Hackatown 2021

HydroSense
Power Consumption Monitoring Software

HydroSense

Made by :

teamName();

07/02/2021

Table of contents

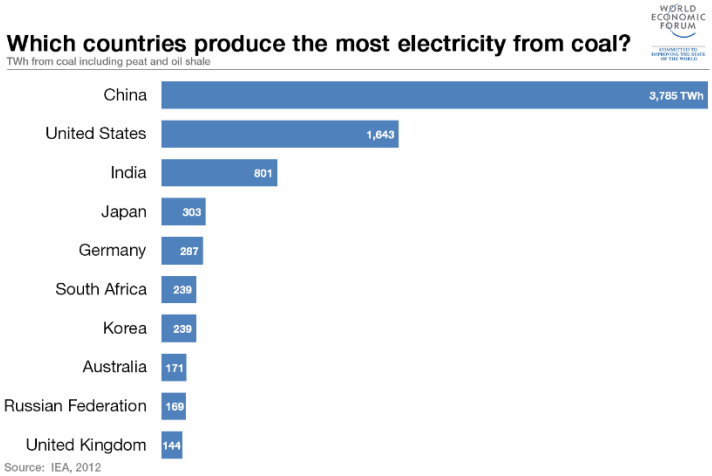
Introduction

Software designed for Hackatown 2021

For the 2021 Hackatown hosted by PolyHx, our team, teanName(); has chosen the “Efficient Usage of Energy” category. The objective of our application is to help the user to observe, analyze, predict, and manage their energy consumption. The application is called “HydroSense - Power Consumption Monitoring Software”. The name comes from “hydro” (hydroelectric plants, the way electricity in generated in Quebec) and “sense” (because of our use of sensors to find and calculate the power consumption).

HydroSense

We use electrical energy all the time in our daily lives. Everything from the lighting in our homes, offices and streets, cooking food, to our computers and internet, pretty much everything in our daily lives use electricity. Due to the (fairly) recent emergence of new technology, our power consumption has grown substantially and so has the energy production to keep up with that demand. This energy is a precious resource that is vital to the modern society. Currently, there are a lot of places in the world that use non-renewable energy resources to generate electricity.



So, if we can efficiently and effectively manage our power consumption, we can reduce our consumption and so we can reduce our carbon footprint. The goal of this project is to have access to your power consumption at all times on either your computer of your smartphone so that you can manage your power according to your needs.

The team

Faizan Ahmad

Hey there. I'm Faizan and I am a software engineering student. I am interested in financial technology, web, AI, machine learning, IoT, microcontrollers and electronics. In my spare time, I enjoy watching TV shows, investing in stocks, carving and building objects out of wood, playing video games and building electronics devices. I am in charge of team coordination and the hardware section of the HydroSense project.

Joseph Saliba

My name is Joseph and I am a passionate Software Engineering student at Concordia. When it comes to software development, I am really interested in Artificial Intelligence. I enjoy simulating scenarios and experimenting with AI in Game theory. In my free time I enjoy watching TV shows, playing chess, video games and hanging out with friends. I am in charge of the desktop program's database management and charts.

Usama Saleem

I'm a software engineering student at Concordia University. Currently, I work for Hewlett Packard and have a growing passion for graphic design in addition to exercising my love for coding. I am in charge of the visual aspects of the GUI and the logo design.

Marwa Khalid

My name is Marwa Khalid, an undergraduate student studying Computer Science at Concordia University, in Montreal. Currently, I am constantly exploring different programming languages and growing a passion for web development and security systems. I am in charge of the different GUI interfaces.

Disclosure

Our team is composed of 2nd year software engineering with limited experience in software development. This is our first Hackathon and to be honest, we don't expect to win, but we aim to do the best we can in the time we have. Due to some external factors (assignments, exams, work, lack of hardware and resources, etc.) and a teammate's birthday, we didn't have enough time to complete the project. On Friday, we split the task into two components (hardware and software) and assigned people to them based on their experience and expertise. One person had to worth on the hardware and the others on the software (desktop and mobile). So, we did the best we could in the limited time we had. This document covers all the hardware configuration needed and comes with a file containing the (incomplete) GUI code for a desktop application.

Note: for this project, we will only be measuring the AC current and not taking into account true, reactive apparent, and deformed power. We will calculate the power by assuming that everything in the house/office runs on 120V and that it is a resistive load.

Setup

Required components

Sensor module

- ESP32 Devkit
- 5V power supply (AC to DC) to power the ESP32
- Current sensor (DFRobot Gravity Analog Current Sensor)

Server

- Raspberry Pi 4 model B (4 GB)
    - Installed Raspbian
    - Set up a LAMP server
- 16GB microSD card
- Power supply (5V 2.5A minimum)
- Optional:
    - Case
    - Cooling fan
    - Ethernet cable

Modules

Sensor Module

The sensor module is based on 3 components: a microcontroller, a current sensor, and a power supply.

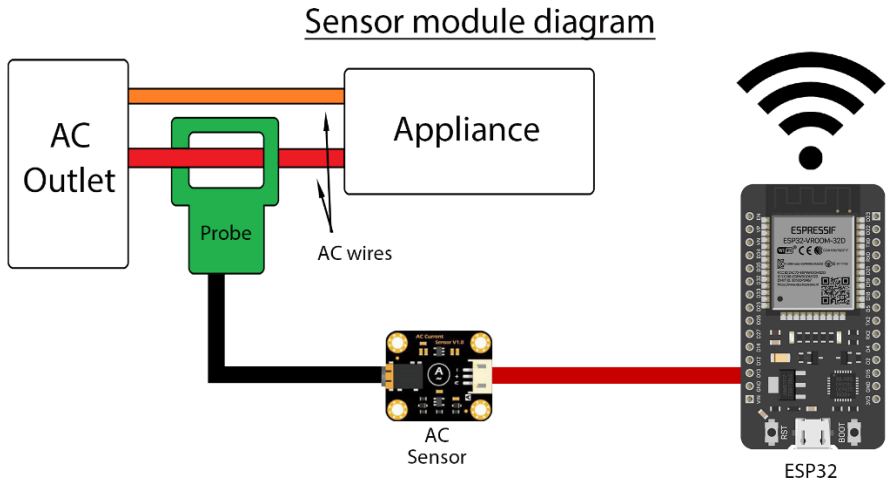
The heart of the sensor module is an ESP32. It's a low-cost and low-power system on a chip microcontroller that has integrated Wi-Fi and Bluetooth and is perfect for our sensor module.

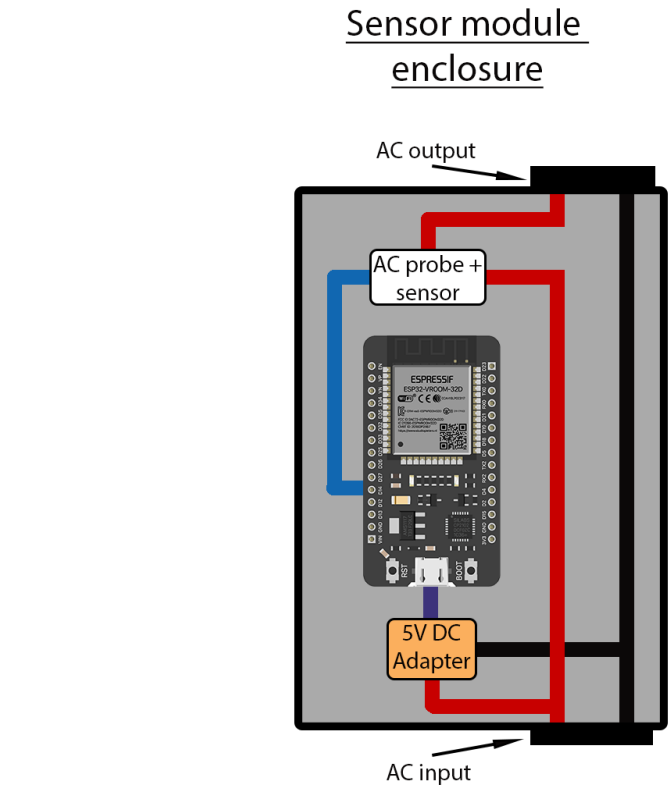
The DFRobot Gravity Analog AC Current Sensor can be installed without any non-destructive means (cutting wires and soldering). The clamp (blue object in the picture) can be connected directly on top of one of the AC wires (preferably the live wire) and interfaces with the signal conversion module that can be connected to a ESP32 microcontroller.

A generic 5V power supply. We only need the internal circuit of the black box.

Sensor module concept:

The idea of the sensor module is that it will monitor the current that passes through the probe from the AC outlet to the appliance. Note that we use the term “appliance” here and it can be anything from a light bulb to a fridge or computer. This current is quantified and sent through the ESP32 to ther local server (explained later) over wifi. The following diagram shows the general operation of the sensor module.

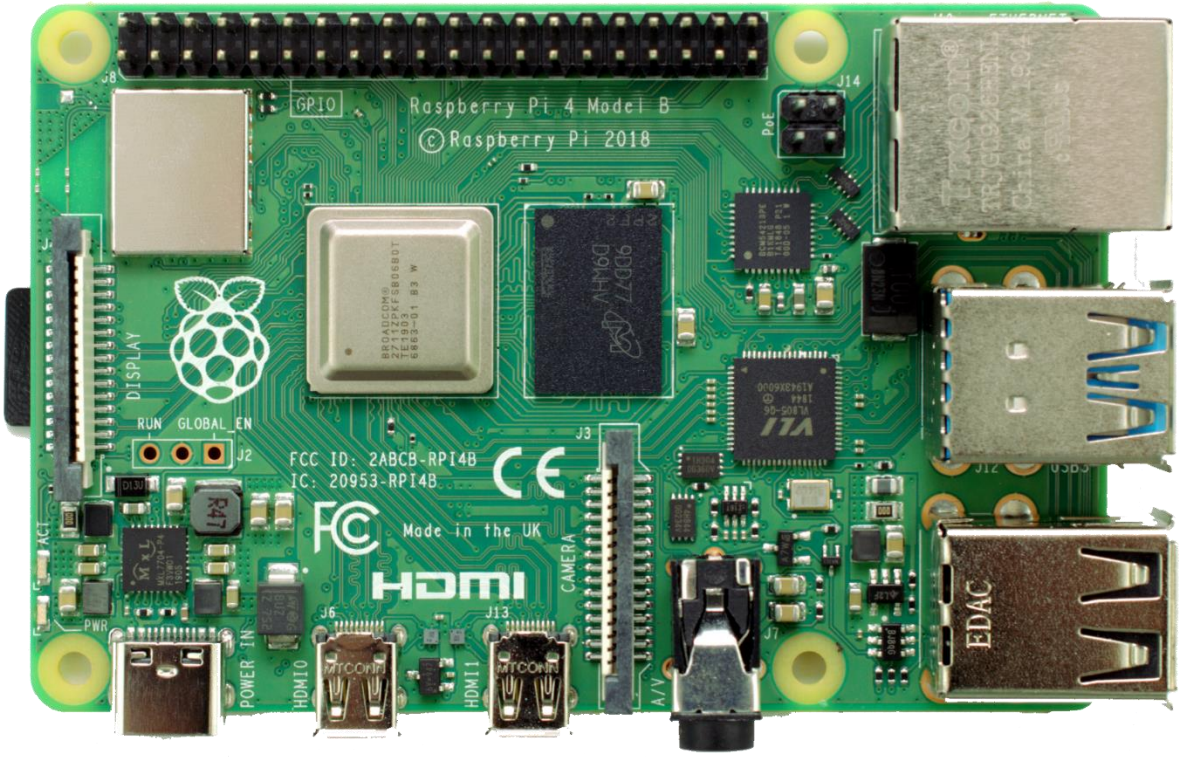




Server

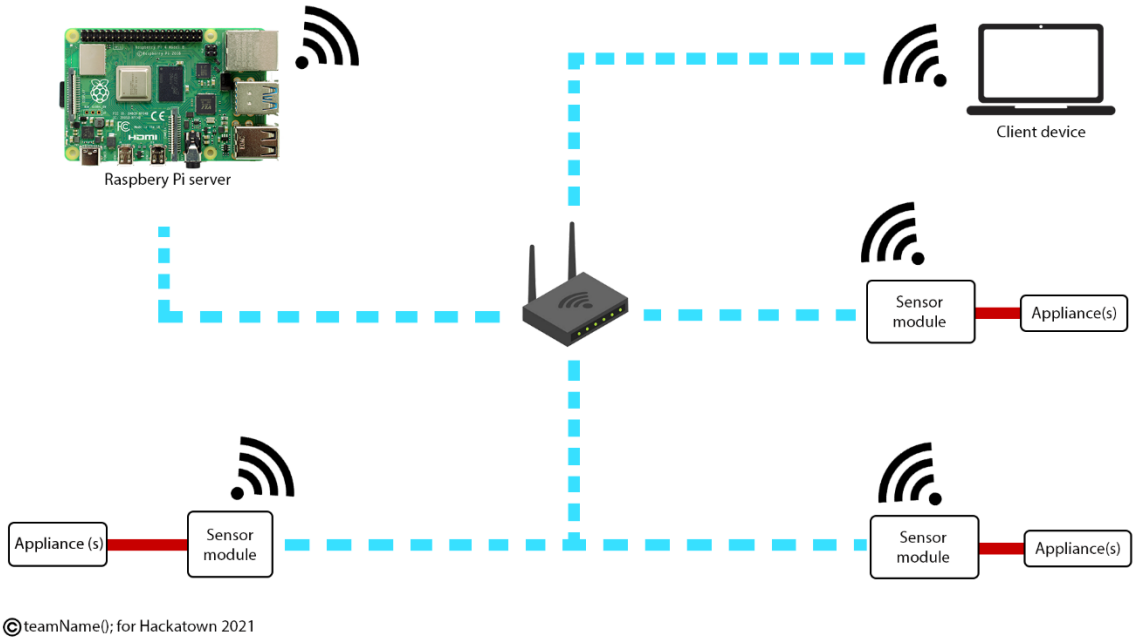
The idea of the sensor module is that it will monitor the current that passes through the probe from the AC outlet to the appliance. Note that we use the term “appliance” here and it can be anything from a light bulb to a fridge or computer. This current is quantified and sent through the ESP32 to ther local server (explained later) over wifi. The following diagram shows the general operation of the sensor module.

The “server” will be a Raspberry Pi (more details later) that will host a MySQL database and PHP app.

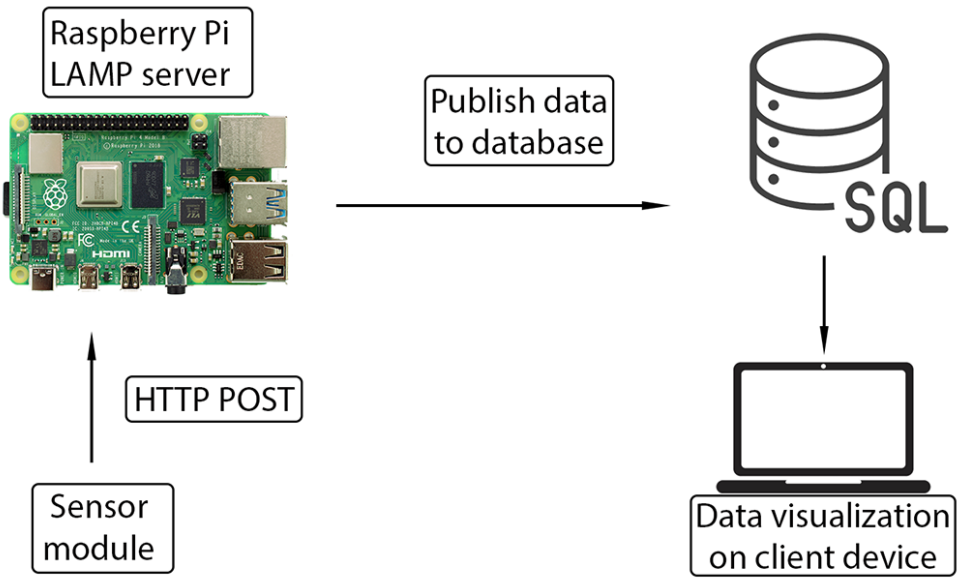


High-level overview

Complete system diagram



All the sensor modules will send their data to a Raspberry Pi server over a local Wi-Fi connection. The client device (on the same network) can access that data with the help of a program (for a desktop) or application (for a smartphone). Below is a more detailed diagram explaining how the sensor module and client device communicate with the server.



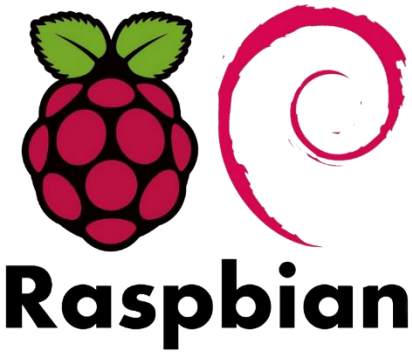
Hosting the database

This part of the project consists of running a LAMP server on a Raspberry Pi so that we can store the data from the sensor module and then help visualize it in the client device.

We start by installing the Raspbian operating system on the raspberry pi and configure it with a LAMP server and install phpMyAdmin (MySQL administration tools).

Using myPHPAdmin, we can now create a new database and then a SQL table.

Once that is done, we can make a php script that will receive the incoming data from the sensor module and place that data in the database. After that we can make another php script to display the data (and to send it to the client device).



Programming the ESP32

The ESP32can be programmed using the Arduino IDE. We should create a new class for reading the input from the AC sensor. Next, we can configure the main code with the network and server credentials and finish linking the sensor class and finalizing the loop code.