Arduino Project

Team Members: Hussein Jaber, Carlita Khawand and Enam Mikem.

Introduction

In this project, we are going to detect the temperature and the uv light. After sensing those data, we are going to send those values via an Http request in order to display the results on a web interface and plot different graphs showing us the change of those variables over time. We decided to take on this project because pollution is a real issue in the world.

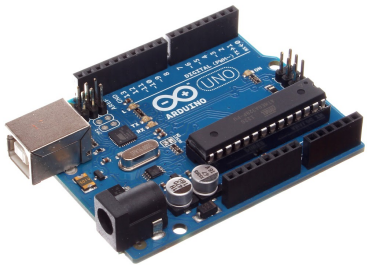
Components used

Arduino

Arduino is an open-source electronics platform that uses simple hardware and software to make it easy to use. Arduino boards can take inputs - such as light from a sensor, a finger on a button, or a Twitter message - and convert them to outputs - such as turning on an LED, triggering a motor, or publishing anything online. By providing a set of instructions to the board's microcontroller, you may tell it what to do. The Arduino programming language (based on Wiring) and the Arduino Software (IDE) (based on Processing) are used to accomplish this.

We used arduino with different sensors to record data from the environment and send it to our database via http requests.

Arduino wifi rev 2



The Arduino UNO WiFi Rev.2 is the easiest point of entry to basic IoT with the standard form factor of the UNO family. The Arduino UNO WiFi Rev.2 is your one-stop-solution for many of the basic IoT application scenarios, whether you're seeking to establish a sensor network connected to your workplace or home wifi, or a BLE device delivering data to a cell phone.

When you plug this board onto a device, you may use its secure ECC608 crypto chip accelerator to connect it to a WiFi network. The Arduino Uno WiFi has the same functionality as

the Arduino Uno Rev3, but with WiFi/Bluetooth and a few more improvements. It contains an inbuilt IMU (Inertial Measurement Unit) LSM6DS3TR and uses Microchip's brand new ATmega4809 8-bit microprocessor.

Arduino sensors

A sensor is a device, module, machine, or subsystem that detects events or changes in its surroundings and transmits the data to other electronics, most commonly a computer processor. Sensors are always used in conjunction with other electronics.

We used arduino sensors to record data such as humidity or temperature then send it to our database through http requests

Softwares used

Arduino ide

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

We used arduino ide to write our code and upload it on our board to receive data from sensors and send it.

Vscode

Microsoft's Visual Studio Code is a source-code editor for Windows, Linux, and macOS. Debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git are among the features. Users can customize the theme, keyboard shortcuts, and preferences, as well as install extensions that offer new features.

We used it to write the code for our web page back-end and front-end.

Mysql

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.
The data received from arduino is sent to mysql and will be used to display charts.

Technologie used:

Node.js

Node.js is a cross-platform, open-source back-end JavaScript runtime environment that uses the V8 engine to execute JavaScript code outside of a web browser. Node.js allows developers to utilize JavaScript to create command-line tools and server-side scripting, which involves running scripts on the server before sending the page to the user's browser. As a result, Node.js symbolizes a "JavaScript everywhere" paradigm,bringing web-application development together around a single programming language rather than separate languages for server-side and client-side scripts.

Nodejs was used to build the backend of our web page and to use socket.io.

Apache

The Apache HTTP Server, also known as Apache, is a free and open-source cross-platform web server software that is distributed under the Apache License 2.0. Under the aegis of the Apache Software Foundation, Apache is created and maintained by an open community of developers.

Apache was used to run our webpage and look at the data received from the arduino sensors.

Librarie used:

Socket.io:

Socket.IO is a JavaScript library for realtime web applications. It provides instantaneous, bi-directional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for Node.js. Both components have a virtually identical API. Like Node.js, it is event-driven.

Socket.IO largely employs the WebSocket protocol with polling as a fallback option,while giving the same interface. Although it can be used as just a wrapper for WebSocket, it has many additional features, like broadcasting to many sockets, storing data connected with each client, and asynchronous I/O.

Socket.io is used in our application to update charts instantaneously once data is added in the database.

Canvas.js

CanvasJS is a JavaScript and HTML5 charting toolkit based on the Canvas element. It works on a variety of platforms, including iPhone, iPad, Android, Windows Phone, Microsoft Surface, and desktop computers. This enables you to design complex dashboards that run across all platforms without sacrificing your web application's maintainability or functionality. CanvasJS comes with stunning themes and is over 10x faster than traditional Flash and SVG Charts, resulting in dashboards that are lightweight, gorgeous, and responsive.

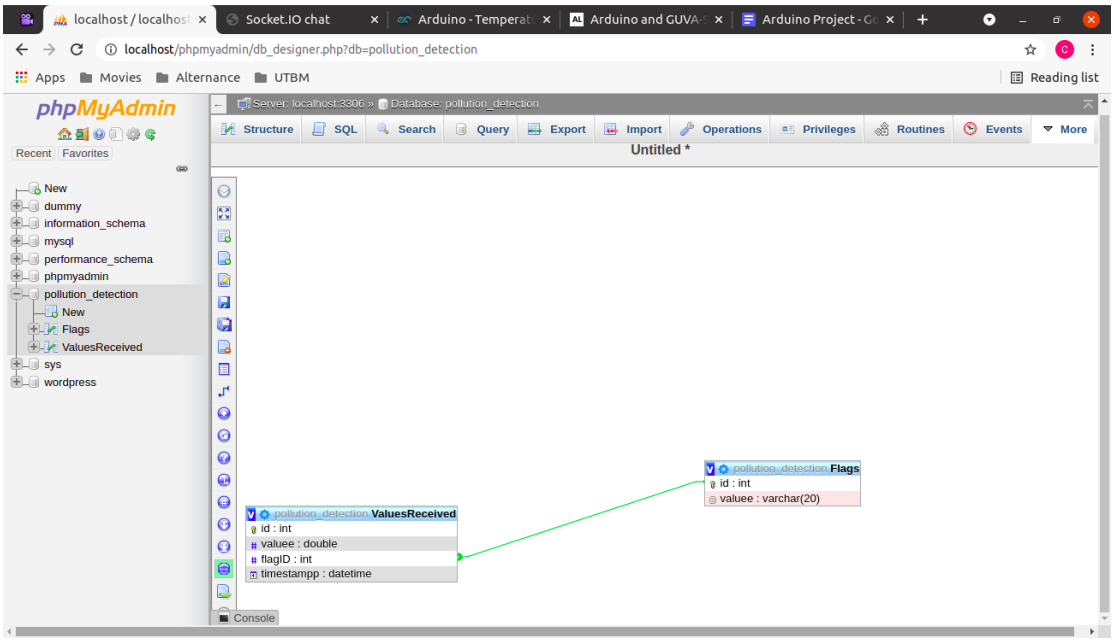
Canvasjs is used in our project to display data received from arduino into charts.

How it works

Our Arduino wifi board is connected to sensors that capture data such as the humidity and the temperature from the environment. This data is posted to our data/*base using http requests.

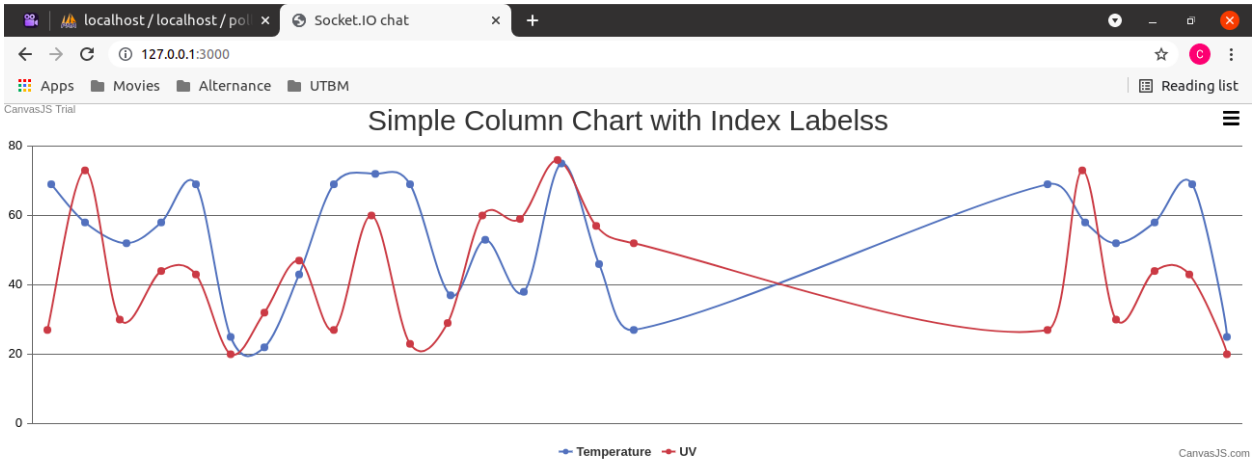
Using apache we run our webpage that can display charts based on the data received from arduino.
Periodically , using nodejs and socket.io , we check for any changes in the database and if new data was received , we add this data to our charts , classified by types (humidity , temperature). The data is added into a series that will be rendered using canvasjs.

Database Structure



Tables: Flags(int:id, varchar(20):valuee).
ValuesReceived(int:id, double:valuee, int:flagID (foreignkey),, datetime:timestamp).

Results



This graph displays how the temperature and UV lights are changing over time. Blue is for Temperature and pink for UV.