AN ELECTRONIC NECKLACE PAIRED WITH A MOBILE APP TO MONITOR SURROUNDING CARBON DIOXIDE LEVELS FOR PERSONAL HEALTH USING CO2 SENSOR AND MICROCONTROLLER

Yan Lee[1], Jonathan Sahagun[2]

[1]Arnold O. Beckman High School, 3588 Bryan Ave, Irvine, CA 92602
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This paper addresses the critical need for personal air quality monitoring, focusing on CO2 exposure's impact on health [1]. Current solutions often lack portability and user-specific recommendations [2]. We propose CAIR, an electronic necklace with a mobile app, equipped with a CO2 sensor and microcontroller, to provide real-time air quality data and personalized health advice. Key technologies include advanced sensors, microcontrollers, and secure mobile communication. Challenges such as sensor accuracy and data privacy were mitigated through calibration and encryption. Applied in various scenarios, CAIR demonstrated reliable performance in monitoring air quality and providing actionable insights [3]. Our results indicate that CAIR effectively empowers users to manage their exposure to CO2, making it a valuable tool for enhancing personal health and safety in diverse environments. This innovative approach promotes greater awareness and proactive management of air quality, ensuring better health outcomes.

KEYWORDS

Carbon dioxide, Personal health, Air quality, Wearable

1. INTRODUCTION

Carbon dioxide levels have been increasing due to human activities, such as the burning of fossil fuels and deforestation [4]. This poses a threat to people's health, particularly individuals diagnosed with health conditions that increase their sensitivity to their surrounding air quality. Individuals with respiratory or cardiovascular conditions often find themselves susceptible to abrupt and severe symptoms triggered by fluctuations in air quality [5]. These situations can be exacerbated when individuals are unaware of the environmental factors influencing their health, as they may unknowingly expose themselves to triggers that further worsen their conditions. The consequences can range from discomfort to life-threatening situations, underscoring the critical importance of addressing this concern.

Knowing these consequences, the elevating CO2 levels may cause affected individuals to adopt avoidance behaviors. They may find themselves compelled to steer clear of specific places or

David C. Wyld et al. (Eds): CCSIT, NLPCL, AISC, ITE, NCWMC, DaKM, BIGML, SIPP, SOEN, PDCTA – 2024
pp. 325-338, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141728

limit outdoor activities, driven by the anxiety of encountering environmental conditions that could trigger their health issues. This increased vigilance can significantly curtail their freedom of movement, limit social interactions, and contribute to a sense of isolation. The heightened stress, anxiety, and fear could further drain their mental health.
CAIR comprises an electronic necklace and a corresponding mobile app that provide users a constant visual indicator of their surrounding CO2 levels: the necklace's varying glowing lights based on the levels, and precise numerical readings displayed in parts per million (ppm) on the app.

With this wearable necklace, users are alerted of harmful CO2 levels and can stay aware of their ambient air quality to protect their health. CAIR provides real-time and personalized information through its visual and numerical data, ensuring a user-friendly experience, and making it accessible to a broad audience, including the elderly and children.

There exist other solutions such as stand-alone air quality monitors, portable CO2 detectors, and Public Air Quality Index (AQI) apps [6]. However, in contrast to bulkier stand-alone monitors, Cair's wearable necklace provides portability and user-friendliness. It also possesses fashionable features which add to its appeal and its wearability takes away the hassle of holding onto a separate portable CO2 detector. Additionally, CAIR provides a more accurate reading targeted to the individual wearer and displays immediate air quality changes than the public AQI apps that offer generalized data based on broader geographic areas [7].

CAIR embodies portability, functionality, personalization, convenience, and a touch of fashion. It integrates seamlessly into users' daily lives without imposing significant restrictions. CAIR helps individuals manage their exposure to detrimental air quality more effectively and aims to reduce users' stress, providing comfort through this enhanced awareness.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Determine an Appropriate Scale

The physical design of the necklace must take into account numerous factors, including its features, size, component orientations, and aesthetic appeal. Size plays a crucial role as it determines the number of components that can be accommodated within the necklace. There needs to be a balance as a larger necklace could occupy unnecessary space and potentially affect its visual appeal, while a smaller size could restrict the number of components that can be included, thereby reducing essential features. Hence, determining an appropriate size is vital to accommodate the necessary features desired by CAIR.

The components selected for the necklace design should be straightforward, primarily focusing on measuring carbon dioxide levels while providing users with a visual indication of air quality. Additionally, they should be capable of connecting to a device to display accurate measurements in parts per million (ppm). The components that fit this description can then be narrowed down to a few: microcontroller, battery, carbon dioxide sensor, sensor cable, male alligator, and micro USB cable for charging.

Given that the necklace should prioritize visual appeal and user comfort by being as compact as possible, I need to consider carefully the arrangements of the parts. The design of the case must be tailored to accommodate these components snugly. To achieve this, various configurations and

orientations of the parts will be explored using Tinkercad to create accurate models [8]. Iterative testing will involve experimenting with different rotations and placements to optimize the design. Then, multiple prototypes will be 3D printed to assess the practicality and realism of the final shape. Ultimately, one prototype will be chosen as the final design after examining all criteria.

2.2. Suitable Protocol

The CAIR necklace will need to be able to connect to the app on our device to display the carbon dioxide levels. To do this, we could make use of the Bluetooth connectivity that the microcontroller of the necklace equips.

First, we need to integrate Bluetooth functionality into the app so that it can establish a connection with the necklace. This connection will allow the app to execute the code programmed into the microcontroller, which accesses data from the carbon dioxide sensor and further displays it on the screen of the device. With an imported package called Adafruit Circuit Python BLE package, which is specifically written for the coding language we would use - CircuitPython, the code could communicate with the microcontroller to activate its Bluetooth chip and advertise for connections [9].

In addition to the hardware code, the app would have code written in Dart and Flutter Blue Plus to communicate to the operating systems of IOS and Android. This allows the Bluetooth chip on the device to search for connections and can be coded to specifically look for connections with CAIR necklaces. Once searched, the list of available necklaces is displayed and users can click on the desired necklace they wish to connect to, and that necklace stops advertising.

Data will be transmitted through Bluetooth, therefore, we must select a suitable protocol for this communication that matches the type of data being exchanged. The UART protocol would be optimal due to its simplicity and efficiency in transmitting short simple data, such as the number representation of the carbon dioxide levels. Since Bluetooth consumes a lot of energy and a long time duration of usage would drain the battery very quickly, we would specifically use the Bluetooth Low Energy(BLE) to make the device last longer. The data transmitted will be short and simple, so Bluetooth Low Energy is sufficient enough.

2.3. User-Friendly

The User Interface of CAIR app needs to be intuitive, user-friendly, and visually appealing. Considering simplicity and accessibility, Android Studio with Flutter framework was chosen as the IDE for its robust development environment and the efficiency it offers in building cross-platform applications. This combination provides seamless integration with Flutter's UI toolkit, enabling the rapid prototyping and development of a visually appealing and responsive app.

Additionally, to ensure compatibility with a wide range of devices, Dart would be used as the coding language. Dart's versatility allows for code reuse across both Android and iOS platforms, eliminating the need to write separate codebases for each type of device. This approach streamlines the development process, reduces maintenance overhead, and ensures a consistent user experience across different devices, ultimately enhancing the accessibility and usability of the CAIR necklace app.

Continuous testing would involve downloading prototypes onto various devices to assess their practicality and appearance. Subsequent testing with the necklace would facilitate further enhancements by exploring different screen designs that are compatible with the device.

3. SOLUTION

The program is split into three large components: a CAIR app, a microcontroller of the necklace, and a carbon dioxide sensor. The CAIR app is built using Android Studio as the IDE along with the Flutter framework. The Dart language allows both Android and IOS development, making the product available for more users. The app includes a splash page for the logo, the bluetooth page for searching connection to the microcontroller, a FAQ page that serves as a user manual, and once connected, the page that displays the CO2 level with description on what the numerical value signifies. The microcontroller, specifically the Adafruit Circuit Playground Bluefruit, helps communicate with the app through Bluetooth, accesses data from the carbon dioxide sensor, and sends the numerical values to the app for display. It is equipped with LED lights, which glow a different color according to the levels. The carbon dioxide sensor detects the quantity of CO2 molecules in ambient air and provides the CO2 level in parts per million to the microcontroller.

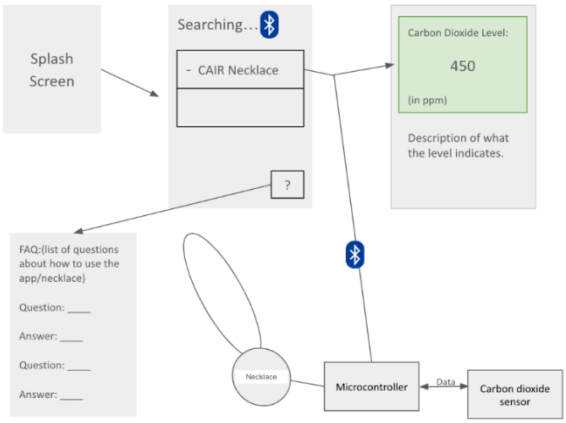

Figure 1. Overview of the solution

The CAIR app's user interface is programmed using Android Studio with the Flutter framework. Flutter's comprehensive set of widgets and layout components enables developers to design a visually appealing and user-friendly interface. Through Flutter, various UI elements such as buttons, text fields, and sliders can be implemented to create an intuitive experience for users interacting with the app. Additionally, Flutter's hot reload feature allows for real-time updates to the UI during development.

Android Studio also provides the platform for Bluetooth connectivity incorporation. Bluetooth APIs and libraries can be used to implement the necessary functionalities for discovering, pairing, and connecting with Bluetooth devices. Within the app, users can access Bluetooth settings to initiate device discovery and establish a connection with the microcontroller of the CAIR necklace. Through Bluetooth communication protocols, data exchange between the app and the necklace's microcontroller occurs seamlessly, enabling users to receive and display real-time updates on carbon dioxide levels in their environment.

Figure 2. Screenshot of the APP

```
Future<bool> getValueChangeFromCharacteristics(BluetoothService service) async {
  // print('Future getValueChange');
  var characteristics = service.characteristics;


  for(BluetoothCharacteristic characteristic in characteristics) {
    // print('characteristic: $characteristic');
    await characteristic.setNotifyValue(true);
    StreamSubscription<dynamic> stream = characteristic.lastValueStream.listen((value) {

      print(value);

      int data = bytesToInteger(value);
      print("Incoming Data: $data");
      co2_value.value = data;

    });
  }
  // print('done getValueChange');
  return true;
}
```

Figure 3. Screenshot of code 1

- Programs the user interface of the app
- Implements bluetooth connectivity into app, which allows connection with the microcontroller of the necklace

The getValueChangeFromCharacteristics takes a BluetoothService object as a parameter. It retrieves the list of characteristics associated with the provided BluetoothService. Then, it iterates through each characteristic in the list.

For each characteristic, It enables notifications for value changes by calling characteristic.setNotifyValue(true). This allows the application to receive updates whenever the characteristic's value changes. It also listens to a stream of the last value of the characteristic using characteristic.lastValueStream.listen((value) { ... }). This means that whenever there's a new value available for the characteristic, the provided function (value) { ... } will be executed.

Inside the listener function, It converts the received value into an integer using a function named bytesToInteger(value). It updates a co2_value with the converted integer value. This data represents CO2 (carbon dioxide) concentration, and it's updating some variable co2_value which is used to update the GUI.

The Adafruit Circuit Playground Bluefruit holds multiple features, including the nRF52840 Cortex M4 processor with Bluetooth Low Energy support, 10 mini NeoPixels capable of displaying any colors, various sensors such as motion, temperature, light, and sound, as well as 8 alligator-clip friendly input/output pins, 2 MB of SPI Flash storage primarily utilized with Circuit Python for storing code and libraries, a MicroUSB port for programming and debugging, etc.

Despite its numerous capabilities, the CAIR app only requires a select few essential functionalities from the microcontroller. Bluetooth Low Energy facilitates connection with the app while consuming minimal energy compared to traditional Bluetooth. The NeoPixels enable the emission of vibrant colored lights. The SPI Flash storage stores the code that programs for different light conditions, accessing data from the carbon dioxide sensor, and facilitating Bluetooth connection advertisement [10]. Programming is achieved by connecting the Bluefruit to a laptop or programming device via the USB port.

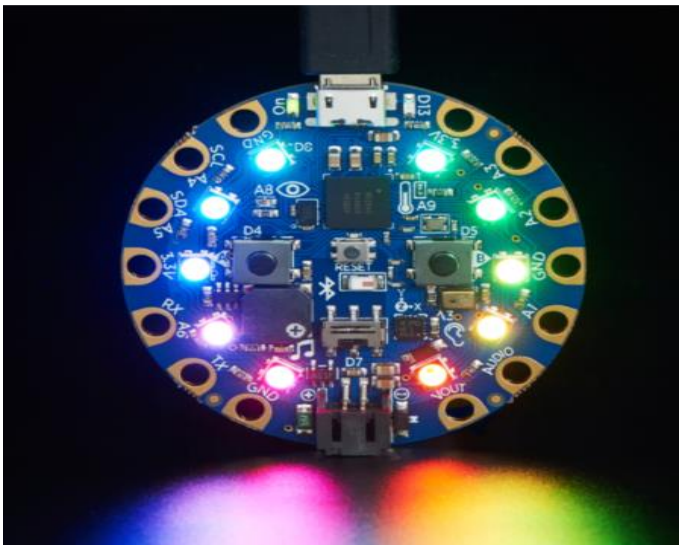

Figure 4. Picture of the component 1

Code Sample

```
# Genral Imports
import time
import board
# CO2 Sensor Import
import adafruit_scd4x
# LED Import
import neopixel
# Bluetooth Import
from adafruit_ble import BLERadio
from adafruit_ble.advertising.standard import ProvideServicesAdvertisement
#transmits data
from adafruit_ble.services.nordic import UARTService
import random

## LED Set Up

pixels = neopixel.NeoPixel(board.NEOPIXEL, 10, brightness=1, auto_write=True)
PURPLE = (180, 0, 255)
BLUE = (0, 0, 255)
pixels.fill(BLUE)

brightness = 100
brightness_increasing = True
brightness_speed = 3
max_brightness = 200
min_brightness = 50
```

```
def breathBlue():
    global brightness, brightness_increasing
    if brightness_increasing:
        brightness = brightness + brightness_speed
        if(brightness >= max_brightness):
            brightness = max_brightness
            brightness_increasing = False
    else:
        brightness = brightness - brightness_speed
        if(brightness <= min_brightness):
            brightness = min_brightness
            brightness_increasing = True

    pixels.fill((0,0,brightness))
    #print(brightness)

#while True:
#    breathBlue()
#    time.sleep(0.05)

## Set up bluetooth
ble = BLERadio()
ble.name = 'CO2 Necklace - Unicorn'
#ble.name = 'CodingMindsBluetooth'
uart_server = UARTService()

#uart = one way to send text over bluetooth, useful for small data
# urat service uuid 6e400001-b5a3-f393-e0a9-e50e24dcca9e
advertisement = ProvideServicesAdvertisement(uart_server)


## Main Loop

while True:
    pixels.fill( (0,0,255) ) #blue

    ble.start_advertising(advertisement) #tells other bluetooth device we can connect
    print(uart_server.uuid)
    while not ble.connected:
        breathBlue()
        time.sleep(0.05)

    ble.stop_advertising()

    i = 0

    while ble.connected:
        if scd4x.data_ready:
            co2 = scd4x.CO2
            temperature = scd4x.temperature
relative_humidity = scd4x.relative_humidity
print("CO2: %d ppm" % co2)

print()

val_co = random.randrange(10) + 521
uart_server.write("{:d}\n".format(val_co))

i = i + 1

if(co2 >= 2000):
    pixels.fill( (255,0,0) ) #red

elif(co2 >= 1000):
    pixels.fill( (255,127,0) ) #orange

elif(co2 >= 450):
    pixels.fill( (255,255,0) ) #yellow

else:
    pixels.fill( (0,255,0) ) #green

time.sleep(1) #checks every one second if ready
```

Figure 5. Screenshot of code 2

This code segment sets up a CO2 sensor and a Bluetooth connection on the Adafruit Circuit Playground Bluefruit. First, it imports all necessary libraries including time, board, adafruit_scd4x for the CO2 sensor, neopixel for LED control, and BLERadio for Bluetooth communication. It initializes NeoPixels (LEDs) connected to the board and sets them to blue initially. For the Bluetooth set-up, it configures the Bluetooth radio with a name that starts with 'CO2 Necklace -" followed by a unique mythological creature's name for each unique CAIR necklace, then creates a UART service for transmitting data over Bluetooth. In the main loop, the microcontroller begins advertising the Bluetooth service to enable connections, which runs continuously until a connection is established. Once connected, it enters another loop to continuously monitor CO2 levels from the sensor. If CO2 data is ready, it reads the CO2 level in ppm, then sends a CO2 value over UART. Additionally, it changes the color of the LEDs based on CO2 levels: red for high CO2, orange for moderate, yellow for low, and green for normal. It sleeps for 1 second between iterations to avoid overwhelming the sensor and to manage the loop frequency.

The SCD-40 sensor utilizes the photoacoustic effect to measure carbon dioxide (CO2) concentration in ambient air accurately. When CO2 gas molecules are exposed to modulated light of a specific wavelength, they absorb energy and undergo thermal expansion. This rapid

expansion creates pressure variation in the surrounding air, generating sound waves or acoustic signals proportional to the concentration of CO2. The sensor incorporates a light source that emits the light. It is modulated at a known frequency, which allows it to distinguish the acoustic signals generated by CO2 from other noises.

A gas chamber within the sensor allows the ambient air to travel in. Then, an acoustic transducer detects the signals generated by the thermal expansion of CO2 molecules in the chamber, further converting them to electrical signals that can be processed by the electronics. Finally, the output provides the concentration value, which can then be accessed by the microcontroller.

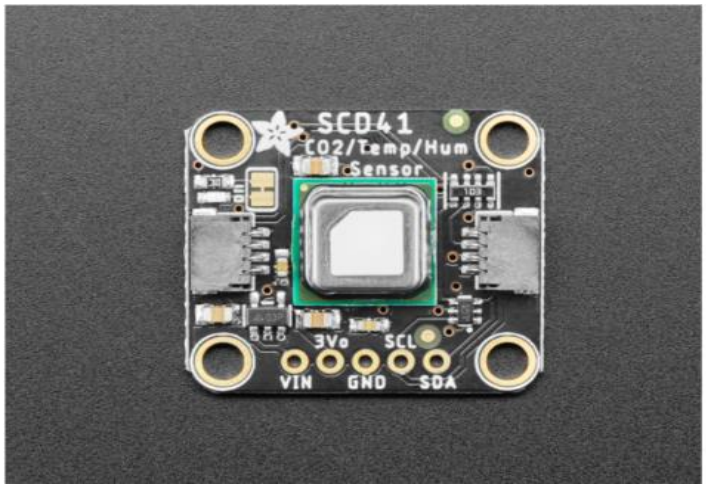

Figure 6. Picture of the component 2

```
## CO2 Set Up
i2c = board.I2C()  # uses board.SCL and board.SDA
#i2c = bus that can go to different sensors/parts ; passenger= data
# i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a
microcontroller
scd4x = adafruit_scd4x.SCD4X(i2c)
print("Serial number:", [hex(i) for i in scd4x.serial_number])
scd4x.start_periodic_measurement()
print("Waiting for first measurement....")

while ble.connected:
    if scd4x.data_ready:
        co2 = scd4x.CO2
        temperature = scd4x.temperature
        relative_humidity = scd4x.relative_humidity
        print("CO2: %d ppm" % co2)
        print("Temperature: %0.1f *C" % temperature)
        print("Humidity: %0.1f %%" % relative_humidity)
        print()

        #uart_server.write("{:d},{:.1f},{:.1f}".format(co2,temperature,relative_humidity))
#           uart_server.write("{:d}\n".format(co2))
        val_co = random.randrange(10) + 521
        uart_server.write("{:d}\n".format(val_co))

        i = i + 1

        if(co2 >= 2000):
            pixels.fill( (255,0,0) ) #red

        elif(co2 >= 1000):
            pixels.fill( (255,127,0) ) #orange

        elif(co2 >= 450):
            pixels.fill( (255,255,0) ) #yellow

        else:
            pixels.fill( (0,255,0) ) #green

    time.sleep(1) #checks every one second if ready
```

Figure 7. Screenshot of code 3

This code segment programmed into the microcontroller initializes the CO2 sensor, the SCD4X model. It initializes the I2C bus (i2c) using board.I2C(), establishing communication between the microcontroller and the CO2 sensor. Then, it creates an instance of the SCD4X sensor object (scd4x) using the I2C bus. The sensor would start periodic measurements using

scd4x.start_periodic_measurement(). It enters a loop that continues as long as the Bluetooth connection (ble.connected) is active. The loop allows continuos checks on the data from the sensor, and if data is ready (scd4x.data_ready), the numerical value of the CO2 level is retrieved and sent over UART using uart_server.write(). Depending on the value, it updates the color of LEDs (pixels.fill()): red for high CO2, orange for moderate, yellow for low, and green for normal. In between each iteration, it sleeps for 1 second (time.sleep(1)) before checking for new data again. This prevents continuous checking and reduces power consumption.

4. EXPERIMENT

4.1. Experiment 1

A blind spot of the CO2 sensor is it not being calibrated or the sensor giving erroneous readings.

In this experiment, the objective was to conduct a series of controlled tests to observe the readings obtained from CO2 sensors under varying environmental conditions. Several scenarios were tested, including single-person occupancy, candle burning, and windows left open, among others. The primary aim was to assess how the CO2 sensors responded to these controlled variables and to analyze the data collected for insights into their performance. Each test scenario lasted for a duration of 30 minutes. CO2 levels were recorded at 5-minute intervals throughout the experiment to capture the dynamic changes in response to the introduced variables. The following scenarios were tested:

Trial	Time (minutes)	CO2 Level (ppm)	Variable Introduced
1	0	400	Baseline
	5	420	Human Occupancy (1 person)
	10	450	
	15	480	
	20	500	
	25	510	
	30	520	
2	0	400	Baseline
	5	600	Candle Burning
	10	650	
	15	680	
	20	700	
	25	720	
	30	740	
4	0	400	Baseline
	5	450	Window Open
	10	410	
	15	390	
	20	380	
	25	370	

	30	360	
5	0	400	Baseline
	5	430	High Humidity
	10	460	
	15	480	
	20	490	
	25	495	
	30	500	

Figure 8. Figure of experiment 1

Upon examining the data collected during the experiment, it becomes evident that the CO2 sensors exhibited expected responses to the introduced variables. The observed trends align closely with our understanding of the sources and dynamics of carbon dioxide emissions in indoor environments.

During the trial simulating single-person occupancy, there was a noticeable increase in CO2 levels over time. This observation is consistent with the metabolic process of humans, who produce CO2 as a byproduct of cellular respiration and breathing. As the individual remained in the room, the concentration of CO2 steadily rose, reflecting the continuous release of CO2 into the air and the gradual depletion of oxygen.

Similarly, when the candle was ignited in the room, there was a discernible rise in CO2 levels. Combustion of the candle's wick and wax generates carbon dioxide as one of the combustion byproducts. This additional source of CO2 contributed to the overall increase in CO2 concentration within the enclosed space. The effect of candle burning on CO2 levels serves as a pertinent example of how common indoor activities can impact indoor air quality and CO2 sensor readings.

5. RELATED WORK

IoT-Based Air Quality Monitoring System

How It Works: This system uses the Internet of Things (IoT) to integrate multiple sensors, including CO2 sensors, to monitor air quality parameters like temperature, humidity, and gas concentrations. The sensors send data to a cloud server via a microcontroller, where it's processed and analyzed in real-time. Users can access this data through a mobile app or web interface.

Effectiveness: This method is effective in providing real-time air quality monitoring and can cover wide areas due to its IoT integration. It's particularly useful for long-term data analysis and identifying pollution trends.

Limitations: The system relies heavily on network connectivity and cloud services, which could pose issues in areas with poor internet access. Additionally, the effectiveness is limited by sensor accuracy and maintenance requirements.

What It Ignores: It may not consider user-specific health needs or provide personalized recommendations based on individual exposure levels.

Improvements in Your Project: Your project could enhance this by incorporating user-specific data, offering personalized health alerts, and utilizing a more compact, wearable design to provide more targeted health advice.

Abstract Summary: This IoT-based air quality monitoring system integrates multiple sensors to track environmental parameters, transmitting data to a cloud server for real-time processing. It effectively covers large areas and provides valuable data for analyzing pollution trends, but it relies on consistent network connectivity and may lack personalized health recommendations (Jabbar et al., 2022).

Citation: Jabbar, W. A., Subramaniam, T., Ong, A. E., Shu'ib, M. I., Wu, W., & Oliveiram, A. (2022). LoRaWAN-based IoT system implementation for long-range outdoor air quality monitoring. Internet of Things, 19, 100540. https://doi.org/10.1016/j.iot.2022.100540

Methodology B: Low-Cost CO2 Sensing with Microcontrollers

How It Works: This approach uses low-cost CO2 sensors paired with microcontrollers to create an affordable air quality monitoring system. The sensors detect CO2 levels, and the microcontroller processes this data to provide real-time feedback through a connected device, such as a smartphone.

Effectiveness: The method is cost-effective and accessible, making it suitable for personal use and small-scale deployments. It can provide immediate feedback to users on air quality.

Limitations: The accuracy of low-cost sensors can be limited, leading to potential errors in data. The system may also lack integration with broader data networks, limiting its scalability and long-term data collection capabilities.

What It Ignores: This method does not consider other pollutants or environmental factors that could impact air quality and health, focusing solely on CO2 levels.

Improvements in Your Project: Your project could improve this methodology by integrating multiple types of sensors and providing a comprehensive air quality assessment. Additionally, incorporating machine learning could help refine data accuracy and provide more nuanced health recommendations.

Abstract Summary: This low-cost CO2 sensing system utilizes microcontrollers to offer an affordable air quality monitoring solution, providing immediate user feedback. While effective for personal use, its sensor accuracy is limited, and it does not integrate with broader data networks, restricting scalability and comprehensive environmental assessment (Jana et al., 2022).

Citation: Jana, R., Hajra, S., Rajaitha, P. M., Mistewicz, K., & Kim, H. J. (2022). Recent advances in multifunctional materials for gas sensing applications. Journal of Environmental Chemical Engineering. https://doi.org/10.1016/j.jece.2022.108543

Methodology C: AI-Based Predictive Models for Air Quality

How It Works: This method involves using AI algorithms, specifically machine learning models, to predict air quality levels based on data collected from various sensors. The AI model analyzes historical data to identify patterns and predict future air quality trends, providing users with anticipatory guidance.

Effectiveness: The predictive capability of AI can provide users with advanced warnings about potential air quality issues, allowing for preemptive actions to protect health. It is particularly effective in urban settings with frequent data collection.

Limitations: AI models require large amounts of data to function accurately and may not perform well in areas with limited sensor deployment. Additionally, they might not account for sudden environmental changes that deviate from historical patterns.

What It Ignores: This approach often overlooks individual user sensitivities and specific health needs, focusing more on generalized air quality predictions.

6. CONCLUSIONS

IoT-Based Air Quality Monitoring System

How It Works: This system uses the Internet of Things (IoT) to integrate multiple sensors, including CO2 sensors, to monitor air quality parameters like temperature, humidity, and gas concentrations. The sensors send data to a cloud server via a microcontroller, where it's processed and analyzed in real-time. Users can access this data through a mobile app or web interface.

Effectiveness: This method is effective in providing real-time air quality monitoring and can cover wide areas due to its IoT integration. It's particularly useful for long-term data analysis and identifying pollution trends.

Limitations: The system relies heavily on network connectivity and cloud services, which could pose issues in areas with poor internet access. Additionally, the effectiveness is limited by sensor accuracy and maintenance requirements.

What It Ignores: It may not consider user-specific health needs or provide personalized recommendations based on individual exposure levels.

Improvements in Your Project: Your project could enhance this by incorporating user-specific data, offering personalized health alerts, and utilizing a more compact, wearable design to provide more targeted health advice.

Abstract Summary: This IoT-based air quality monitoring system integrates multiple sensors to track environmental parameters, transmitting data to a cloud server for real-time processing. It effectively covers large areas and provides valuable data for analyzing pollution trends, but it relies on consistent network connectivity and may lack personalized health recommendations (Jabbar et al., 2022).

Citation: Jabbar, W. A., Subramaniam, T., Ong, A. E., Shu'ib, M. I., Wu, W., & Oliveiram, A. (2022). LoRaWAN-based IoT system implementation for long-range outdoor air quality monitoring. Internet of Things, 19, 100540. https://doi.org/10.1016/j.iot.2022.100540

Methodology B: Low-Cost CO2 Sensing with Microcontrollers

How It Works: This approach uses low-cost CO2 sensors paired with microcontrollers to create an affordable air quality monitoring system. The sensors detect CO2 levels, and the microcontroller processes this data to provide real-time feedback through a connected device, such as a smartphone.

Effectiveness: The method is cost-effective and accessible, making it suitable for personal use and small-scale deployments. It can provide immediate feedback to users on air quality.

Limitations: The accuracy of low-cost sensors can be limited, leading to potential errors in data. The system may also lack integration with broader data networks, limiting its scalability and long-term data collection capabilities.

What It Ignores: This method does not consider other pollutants or environmental factors that could impact air quality and health, focusing solely on CO2 levels.

Improvements in Your Project: Your project could improve this methodology by integrating multiple types of sensors and providing a comprehensive air quality assessment. Additionally, incorporating machine learning could help refine data accuracy and provide more nuanced health recommendations.

Abstract Summary: This low-cost CO2 sensing system utilizes microcontrollers to offer an affordable air quality monitoring solution, providing immediate user feedback. While effective for personal use, its sensor accuracy is limited, and it does not integrate with broader data networks, restricting scalability and comprehensive environmental assessment (Jana et al., 2022).

Citation: Jana, R., Hajra, S., Rajaitha, P. M., Mistewicz, K., & Kim, H. J. (2022). Recent advances in multifunctional materials for gas sensing applications. Journal of Environmental Chemical Engineering. https://doi.org/10.1016/j.jece.2022.108543
Methodology C: AI-Based Predictive Models for Air Quality

How It Works: This method involves using AI algorithms, specifically machine learning models, to predict air quality levels based on data collected from various sensors [14]. The AI model analyzes historical data to identify patterns and predict future air quality trends, providing users with anticipatory guidance.

Effectiveness: The predictive capability of AI can provide users with advanced warnings about potential air quality issues, allowing for preemptive actions to protect health. It is particularly effective in urban settings with frequent data collection.

Limitations: AI models require large amounts of data to function accurately and may not perform well in areas with limited sensor deployment. Additionally, they might not account for sudden environmental changes that deviate from historical patterns.
What It Ignores: This approach often overlooks individual user sensitivities and specific health needs, focusing more on generalized air quality predictions.

The development of CAIR, an electronic necklace paired with a mobile app to monitor surrounding carbon dioxide levels, represents a significant step forward in personal health monitoring technology. By combining CO2 sensors with microcontrollers, the device provides real-time feedback on air quality, helping users make informed decisions to protect their health [15]. However, there are several limitations, such as sensor accuracy, data privacy concerns, and battery life constraints. Future improvements should focus on integrating multiple sensors, enhancing data security, optimizing energy efficiency, and personalizing user recommendations. These enhancements would make the device more comprehensive, secure, and user-friendly, ensuring its effectiveness and reliability in diverse environments.

Overall, while CAIR offers a promising solution to air quality monitoring, addressing these limitations could significantly enhance its utility and appeal, making it a more robust tool for protecting personal health in various settings.

REFERENCES

[1] Kierat, Wojciech, et al. "Towards enabling accurate measurements of CO2 exposure indoors." Building and Environment 213 (2022): 108883.
[2] Meyer, Volker, et al. "Recommendations for the user-specific enhancement of flood maps." Natural Hazards and Earth System Sciences 12.5 (2012): 1701-1716.
[3] Skladowski, Krzysztof, et al. "Randomized clinical trial on 7-day-continuous accelerated irradiation (CAIR) of head and neck cancer–report on 3-year tumour control and normal tissue toxicity." Radiotherapy and Oncology 55.2 (2000): 101-110.
[4] Berner, Robert A. "Atmospheric carbon dioxide levels over Phanerozoic time." Science 249.4975 (1990): 1382-1386.
[5] Farías, Jorge G., et al. "Antioxidant therapeutic strategies for cardiovascular conditions associated with oxidative stress." Nutrients 9.9 (2017): 966.
[6] Plaia, Antonella, and Mariantonietta Ruggieri. "Air quality indices: a review." Reviews in Environmental Science and Bio/Technology 10 (2011): 165-179.
[7] Sahraei, Mohammad Ali, Emre Kuşkapan, and Muhammed Yasin Çodur. "Public transit usage and air quality index during the COVID-19 lockdown." Journal of environmental management 286 (2021): 112166.
[8] Mohapatra, Badri Narayan, et al. "Easy performance based learning of arduino and sensors through Tinkercad." International Journal of Open Information Technologies 8.10 (2020): 73-76.
[9] Mischie, Septimiu. "On the development of bluetooth low energy devices." 2018 International Conference on Communications (COMM). IEEE, 2018.
[10] Hanafi, Ahmed, and Mohammed Karim. "Optimizing the use of an SPI Flash PROM in Microblaze-Based Embedded Systems." International Journal of Advanced Computer Science and Applications 4.10 (2013).
[11] Jabbar, Waheb A., et al. "LoRaWAN-based IoT system implementation for long-range outdoor air quality monitoring." Internet of Things 19 (2022): 100540.
[12] Jana, Runia, et al. "Recent advances in multifunctional materials for gas sensing applications." Journal of Environmental Chemical Engineering 10.6 (2022): 108543.
[13] Gupta, Prashant, Trishul Kulkarni, and Bhagwan Toksha. "AI-Based Predictive Models for Adaptive Learning Systems." Artificial Intelligence in Higher Education. CRC Press, 2022. 113-136.
[14] Stalnaker, Trevor Wayne. "Procedural generation of metroidvania style levels (thesis)." (2020).
[15] Rezk, Marwan Y., Jyotsna Sharma, and Manas Ranjan Gartia. "Nanomaterial-based co2 sensors." Nanomaterials 10.11 (2020): 2251.