. Reducing the cost of components. Etc.
- Adding OCR or any AI system to automate the readings of the pictures
- Adding AI to create alarms automatically when measures are not matching previous ones.