Bachelor's Degree in Electronic, Telecommunications and Computers Engineering

P16: Home Energy Management System

Paulo Rodrigues | 47118 • Carlos Santos | 45938

INTRODUCTION

In the current context of seeking sustainable and efficient solutions in the energy sector, European energy cooperatives promote efficiency by implementing innovative systems. Our project introduces a smart home open-source system with sensors to track energy use, empowering users to make informed choices and reduce waste and costs. It also focuses on being able to use different hardware making it portable between implementations.

OBJECTIVES

- Implementation of an open-source solution;
- Implementation of an embedded circuit;
- Retrieve data using sensors and MCU;
- Home Assistant installed on a Raspberry Pi 5;
- Wi-Fi and MQTT configuration:
- Deliver data to from the sensors to a Raspberry Pi 5;
- Usage of the Home Assistant interface to make decisions.
- Commands sent from the Home Assistant to a Relay.

PROPOSED SOLUTION

After researching candidate components to use in this project, the proposed solution is the system shown in the figure below.

From right to left, an ESP32 is connected to sensors, such as a temperature and humidity sensor, a light sensor and a current sensor.

These sensors make measurements and send them to the ESP32 (Cofy-Cookie), then sending this data to a Raspberry Pi 5 (Cofy-Box), running the Home Assistant application, via MQTT protocol.

The user accesses the client interface of the Home Assistant application where he can see these measurements and take the actions, or program automation. This is done by sending the command from the Home Assistant to a relay, connected to the ESP32 and making it possible to control the said home appliance.

There was also a router configured with network credentials including the SSID and password, to ensure a secure connection and enabling the MQTT usage.

IMPLEMENTATION

The implementation involved several steps:

- Assembling and mounting the Cofy-Cookie on a breadboard, connecting each sensor to the ESP32, and successfully testing the measurements.
- Configuring a router with SSID and password for secure Wi-Fi.
- Setting up the Raspberry Pi, installing CasaOS, Home Assistant, and an MQTT broker, and connecting it to the network.

Configuring Home Assistant by writing a YAML file to read MQTT topics and displaying measurements and control buttons on the Home Assistant page for easy user interaction.

RESULTS

The Cofy-Cookie module successfully collected data from various sensors, including temperature, humidity, light intensity, and electrical parameters. The data was continuously monitored and transmitted to the Home Assistant for visualization and control.

The Home Assistant application was also able to control the Relay via MQTT command, achieving the project's original goal.

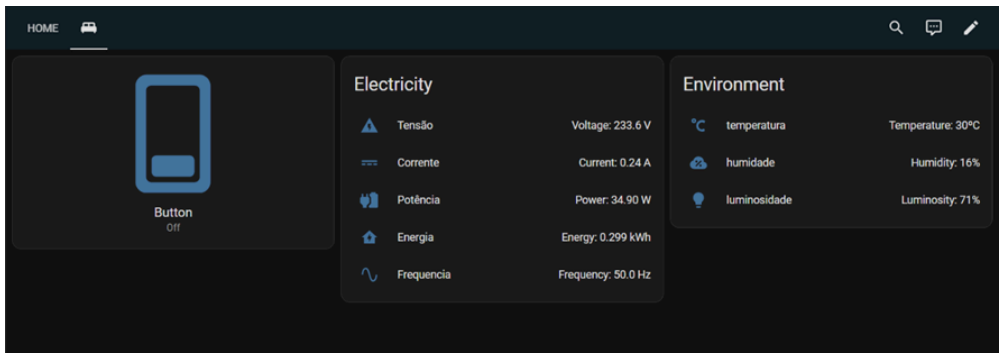

Fig. 2 - Home Assistant

CONCLUSION & FUTURE WORK

The project began with a survey of the required hardware specifications and the establishment of specific milestones to achieve the project's objectives. Following this, the implementation phase involved setting up the server, connecting various components, and ensuring energy-efficient operation. Through this approach, the project successfully met its goals, paving the way for further enhancements and expansions in home automation and energy monitoring systems.

Overall, the project was successfull, collecting sensor data, enabling Wi-Fi and MQTT communication, monitoring and controling the system throught the Home Asssistant application, all while maintaining low power consumption.

For future work, efforts could be directed towards incorporating different sensors with a bigger range of values to accomodate different regions worldwide, new areas of the house could be added to the Home Assistant application and even showing different views for the data received such as graphics. Additionally, new sensors, such as wind sensors, could be incorporated. Secutiry could also be an area to be enhanced, implementing SSL/TLS encryption for MQTT communication as well as embedding static CA certificates in the device firmware and regular firmware updates could enhance the system's defenses against potential threats.

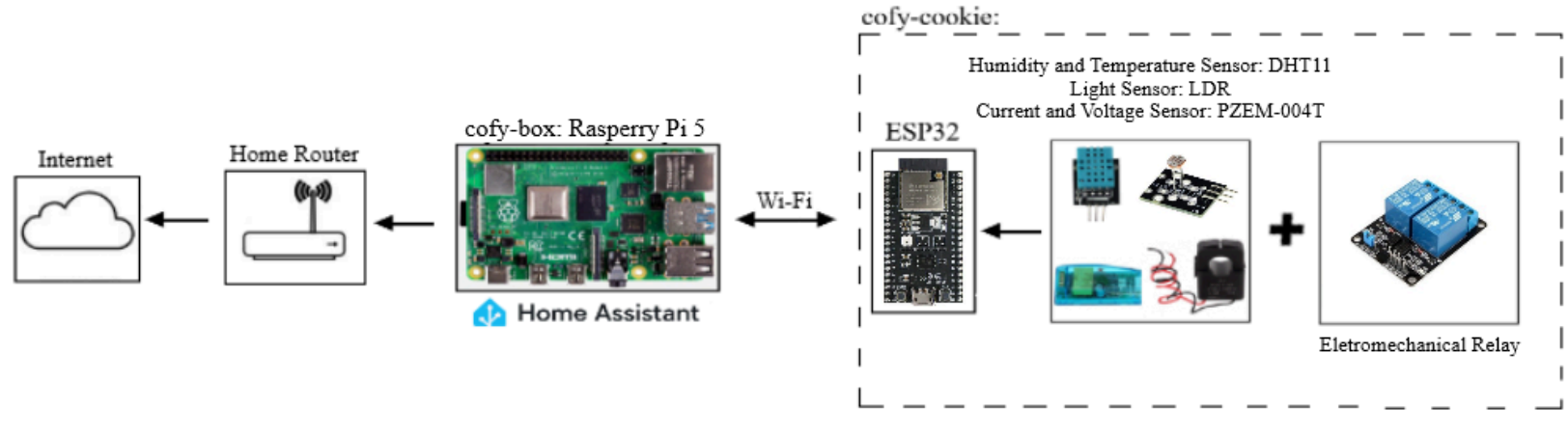

Fig. 1 - Project Scheme