SPROM

Smart Proximity for Obstacle Monitoring

By Adam Roque & Elisa Veloso

Presentation Summary

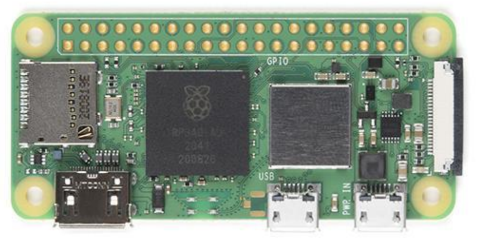
1) Introduction : (What is SPROM - Project goals)

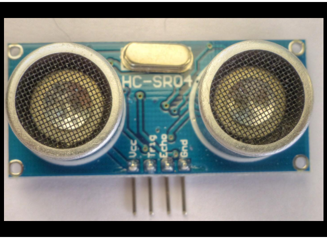
2) How we're doing it : (components - codes - prototype testing)

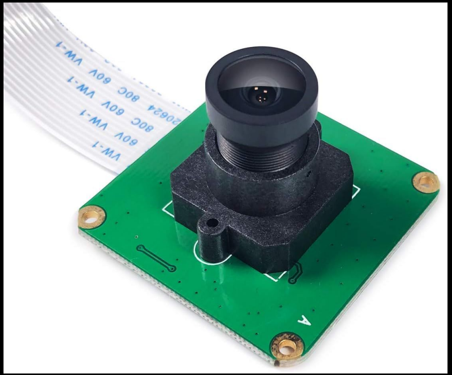
3) Conclusion

Motivation

- In precision agriculture, it is important to monitor the collection of sugarcane from the fields
- Currently, a truck harvests the sugarcane through a pipe and, upon arriving at the processing facility, a probe collects samples and deposits them into a bucket
- Trained machine learning models are used to detect the bucket's approach for sample collection, but maintaining these models is complex and specific to each installation.







Introduction to SPROM

What is SPROM?

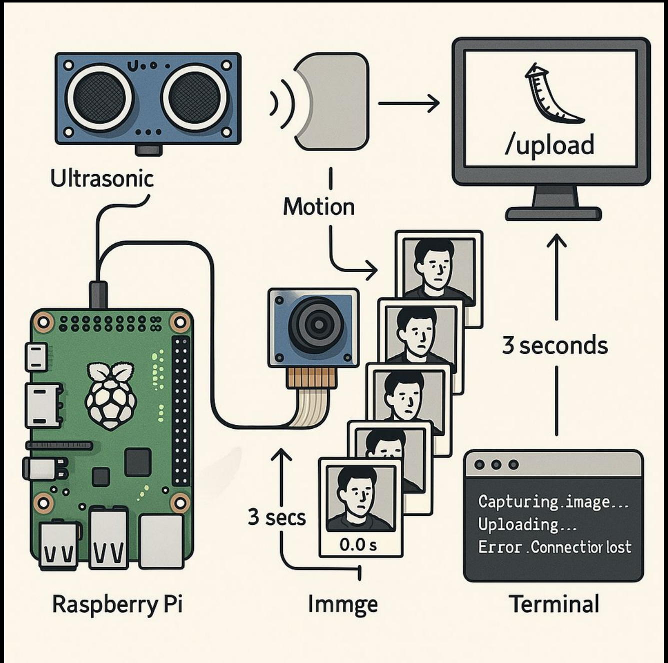
- A smart proximity detection system built around a Raspberry Pi and ultrasonic sensor
- Captures images of objects when they approach a defined zone
- Displays images on Flask Server
- Offers a lightweight, affordable alternative to complex machine learning systems

Introduction to SPROM

Project Goals

- Detect the approach of objects in agricultural contexts
- Serve as a reliable, low-maintenance, and affordable alternative to complex object detection systems
- Improve efficiency and accuracy in real-world scenarios like sugarcane sample collection
- Intended to be installed in fixed positions near sugar cane sampling stations zone, where object detection is critical
- Capture images of the bucket and display it on a Flask server

How we’re doing it



```
Capturing.image...
Uploading...
Error .Connectior lost
```

Codes

1. Ultrasonic Sensor (HC-SR04) continuously measures the distance to any object in front of it.
2. If an object is closer than 30 cm, the Python script:
3. Waits 2 seconds
4. Then captures 6 photos in 3 seconds (2 fps) using the Pi Camera.
5. Each photo is automatically sent to a Flask server running on my laptop.
6. The server receives and stores the images, displaying them in a live web gallery.
7. After upload, the images are deleted from the Raspberry Pi to save memory.

How we're doing it

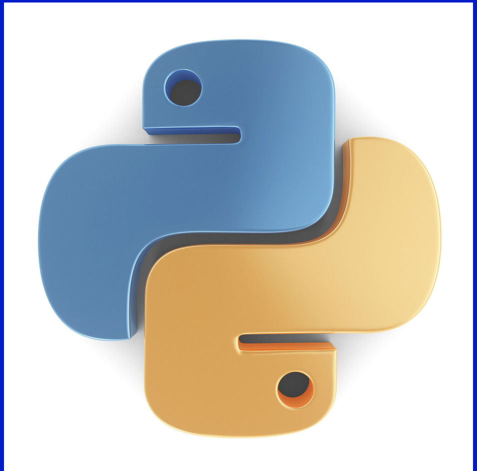
List of components :

Hardware requirements :

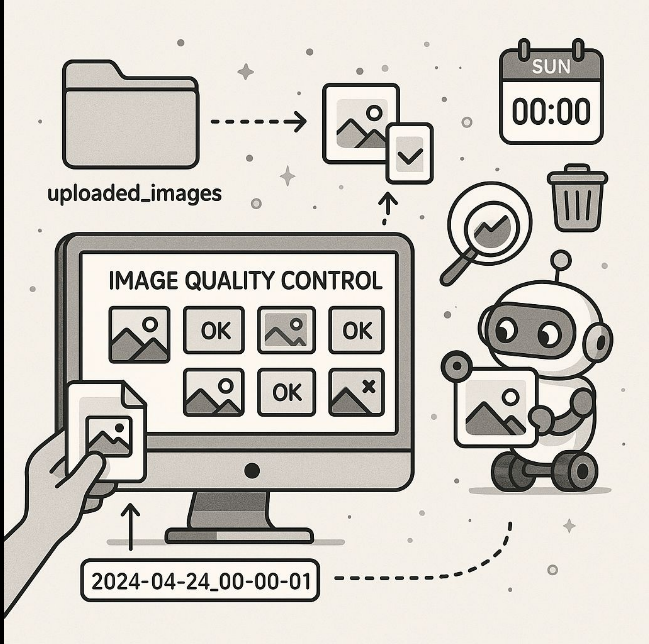
- Developer boards - Ultrasonic distance sensor, HC-SR04;
- Raspberry Pi Zero 2W, 4x 1GHz, 512MB RAM, WiFi, BT;
- Arducam 5MP OV5647 Miniature Camera Module with M6 Lens and Flexible Cable for Pi Zero, Pi 5 and Pi Compute Module;
- Wires and breadboard for the circuits (ELEGOO Jumper Wire Cable Male Female 200 mm Set of 50 Together with a 170 Contacts Breadboard;
- 1 kΩ and 2 kΩ resistors

Software requirements :

- Language: Python
- Frameworks: Flask (implements, Html, css, Javascript)



How we're doing it

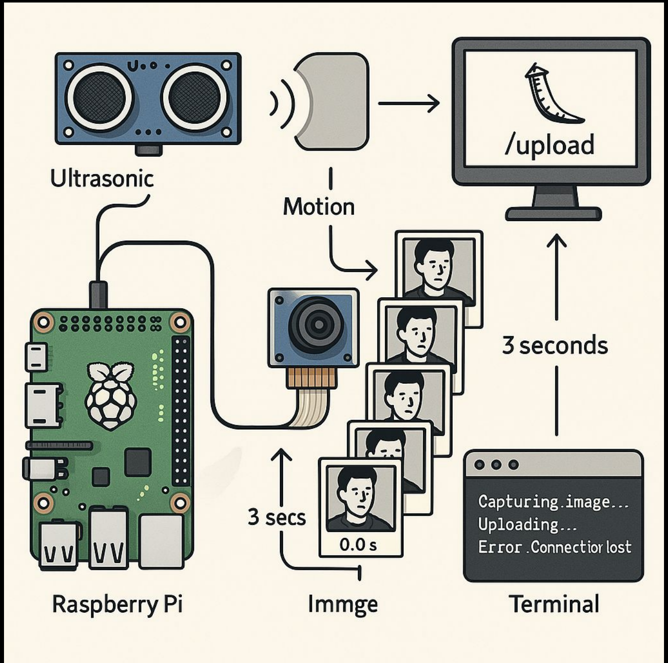


Our codes :

app_english.py – **Web App with Flask**

- **Uploading :** When someone uploads an image through the upload route, the app saves it in a folder called "uploaded_images".
- **Image Quality Control System** : Detects and classifies uploaded images as "OK" (sharp) or "Not OK" (blurry) using **OpenCV** and **Laplacian variance**.
- **Web Interface with Flask** : Displays all captured images in a gallery with filters (by date, time, status) and lets users manually update image status.
- **Image Upload Automation** : Uploaded images are renamed with a timestamp and saved in a folder to keep filenames unique and organized.
- **Automatic Weekly Cleanup** : Deletes all images every Sunday at midnight to keep the system lightweight.

How we’re doing it



```
Capturing.image...
Uploading...
Error .Connectior lost
```

Our codes :

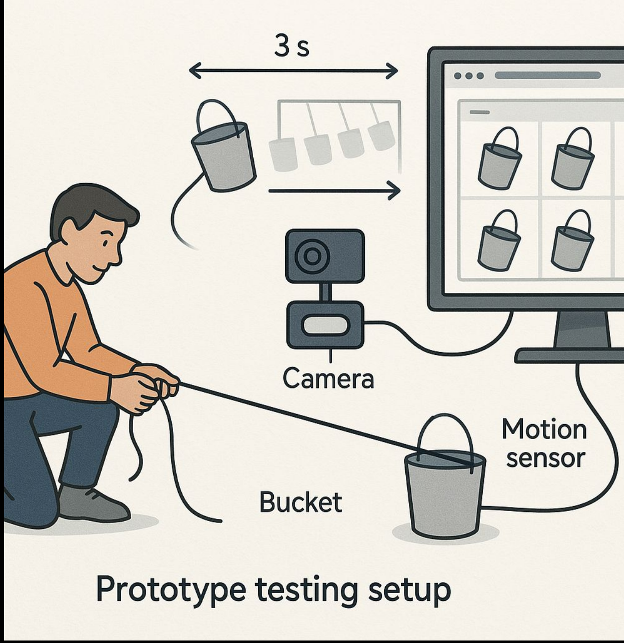
sensor_camera_uploader_english.py – **Sensor + Camera System**

- **Motion Trigger via Ultrasonic Sensor** : Detects nearby objects with an ultrasonic sensor connected to a Raspberry Pi.
- **Burst Image Capture** : Takes six images in three seconds when an object is detected, ensuring full coverage of the moment.
- **Automated Upload to Server** : Each captured image is sent immediately to the `/upload` endpoint of the Flask app.
- **Local Logging for Debugging** : Keeps a log of actions and any errors to help monitor the system's performance.

How we’re doing it

Prototype testing:

- Tie a rope on the bucket
- slide it in front of the camera sensor
- the camera will take 6 pictures in 3 seconds
- after on the website we can analyze the pictures



Prototype testing setup

Conclusion

The SPROM system is an effective alternative to complex machine learning models for object detection. By combining a simple ultrasonic sensor, Raspberry Pi, and a image processing web app built with Flask and OpenCV, we've created a lightweight, affordable, and reliable alternative for monitoring physical events such as sample collection in agriculture.

SPROM opens the door to **more accessible automation**, especially in environments where simplicity, speed, and precision matter.

Special thanks to

- Prof. Dr. Natividad Martínez Madrid for the guidance on defining the project scope

- Nico for soldering the Raspberry Pi header

- Rodney Santos for helping with the prototype assembly

Thank you