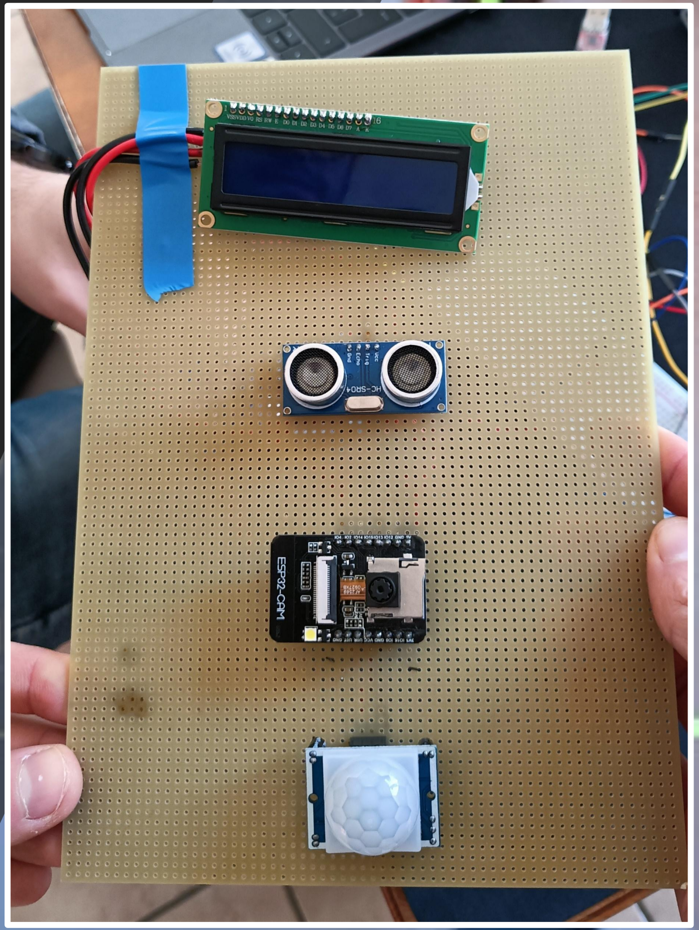
SECURE FACE

Embedded Software for the Internet of Things

Link to the YouTube video:

https://youtu.be/Rwgl8lMb24U



Problem Statement

Secureface is a mechanism to record and identify a person by his face.

- Can be used in Homing or industrial environments.
  - Examples:
    - Alarm systems
    - Security camera
    - Automatic door open

Working Scheme

START

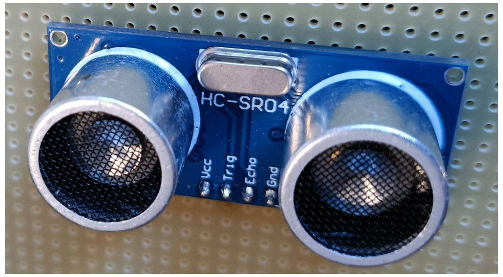
User runs the program

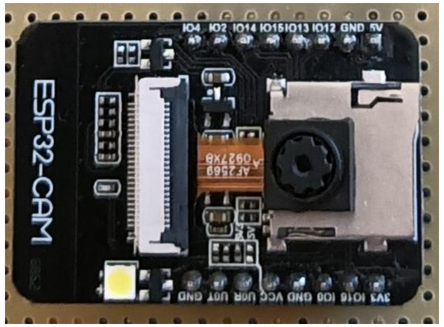
PIR Sensor

Capts a movement and makes the camera record a video

Ultrasonic sensor

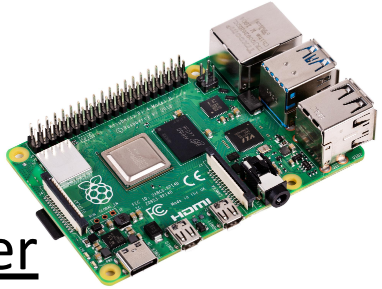
Waits: if a person comes next to it, it overrides the video process and starts the recognition script





ESP32cam Ai-thinker board

Sends photos via Wi-fi to the server

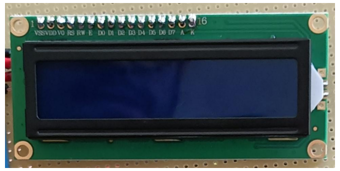


The raspberry server

Starts the recognition script and sends back the result or make the video

LCD display

Prints the name of the person or writes state info



Software Architecture

Esp32cam:

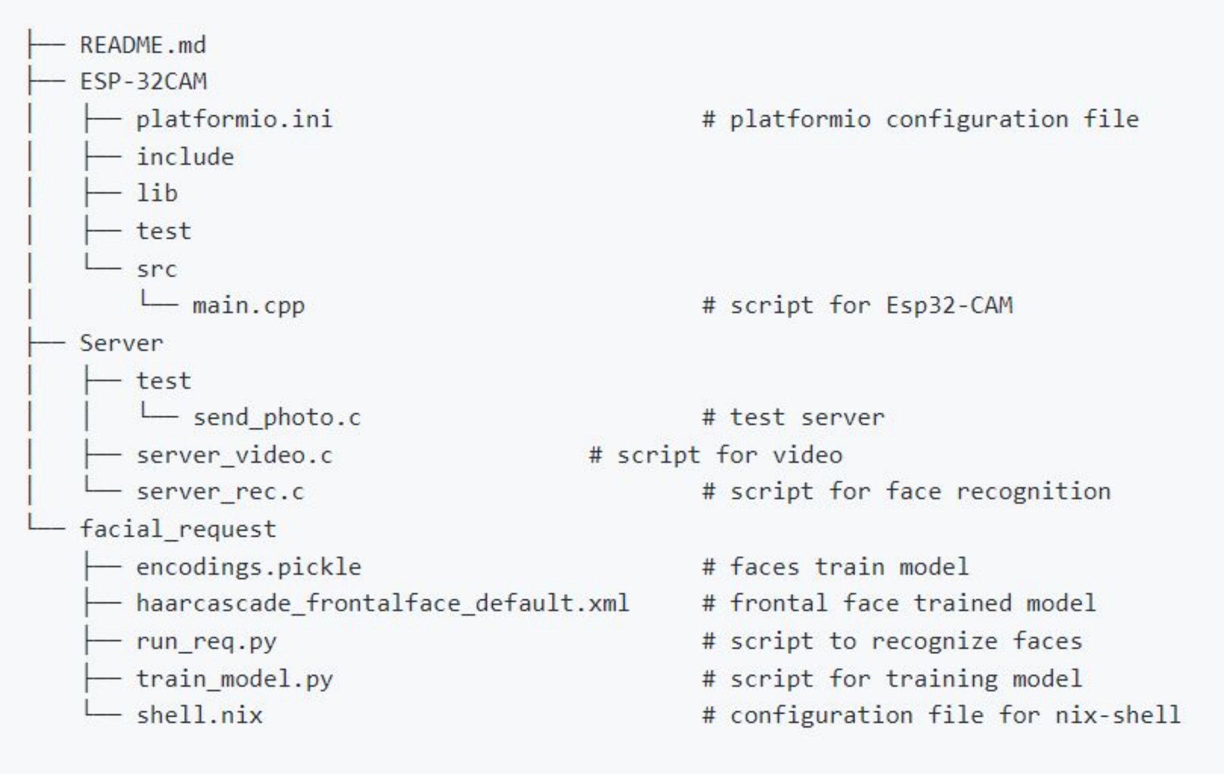
- Read sensor
- Send photo via tcp
- Display status and result

Server

- Receive images and create video
- Run face recognition process

Facial_rec:

- Train model
- Match images with model



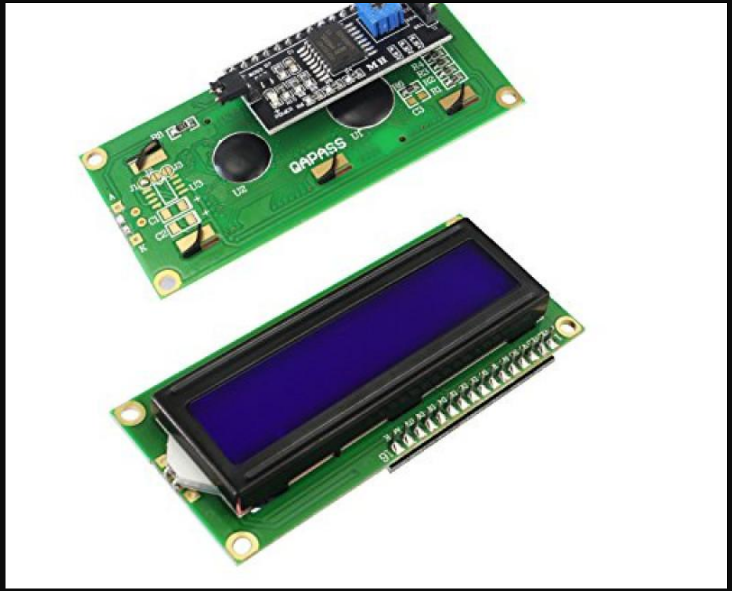
```
├── README.md
├── ESP-32CAM
│   ├── platformio.ini                          # platformio configuration file
│   ├── include
│   ├── lib
│   ├── test
│   └── src
│       └── main.cpp                            # script for Esp32-CAM
├── Server
│   ├── test
│   │   └── send_photo.c                        # test server
│   ├── server_video.c                  # script for video
│   └── server_rec.c                            # script for face recognition
└── facial_request
    ├── encodings.pickle                        # faces train model
    ├── haarcascade_frontalface_default.xml     # frontal face trained model
    ├── run_req.py                              # script to recognize faces
    ├── train_model.py                          # script for training model
    └── shell.nix                               # configuration file for nix-shell
```

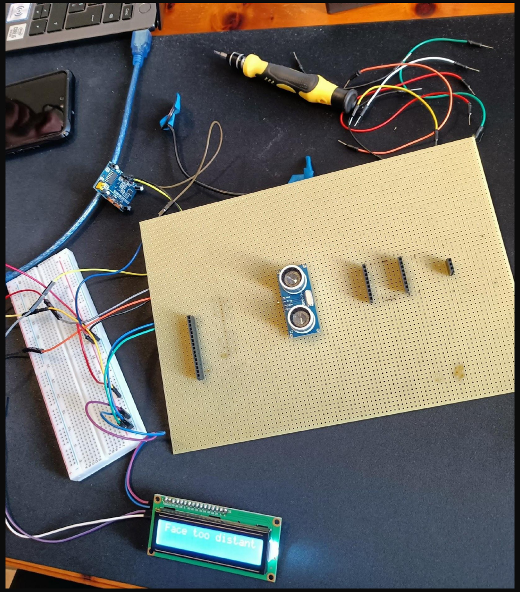
Testing

Software tests:

- Server test scripts
- Video & recognition tests

- Problems:
  - LCD pins were too many for the ESP32cam □ Got a new LCD with I2C module
  - Raspberry too weak □ More time for computing and testing
  - Double functional pin problems for pin0 and pin16





Conclusions and Future Work

- Implement a crypted communication
- Create a database on which save the camera's recordings
- Create an user interface
- Add a door opening actuator
- Add a mobile arm to camera and make a face tracking

Students (group 25):
Enrico Carnelos
Roberto Lorenzon
Fabio Grotto