International Journal of Engineering and Techniques - Volume 10 Issue 4, August 2024

RESEARCH ARTICLE OPEN ACCESS

Real-Time Data Capture (RDC) System for Greenhouse Gas Monitoring: A Technology Approach to Addressing Environmental Concerns in Rivers State

Engr. Orimadike Okechukwu Stanislaus [1], Dr. Okengwu Ugochi Adaku [2], Prof. Omijeh Bourdillon Odianonsen [3]
1(Centre for Information and Telecommunications Engineering, University of Port Harcourt, Nigeria.
orimadikeokechukwu@gmail.com)
2 (HOD, Computer Science, University of Port Harcourt, Nigeria.
ugochi.okengwu@uniport.edu.ng)
3 (Director, CITE, University of Port Harcourt, Nigeria.
bourdillon.omijeh@uniport.edu.ng)

Abstract:

In 2023, global greenhouse gas emissions reached 37.4 gigatonnes, pushing global temperatures above 1.5°C over pre-industrial levels—an alarming climate crisis. While science-based technological advances in clean energy are progressing, industrial activities in Nigeria, particularly in Port Harcourt and the oil-producing region, continue to have a significant impact. Unfortunately, the data capture systems in these areas are unreliable, relying on human-monitored measurements and often unreliable physical data transmission methods. Accurate data collection is crucial for combating emissions. This paper presents the design of a real-time data capture system for greenhouse gas emissions, focusing on sensor selection, MCU comparison, IoT architecture, and data transmission and storage techniques. The system utilizes MQ22 sensors, the ESP32 MCU, and MQTT transmission protocol linked to the GCP cloud, providing regularly updated CSV files. The design was simulated using the Wokwi online simulator, and an interactive web app was developed. The study demonstrates the effectiveness of this IoT system, highlighting its potential for real-time monitoring and data-driven management. This research contributes to environmental monitoring and paves the way for future improvements, such as integrating new sensor technologies, enforcing stricter emission standards, and expanding the research scope. Addressing this significant issue is crucial for progressing toward a sustainable future.

Keywords — **Realtime, Data, IoT, Sensor, Microcontroller, Greenhouse gas, emissions, climate change.**

I. INTRODUCTION

Humanity is currently facing unprecedented climate change, with greenhouse gas emissions reaching record levels in 2023. These emissions have caused global average temperatures in September 2023 to rise more than 1.5°C above pre-industrial levels, signaling a rapidly escalating climate crisis. The European Union's Copernicus Climate Change Service warns that 2024 could potentially be the hottest year on record (EGR, 2023). This trend, combined with extreme weather events, highlights the urgency of addressing climate change. The Intergovernmental Panel on Climate Change (IPCC) predicts that if global warming continues unchecked, the world will experience increasingly severe natural disasters, including hurricanes, wildfires, droughts, and floods (IPCC, 2022).

Despite some relief from the global energy crisis due to declining fossil fuel prices, the energy sector remains the largest source of air pollution. This affects over 90% of the global population and contributes to millions of premature deaths each year (WEO, 2023). The World Health Organization (WHO) identifies air pollution as the most significant environmental risk, responsible for 3 million deaths annually due to outdoor air pollution exposure. In 2012, outdoor air pollution was linked to 6.5 million deaths, accounting for 11.6% of global deaths, with 94% occurring in low- and middle-income countries. These deaths are primarily due to non-communicable diseases like cardiovascular diseases (CVDs), chronic obstructive pulmonary disease (COPD), and lung cancer. Industrial activities are a major source of air pollution, with regions like the Niger Delta and Port Harcourt suffering from poor air quality. In 2017, thick plumes of soot from oil and gas emissions in Port Harcourt led to increased morbidity and mortality rates among low-income families (Waxman et al., 2020).

In response to these challenges, technological advancements and clean energy solutions, such as solar photovoltaic (PV) systems, electric vehicles (EVs), and the Internet of Things (IoT), offer a promising path forward. IoT technology is playing a crucial role in shaping the global CO_2 emissions trend. Stimulated by COVID-19 stimulus packages, the adoption of IoT and clean energy technologies has accelerated significantly since 2019. Although total energy-related emissions increased by around 900 Mega Tonnes (Mt) between 2019 and 2023, the expansion of key clean energy technologies—solar PV, wind power, nuclear power, heat pumps, and electric vehicles—along with advancements in artificial intelligence, has significantly mitigated this increase. Without these technologies, the rise in emissions would have been three times greater (IEA CO_2 Emissions Report, 2023).

IoT presents a promising solution to the challenges of monitoring greenhouse gas emissions. In 2013, the Global Standards Initiative on the Internet of Things (GSI) defined IoT as "the infrastructure of the information society." IoT leverages existing network infrastructure to remotely monitor and control objects, enhancing efficiency, accuracy, and financial value while reducing manual processes by integrating the physical world with computer-based systems. This research focuses on integrating a real-time data capture (RDC) system for greenhouse gas monitoring. It underscores the importance of developing monitoring systems to track and analyze greenhouse gas emissions, facilitating the transition to a low-carbon economy. Accurate, real-time data collection is essential for understanding the current emissions landscape and addressing the primary sources contributing to the issue.

Traditional CO_2 monitoring systems face several limitations, including high costs, limited coverage, and data accuracy issues. IoT systems offer significant advantages, such as real-time data collection, high spatial and temporal resolution, and the ability to provide up-to-date insights into emission trends (Narayana et al., 2024). These capabilities are crucial for timely decision-making and policy formulation, making IoT an essential tool for environmental monitoring and localized environmental assessments.

This paper focuses on the design of a real-time data capture system for monitoring greenhouse gas emissions, emphasizing sensor selection, MCU comparison, IoT architecture, and data transmission and storage techniques. The system utilizes the MQ22 sensor and ESP32, with the MQTT transmission protocol linked to the GCP cloud for regular CSV updates. Simulation was conducted using the Wokwi online simulator prior to construction, and an interactive web app was developed. This research contributes to environmental monitoring and lays the groundwork for future studies. The integration of new sensor technologies, the implementation of stricter emission standards, and an expanded research scope could lead to even more effective outcomes. This research addresses a critical issue of our time and represents a step towards a more sustainable future.

II. LITERATURE REVIEW

The literature reviewed reveals several critical gaps in current approaches to greenhouse gas (GHG) monitoring, particularly in the context of real-time data capture and analysis.

Borhan and Khanaum (2022) provide a comprehensive overview of GHG measurement methods and sensors used in livestock production facilities, highlighting the importance of accurate data collection and the application of quality control protocols. However, the study underscores significant challenges in integrating multiple measurement systems due to the need for interdisciplinary expertise. Additionally, the complexity of combining information from various sources remains a critical issue, which limits the effectiveness of these systems in providing consistent, long-term monitoring.

Guzman et al. (2023) showcase the potential of real-time monitoring tools in reducing GHG emissions and enhancing the energy efficiency of carbon-intensive assets. While the tool successfully integrates emissions and performance data into a single platform, its scalability and real-time capabilities introduce challenges in managing and integrating large volumes of data. This limitation points to a broader issue within the field: the difficulty in handling real-time information efficiently, which is crucial for supporting critical decision-making and reducing environmental impacts.

Haidarzhy (2023) discusses the application of IoT in environmental monitoring, emphasizing the benefits of real-time data collection in promptly identifying and mitigating environmental anomalies. However, the paper also highlights a significant gap in data integration from various IoT devices, which hampers the overall effectiveness of these systems in regulatory compliance and public awareness efforts. This shortcoming is a recurring theme in the literature, indicating a widespread need for more robust data integration frameworks.

Bersani et al. (2022) explore the use of IoT technologies in intelligent greenhouses, noting their potential to optimize resources, improve production quality, and reduce environmental impact. Despite these advantages, the study identifies several challenges, including sensor calibration, noise amplification, and the complexity of integrating data from multiple sensors and controllers. These issues reflect broader difficulties in the IoT field, where the integration and calibration of diverse data sources often complicate the implementation of effective monitoring systems.

The APEC report (2024) on GHG monitoring technologies highlights the benefits of combining surface-level monitoring systems with remote sensing data for more comprehensive GHG analysis. However, the report identifies a significant gap: the lack of standardized data collection and integration protocols. This absence undermines the potential effectiveness of these technologies, as inconsistent data standards can lead to inaccurate or incomplete monitoring outcomes.

Da Costa et al. (2024) examine the role of real-time data monitoring in Life Cycle Assessment (LCA) studies, acknowledging the potential of IoT technologies to enhance data accuracy and timeliness. However, they also point out the challenges of integrating real-time, high-frequency, and multidimensional data into existing LCA frameworks. The need for advanced data analytics and machine learning algorithms to generate actionable insights further complicates the integration process, revealing a critical gap in the current application of IoT in LCA studies.

The proposed Real-Time Data Capture (RDC) System for Greenhouse Gas Monitoring is designed to address several of these critical gaps:

1. Real-time measurements: A significant gap identified across the literature is the absence of real-time monitoring capabilities, which limits the ability to capture sudden changes in GHG concentrations. The RDC system directly addresses this by offering real-time monitoring, thus enhancing the detection of anomalous or time-variant gas concentrations.
2. Sparse spatial coverage: Many existing monitoring methods suffer from limited spatial coverage, often relying on fixed-point or mobile measurements that fail to provide comprehensive data. The RDC system's distributed sensor network ensures broader spatial coverage, allowing for more accurate and detailed monitoring across entire areas.
3. Data integration challenges: The literature repeatedly highlights difficulties in integrating data from multiple sources and sensors. The RDC system tackles this issue by consolidating data into a single platform, streamlining the integration process and improving the overall efficiency of GHG monitoring.
4. Data transmission and storage: The management of large volumes of data is a common challenge in existing systems, particularly regarding efficient transmission and storage. The RDC system leverages advanced IoT technologies and cloud-based storage solutions to overcome these

limitations, facilitating easier data handling and access.

5. Lack of advanced data analysis tools: The literature points out that many current systems provide raw data without the necessary tools for sophisticated analysis. The RDC system incorporates state-of-the-art analytics and machine learning algorithms, enabling more meaningful insights from the collected data.
6. Environmental adaptation: Finally, the literature indicates that existing systems are often poorly adapted to local environmental conditions. The RDC system is specifically tailored to the local environment, climate, and industrial profile, ensuring its relevance and effectiveness in various settings.

In conclusion, the proposed RDC system not only addresses the critical gaps identified in the literature but also offers a more comprehensive and adaptable approach to GHG monitoring, contributing to more effective environmental management and decision-making.

III. METHODOLOGY

A. System Architecture

This is essential for our IoT solution aimed at monitoring and predicting greenhouse gas emissions. It offers a comprehensive strategy for integrating sensors, microcontrollers, communication protocols, and scalable data storage. The goal is to achieve efficient, reliable real-time data collection, transmission, and analysis. The architecture of our IoT system includes multiple gas sensors strategically deployed to detect and monitor the presence and concentration of various gases in the environment. These highly accurate gas sensors convert raw analog inputs into electrical signals whenever specific gases are detected. The system architecture is depicted below.

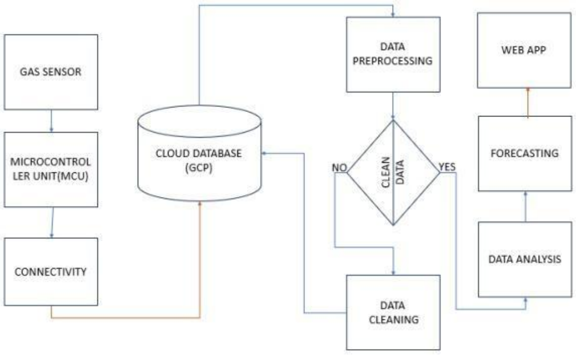


Figure 1: Block flow diagram of system workflow

1. Sensor and Microcontroller Layer

The sensor layer comprises gas sensors connected to a microcontroller unit (MCU), which processes the raw electrical signals from the sensors. The MQ2 sensor (shown in Figure 2) stands out as a superior choice compared to the TGS 813 and SGX MICS sensors, particularly for detecting gases like flame, smoke, alcohol, propane, butane, hydrogen, methane, and carbon monoxide. Its wide detection range of 200 to 10,000 ppm allows it to monitor both low and high concentrations of gases, making it versatile for various applications. Unlike the TGS 813, which requires high temperatures to function effectively, the MQ2 operates efficiently at standard room temperature, simplifying system design and reducing overall power consumption.

Cost is another advantage of the MQ2. It is more affordable than the SGX MICS sensors, making it a practical option for mass production and budget-conscious projects. Additionally, the MQ2 sensor is widely available, often as part of a Chinese clone of the TFA200, and benefits from extensive documentation and strong community support. This ease of access and troubleshooting makes the MQ2 a reliable and cost-effective solution for environments where multiple gases need to be monitored simultaneously. A table of comparison is found below.

Table 1: Comparison of sensors

Feature	**MQ2 Sensor**	**TGS 813 Sensor**	**SGX MICS Sensor**
Detectable Gases	LPG, Smoke, Alcohol, Propane, Hydrogen, Methane, Carbon Monoxide	Carbon Monoxide, Methane, LPG, Alcohol	VOCs, CO, Ozone, Ammonia, NO_2, NOx
Sensitivity Range	200 to 10,000 ppm	High sensitivity to target gases	High sensitivity to target gases
Sensing Material	Tin Dioxide (SnO_2)	Tin Dioxide (SnO_2)	Tin Dioxide (SnO_2)
Operating Voltage	5V DC	5V DC	Varies (typically low voltage)
Power Consumption	Approximately 800mW	Varies, dependent on internal heater usage	Low power requirements
Operating Temperature	Room temperature	300°C to 550°C	Room temperature to moderate temperatures
Response Time	Fast	Moderate to fast	Rapid
Form Factor	Moderate	Moderate	Small form factor
Suitability	General gas detection, safety alarms	Industrial and environmental monitoring	Portable, battery-powered devices
Unique Characteristics	Low-cost, versatile detection capabilities	Requires a high temperature of about 185°C for optimal performance	Compact and can be used for indoor air quality monitoring

Figure 2: MQ2 gas sensor, Lastminute engineers, 2024

Microcontroller units like the Arduino Uno, Raspberry Pi Zero 2 W, ESP32, and STM32 are essential tools for various electronics and IoT projects. The MCU, performs analogue-to-digital conversion, encoding gas concentrations in parts per million (ppm) or similar units. Additionally, the MCU is programmed to perform basic data validation, ensuring the reliability of the data before transmission.

The Arduino Uno R3, based on the Atmega 328 chip, is a versatile microcontroller easily programmable via USB using Arduino's IDE, supporting C and C++. With its flexible I/O pins, the Arduino Uno is a popular choice for robotics, automation, and prototyping projects, especially given its open-source hardware and extensive community support.

The Raspberry Pi Zero 2 W is a compact single-board computer with a 64-bit quad-core ARM Cortex-A53 processor. It retains the small form factor of its predecessor while offering enhanced processing power and energy efficiency. This makes it suitable for IoT, educational, and hobbyist projects that require a balance of performance and compactness. The STM32 microcontroller family, based on the ARM Cortex-M3 core, offers diverse configurations suitable for a wide range of embedded applications, from simple devices to complex real-time systems, providing flexibility and performance in various project needs.

The ESP32 (shown in the figure 3), selected for this research, is a powerful microcontroller known for its dual-core processor, which supports resource-intensive tasks and real-time processing. Its integrated 2.4 GHz Wi-Fi and Bluetooth make it ideal for IoT and smart device applications. The ESP32's rich I/O features, including multiple GPIOs, ADCs, and DACs, allow for seamless integration with various peripherals and sensors. Its low power consumption and multiple power-down modes make it particularly suitable for battery-powered, portable devices. The strong community support, abundant resources, and permissive open-source license further enhance its appeal, making the ESP32 an excellent choice for both prototyping and deployment in research projects. A table of comparison is found below.

Table 2: Comparison of microcontrollers

Feature	Arduino Uno R3	Raspberry Pi Zero 2 W	ESP32	STM32
Processor	ATmega328P	64-bit Broadcom BCM2710 A1 quad-core Cortex-A53	Dual-core Tensilica Xtensa LX6	ARM Cortex-M3
Clock Speed	16 MHz	1 GHz (up to 1.3 GHz overclocked)	Up to 240 MHz	Up to 72 MHz
Memory	2 KB SRAM, 32 KB Flash	512 MB RAM	520 KB SRAM, up to 4 MB Flash	Up to 1 MB Flash, Up to 96 KB SRAM
Wireless Connectivity	None	Wi-Fi (802.11 b/g/n)	Wi-Fi (802.11 b/g/n), Bluetooth 4.2	Varies (available in some models)
I/O Pins	14 Digital I/O, 6 Analog Inputs	40-pin GPIO header	32 GPIO	Up to 80 GPIO
Power Supply	5V (recommended input 7-12V)	5V via micro USB	2.3V - 3.6V	2V - 3.6V (depending on model)
Special Features	PWM outputs, ADC	Mini HDMI, USB OTG	Low power consumption, dual-core, DAC	USB, CAN, Ethernet, advanced peripherals
Form Factor	Standard Arduino footprint	Compact, credit-card size	Compact, various form factors	Various form factors
Primary Use Cases	Education, prototyping, simple automation	Education, media center, IoT projects	IoT, home automation, portable devices	Industrial, automotive, consumer electronics

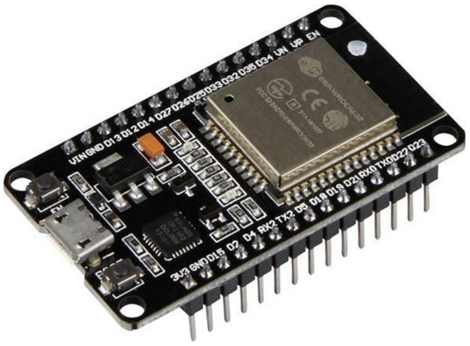


Figure 3: ESP 32 microcontroller, (Indiamart, 2024)

2. Connectivity (Data collection and Transmission)

Effective sensor placement was crucial to minimize the number of sensors needed while ensuring the accuracy and reliability of the data in the IoT sensor network. The strategy for placement focused on key areas:

In urban zones, sensors were strategically placed in high-traffic areas such as Mile 1, Mile 3, Rumuola, and Garrison to monitor changes in pollution levels. Industrial sites like Artillery, Borikiri, and Eleme Junction, known for their high emissions from factories and power plants, were selected for direct monitoring. In rural and agricultural regions, where farming is prevalent, sensors were deployed to track emissions from grazing animals and fertilizers.

Data collection spanned several months, capturing both short-term fluctuations and long-term trends in greenhouse gas emissions. The dataset, consisting of 2,920 entries, was recorded across various locations in Port Harcourt, Rivers State, Nigeria, including Eleme Junction, Rumuola, Borikiri, Mile 1, Choba Junction, Mile 3, Garrison, and Artillery. Each entry includes details on the location, date, and carbon dioxide concentration (in ppm). This extensive dataset allows for a comprehensive analysis and forecasting of CO_2 level variations from different emission sources.

Data Transmission Layer: Processed data is transmitted to the cloud using secure, low-latency protocols like MQTT or HTTP. MQTT, favored in IoT applications, offers low overhead and guaranteed real-time delivery, making it ideal for scenarios requiring continuous monitoring. The ESP32's integrated WiFi module offers powerful data transmission capabilities, making it ideal for various applications such as smart homes, industrial automation, and environmental monitoring. It utilizes the 802.11 b/g/n WiFi standards to transmit data over long distances to cloud servers or local gateways. This feature is particularly advantageous for stationary installations connected to existing WiFi networks, as it reduces the need for additional hardware like cellular modules or wired connections, simplifying setup and lowering overall system costs.

The ESP32 supports high data rates, enabling real-time data streaming and low-latency responses, which are crucial for IoT applications. Its affordability further enhances its appeal, making it a cost-effective solution for many projects. Through its WiFi transmission capabilities, developers can easily add remote monitoring and control functions to embedded devices, with data sent to the cloud for processing and storage. Integration with web services and software is facilitated by available APIs, allowing for advanced features such as data visualization, analysis, and remote diagnostics.

For data transfer, the ESP32 employs MQTT (Message Queuing Telemetry Transport), a lightweight publish-subscribe protocol optimized for IoT environments. MQTT is built on the TCP/IP protocol suite and is designed for scenarios with limited bandwidth. Its broker-based model supports three Quality of Service (QoS) levels, ensuring reliable message delivery even under challenging network conditions. MQTT's efficiency and low overhead make it well-suited for resource-constrained devices. Additionally, its support for Last Will and Testament (LWT) enhances system robustness by notifying other devices if a primary device fails.

This research selected MQTT as the primary protocol due to its suitability for IoT devices, scalability, and robust community support, ensuring effective implementation and integration into IoT systems.

3. Cloud Infrastructure and Data Storage

Once in the cloud, data is stored in Google Cloud Storage (GCS), a scalable, highly available storage solution. GCS replicates data across multiple geographic locations, ensuring durability even in the event of regional failures. This setup guarantees that data remains accessible for subsequent processing and analysis, with layered management and security features. Google Cloud Platform (GCP) offers a comprehensive suite of cloud services, including Google IoT Core, a managed service that enables the secure management and connection of devices globally. The two main components of Google IoT Core are the device manager and the protocol bridge. The device manager handles device registration, authentication, and authorization, while the protocol bridge supports MQTT and HTTP protocols for efficient data communication. Key features of Google IoT Core include:

- Device Management: The Device Manager allows for the registration and management of connected devices, including real-time control of device states and configurations. Devices can send telemetry data to the cloud and receive configuration updates.
- Data Communication Protocols: The service fully supports MQTT and HTTP, with the MQTT bridge offering Quality of Service (QoS) levels 0 and 1 for reliable data transmission.
- Data Processing and Storage: IoT Core integrates with GCP services like Cloud Dataflow, Pub/Sub, Cloud Datastore, BigQuery, and Bigtable, using Cloud Storage for archiving less frequently accessed data.

- Integration and Security: GCP provides extensive integration tools, secure communications through TLS 1.2, per-device JWT authentication, and robust identity and access management controls.
- Performance and Cost: The platform scales efficiently, with costs based on data usage and transaction volume, making it ideal for projects involving large-scale data or complex cloud service integrations.

4. Data Preprocessing and Cleaning
   Before analysis, collected data undergoes preprocessing to enhance quality. This step involves removing noise, correcting errors, and structuring the data uniformly. Preprocessing is critical for improving the accuracy and reliability of subsequent analysis.
5. Data Analysis and Forecasting
   Cleaned data is fed into a machine learning model trained on historical data to forecast future gas concentration levels. The model identifies patterns, detects anomalies, and predicts deviations, enabling proactive responses to potential issues.
6. Web Application Dashboard
   The final layer is a web-based dashboard built with technologies like React or Angular. This dashboard visualizes gas concentration patterns, forecasts, and alerts through graphs, charts, and heatmaps. It allows real-time monitoring and historical data analysis, providing users with customizable alerts and detailed insights into long-term trends. The CO_2 Monitoring and Forecasting Web Application, available at https://co-2-app.replit.app/, is a key component of this IoT system. Designed for researchers and policymakers, it offers open-access, real-time tools for monitoring and forecasting CO_2 levels. The application leverages advanced machine learning to forecast CO_2 concentrations and features a user-friendly interface for easy data interpretation.

   Developed using the Flask framework, the application benefits from Flask's speed, flexibility, and ease of customization. Flask was chosen for its support of essential visualization libraries, its ability to incorporate interactive elements, its lightweight nature, affordability, ease of deployment, and excellent documentation.

   The application includes several key modules. Data ingestion involves collecting real-time and historical CO_2 data from a network of IoT sensors at various locations via APIs. Data cleaning and aggregation are handled by custom Python scripts that process and store the data in a database on the Google Cloud Platform (GCP). The user interface offers intuitive, interactive time series graphs, comparative charts, and forecast overlays, enhancing the overall user experience.

The app interfaces with the IoT network using MQTT and can also receive data from CO_2 sensors through an API. It provides tools for managing the sensor network and processing data for ongoing monitoring and forecasting.

This architecture ensures efficient, scalable monitoring and analysis of greenhouse gas emissions, empowering users with actionable data.

B. Simulation

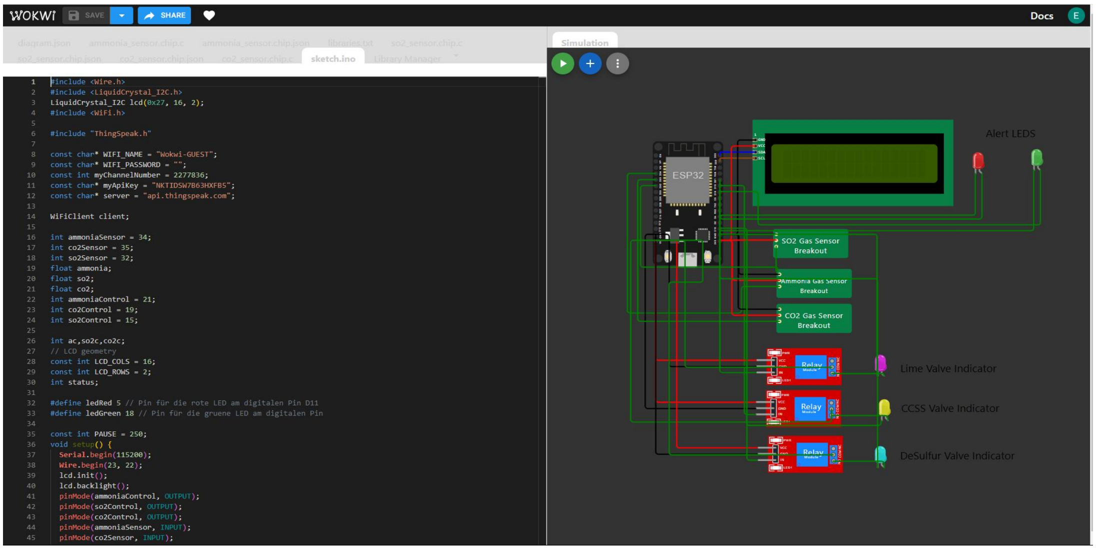
The simulation replicates each part of the study, mirroring the hardware implementation and generating forecast data for various inputs. This IoT system, simulated on the Wokwi platform, tracks and forecasts greenhouse gases using multiple sensors integrated with an ESP32 microcontroller. It showcases real-time data processing and displays results online. Wokwi is a browser-based tool designed for prototyping and testing microcontroller projects with Arduino and ESP32. It provides a vast component library, real-time simulation, and an integrated code editor.

Key features of the Wokwi environment include:

- Extensive Component Library: Access to a wide range of components, from basic LEDs to advanced sensors.

- Real-Time Simulation: Immediate feedback on code and circuit functionality.
- Integrated Code Editor: Improves coding with features like syntax highlighting and error detection.
- Community Collaboration: Facilitates sharing and collaboration on projects, promoting innovation.
- Educational Resources: Offers tutorials and guides for learning electronics and programming.

Wokwi is invaluable for teaching, rapid prototyping, and firmware development, allowing users to test microcontroller interactions without physical hardware. In this research, Wokwi enabled the quick simulation and analysis of greenhouse gas emissions data, providing crucial insights for performance assessment.

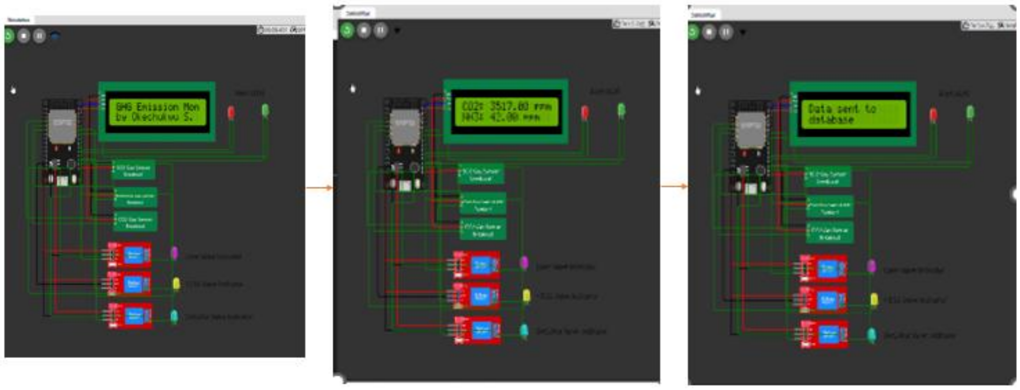


```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
#include <WiFi.h>

#include "ThingSpeak.h"

const char* WIFI_NAME = "Wokwi-GUEST";
const char* WIFI_PASSWORD = "";
const int myChannelNumber = 2277836;
const char* myApiKey = "NKTIDSW7B63HXFB5";
const char* server = "api.thingspeak.com";

WiFiClient client;

int ammoniaSensor = 34;
int co2Sensor = 35;
int so2Sensor = 32;
float ammonia;
float so2;
float co2;
int ammoniaControl = 21;
int co2Control = 19;
int so2Control = 15;

int ac,so2c,co2c;
// LCD geometry
const int LCD_COLS = 16;
const int LCD_ROWS = 2;
int status;

#define ledRed 5 // Pin für die rote LED am digitalen Pin D11
#define ledGreen 18 // Pin für die gruene LED am digitalen Pin

const int PAUSE = 250;
void setup() {
  Serial.begin(115200);
  Wire.begin(23, 22);
  lcd.init();
  lcd.backlight();
  pinMode(ammoniaControl, OUTPUT);
  pinMode(so2Control, OUTPUT);
  pinMode(co2Control, OUTPUT);
  pinMode(ammoniaSensor, INPUT);
  pinMode(co2Sensor, INPUT);
```

Figure 4: Simulation setup interface

IV. RESULTS

- After simulating the system (as shown in Fig 5), it boots up, reads data from the sensors, and automatically sends this data to a cloud-based database.
- The sensors take measurements and send the data to the transmission layer. Every 30 seconds, this data is transmitted to the cloud and saved on a Google Sheet stored on Google Cloud.
- The system operates continuously unless stopped for maintenance. With over 2TB of storage, it can keep reading and saving data for over ten years without reaching capacity.
- This robust system effectively bridges gaps in data storage and transmission, offering a reliable solution for managing large-scale data.
- Data is collected from different setups across distinct locations in Port Harcourt. By gathering data from multiple locations and updating it in real-time, the system addresses gaps in location and spatial coverage.
- The Data Update sheet (shown in Fig 6) stores the collected data, which can then be fetched for further analysis.
- A web-based dashboard (Fig 7) works with the real-time data from the update sheet, automatically applying advanced analytics and machine learning algorithms. This provides live visualizations, forecasts, and insights for each location.
- This system successfully bridges gaps found in existing systems by capturing real-time data and enabling further improvements and analysis.

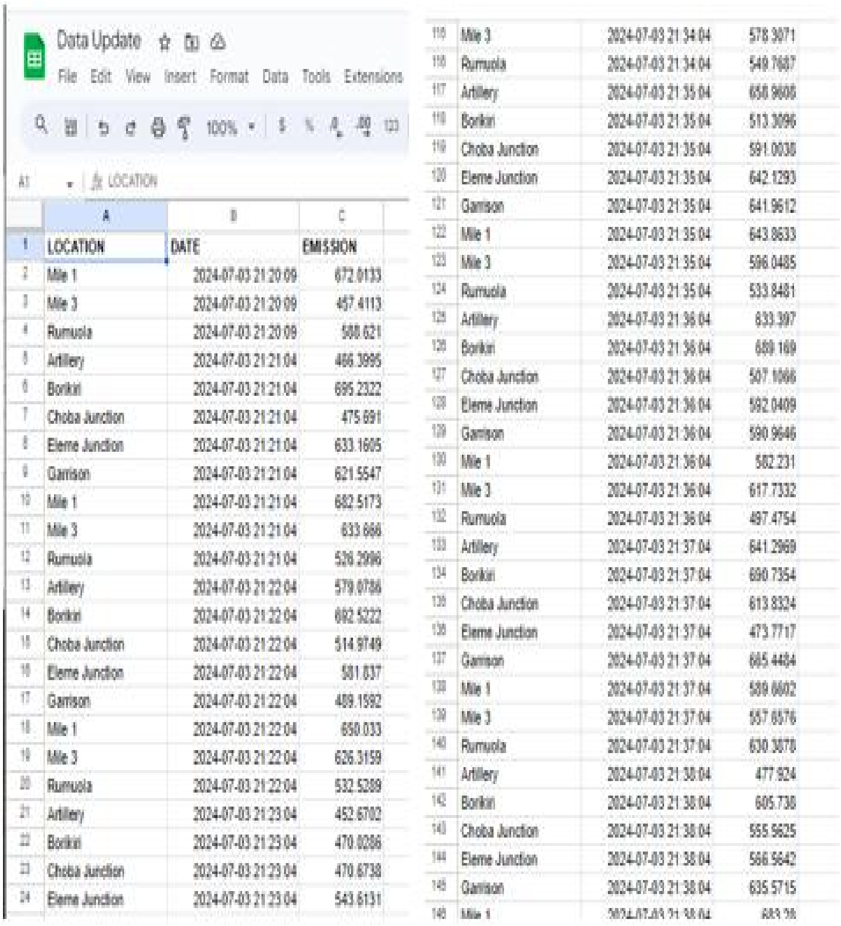


GHG Emission Mon by Okechukwu S.

CO2: 3517.00 PPM NH3: 42.00 PPM

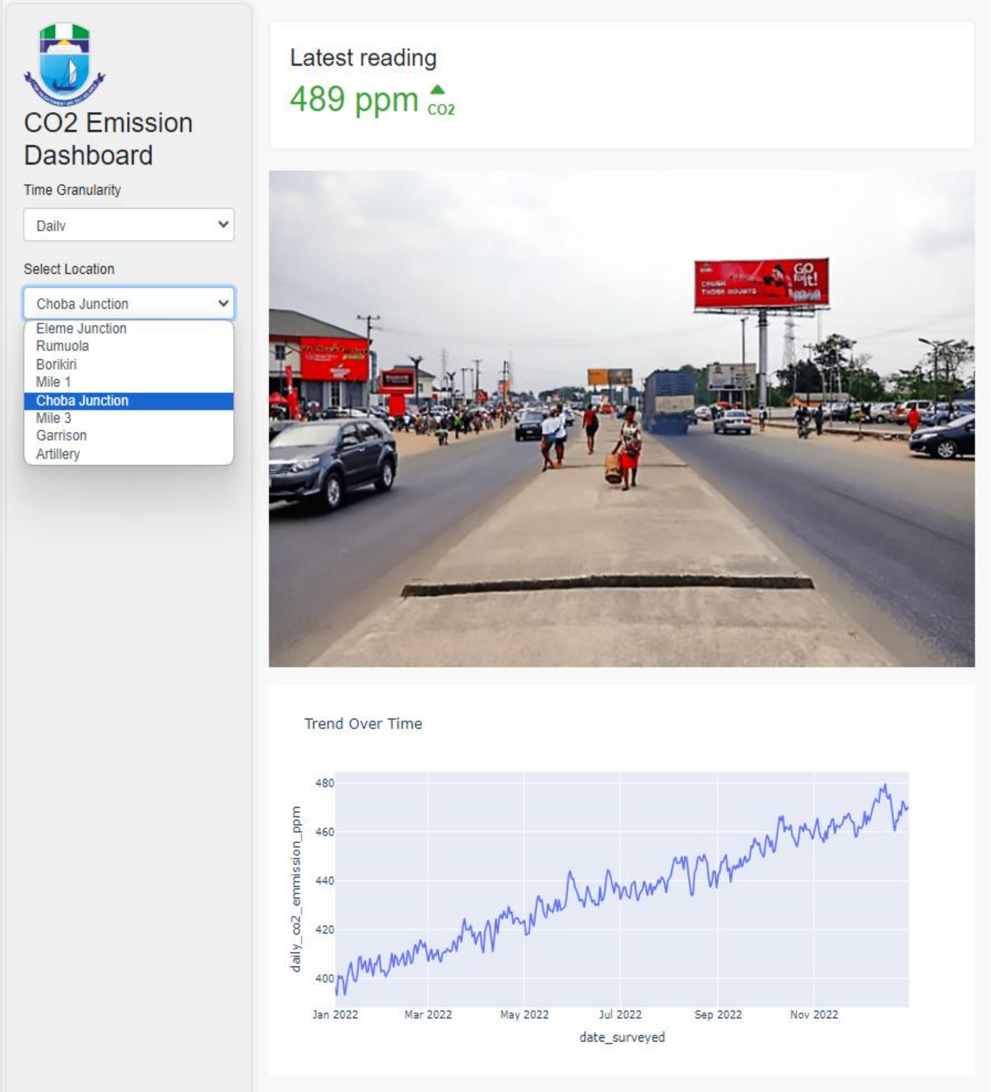
Data sent to database

Figure 5: Simulation Results



	A	B	C
1	LOCATION	DATE	EMISSION
2	Mile 1	2024-07-03 21:20:09	672.0133
3	Mile 3	2024-07-03 21:20:09	457.4113
4	Rumuola	2024-07-03 21:20:09	588.621
5	Artillery	2024-07-03 21:21:04	466.3995
6	Borikiri	2024-07-03 21:21:04	695.2322
7	Choba Junction	2024-07-03 21:21:04	475.691
8	Eleme Junction	2024-07-03 21:21:04	633.1605
9	Garrison	2024-07-03 21:21:04	621.5547
10	Mile 1	2024-07-03 21:21:04	682.5173
11	Mile 3	2024-07-03 21:21:04	633.666
12	Rumuola	2024-07-03 21:21:04	526.2996
13	Artillery	2024-07-03 21:22:04	579.0786
14	Borikiri	2024-07-03 21:22:04	692.5222
15	Choba Junction	2024-07-03 21:22:04	514.9749
16	Eleme Junction	2024-07-03 21:22:04	581.837
17	Garrison	2024-07-03 21:22:04	409.1592
18	Mile 1	2024-07-03 21:22:04	650.033
19	Mile 3	2024-07-03 21:22:04	626.3159
20	Rumuola	2024-07-03 21:22:04	532.5289
21	Artillery	2024-07-03 21:23:04	452.6702
22	Borikiri	2024-07-03 21:23:04	470.0286
23	Choba Junction	2024-07-03 21:23:04	470.6738
24	Eleme Junction	2024-07-03 21:23:04	543.6131
115	Mile 3	2024-07-03 21:34:04	578.3071
116	Rumuola	2024-07-03 21:34:04	549.7687
117	Artillery	2024-07-03 21:35:04	658.9608
118	Borikiri	2024-07-03 21:35:04	513.3096
119	Choba Junction	2024-07-03 21:35:04	591.0038
120	Eleme Junction	2024-07-03 21:35:04	642.1293
121	Garrison	2024-07-03 21:35:04	641.9612
122	Mile 1	2024-07-03 21:35:04	643.8633
123	Mile 3	2024-07-03 21:35:04	596.0485
124	Rumuola	2024-07-03 21:35:04	533.8481
125	Artillery	2024-07-03 21:36:04	633.397
126	Borikiri	2024-07-03 21:36:04	689.169
127	Choba Junction	2024-07-03 21:36:04	507.1066
128	Eleme Junction	2024-07-03 21:36:04	592.0409
129	Garrison	2024-07-03 21:36:04	590.9646
130	Mile 1	2024-07-03 21:36:04	582.231
131	Mile 3	2024-07-03 21:36:04	617.7332
132	Rumuola	2024-07-03 21:36:04	497.4754
133	Artillery	2024-07-03 21:37:04	641.2969
134	Borikiri	2024-07-03 21:37:04	690.7354
135	Choba Junction	2024-07-03 21:37:04	613.8324
136	Eleme Junction	2024-07-03 21:37:04	473.7717
137	Garrison	2024-07-03 21:37:04	665.4484
138	Mile 1	2024-07-03 21:37:04	589.6602
139	Mile 3	2024-07-03 21:37:04	557.6576
140	Rumuola	2024-07-03 21:37:04	630.3878
141	Artillery	2024-07-03 21:38:04	477.924
142	Borikiri	2024-07-03 21:38:04	605.738
143	Choba Junction	2024-07-03 21:38:04	555.9625
144	Eleme Junction	2024-07-03 21:38:04	566.5642
145	Garrison	2024-07-03 21:38:04	635.5715
146	Mile 1	2024-07-03 21:38:04	683.28

Figure 6: Data Update and Storage on Google Cloud



CO2 Emission Dashboard

Time Granularity: Daily

Select Location: Choba Junction

Latest reading

489 ppm CO2

Figure 7: Web-based Dashboard

V. CONCLUSIONS

In this study, we successfully designed and implemented a real-time data capture (RDC) system for monitoring greenhouse gases, addressing the limitations of existing systems. By integrating MQ2 gas sensors, ESP32 microcontrollers, the MQTT communication protocol, and Google Cloud Platform (GCP) for data storage, we developed a scalable and functional solution.

Our system enhances environmental monitoring by collecting real-time data from various locations, which is then analyzed using advanced analytics and machine learning. The user-friendly web-based dashboard allows for easy visualization and interpretation of this data. This research contributes to a better understanding of greenhouse gas emissions and provides a practical tool for mitigating their impact. The system's scalability and adaptability make it suitable for deployment across different environments, including urban areas, industrial sites, and rural landscapes.

Future research could explore integrating additional sensors to monitor a broader range of pollutants, refining forecasting models, and expanding the system's geographical coverage. The insights gained from this research can inform policy decisions, promote sustainable practices, and ultimately contribute to a healthier planet.

REFERENCES

1. Astuti, S. D., Mukhammad, Y., Duli, S. A. J., Putra, A. P., Setiawatie, E. M., & Triyana, K.

(2019). Gas sensor array system properties for detecting bacterial biofilms. *Journal of Medical Signals and Sensors, 9*(3), 158–164. https://doi.org/10.4103/jmss.JMSS_60_18

2. Borhan, M. S., & Khanaum, M. M. (2022). Sensors and methods for measuring greenhouse gas emissions from different components of livestock production facilities. *Journal of Geoscience and Environment Protection, 10*(12), 242–272. https://doi.org/10.4236/gep.2022.1012014
3. Bersani, C., Ruggiero, C., Sacile, R., Soussi, A., & Zero, E. (2022). Internet of Things approaches for monitoring and controlling smart greenhouses in Industry 4.0. *Energies, 15*(10), 3834. https://doi.org/10.3390/en15103834
4. Da Costa, T. P., Da Costa, D. M. B., & Murphy, F. (2024). A systematic review of real-time data monitoring and its potential application to support dynamic life cycle inventories. *Environmental Impact Assessment Review, 105*, 107416. https://doi.org/10.1016/j.eiar.2024.107416
5. Gamess, E., & Hernandez, S. (2022). Performance evaluation of different Raspberry Pi models for a broad spectrum of interests. *International Journal of Advanced Computer Science and Applications, 13*(2). www.ijacsa.thesai.org
6. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010
7. Gupta, D., Bhatt, S., Gupta, M., Kayode, O., & Tosun, A. S. (2020). Access control model for Google Cloud IoT. *Proceedings - 2020 IEEE 6th Intl Conference on Big Data Security on Cloud, BigDataSecurity 2020, 2020 IEEE Intl Conference on High Performance and Smart Computing, HPSC 2020 and 2020 IEEE Intl Conference on Intelligent Data and Security, IDS 2020*, 198–208. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS49724.2020.00044
8. Guzman, J. S., Fuenmayor-Gonzalez, D., Turrel, B., & Dauce, L. (2023, October). Leveraging real-time operational data to reduce greenhouse gas emissions. Paper presented at the ADIPEC, Abu Dhabi, UAE. https://doi.org/10.2118/216258-MS
9. Haidarzhy, V. (2023, November 21). 5 IoT use cases in environmental monitoring. Sirin Software.
10. https://sirinsoftware.com/blog/5-use-cases-of-iot-in-environmental-monitoring
11. International Energy Agency. (2023a). CO2 emissions in 2023. https://iea.blob.core.windows.net/assets/33e2badc-b839-4c18-84cef6387b3c008f/CO2Emissionsin2023.pdf
12. International Energy Agency. (2023b). World energy outlook 2023. https://iea.blob.core.windows.net/assets/86ede39e-4436-42d7-ba2aedf61467e070/WorldEnergyOutlook2023.pdf
13. IPCC. (2022). Global warming of 1.5°C. Cambridge University Press. https://doi.org/10.1017/9781009157940
14. Maier, A., Sharp, A., & Vagapov, Y. (2017). Comparative analysis and practical implementation of the ESP32 microcontroller module for the Internet of Things. *7th IEEE International Conference on Internet Technologies and Applications ITA-17*, Wrexham, UK.
15. Mishra, B., & Kertesz, A. (2020). The use of MQTT in M2M and IoT systems: A survey. *IEEE Access, 8*, 201071–201086. https://doi.org/10.1109/ACCESS.2020.3035849
16. Morsi, I. G., Khedr, M. E., & Aly, A. G. E. (2021). Detection of LPG gas by using multi sensors array and fabricated ZnO gas sensor. *Journal of Physics: Conference Series, 2128*(1). https://doi.org/10.1088/1742-6596/2128/1/012005
17. Narayana, T. L., Venkatesh, C., Kiran, A., J, C. B., Kumar, A., Khan, S. B., Almusharraf, A., & Quasim, M. T. (2024). Advances in real-time smart monitoring of environmental parameters using IoT and sensors. *Heliyon, 10*(7). https://doi.org/10.1016/j.heliyon.2024.e28195
18. Pierleoni, P., Concetti, R., Belli, A., & Palma, L. (2020). Amazon, Google and Microsoft

solutions for IoT: Architectures and a performance comparison. *IEEE Access, 8*, 5455–5470. https://doi.org/10.1109/ACCESS.2019.2961511

19. Saputra, N., Siahaan, M., & Informasi, S. (2021). Sistem dashboard smart energi pada kosan dengan menerapkan IoT berbasis Android. *Pusdansi.org, 1*(1).

20. Statistician, M., Applications, E., Goyal, K., Kumar, R., & Professor, A. (2022). Tinkaercad simulation-based gas detection device using MQ2 sensor. *Mathematical Statistician and Engineering Applications, 71*(4), 13163–13174. http://philstat.org.ph

21. Sudhan, R. H., Kumar, M. G., Prakash, A. U., Devi, S. A. R., & P., S. (2015). Arduino ATmega-328 microcontroller. *IJIREEICE, 3*(4), 27–29. https://doi.org/10.17148/ijireeice.2015.3406

22. Waxman, A. R., Khomaini, A., Leibowicz, B. D., & Olmstead, S. M. (2020). Emissions in the stream: Estimating the greenhouse gas impacts of an oil and gas boom. *Environmental Research Letters, 15*(1). https://doi.org/10.1088/1748-9326/ab5e6f

23. Zhang, H. F., & Kang, W. (2013). Design of the data acquisition system based on STM32. *Procedia Computer Science, 17*, 222–228. https://doi.org/10.1016/j.procs.2013.05.030