Proceedings of the Creative Construction Conference (2023)
Edited by: Miroslaw J. Skibniewski & Miklos Hajdu
https://doi.org/10.3311/CCC2023-007
Keszthely, Hungary
20-23 June 2023

DEVELOPING A DIGITAL TWIN ON A UNIVERSITY CAMPUS TO SUPPORT EFFICIENT AND SUSTAINABLE BUILDINGS

Borja García de Soto[1], Zulema Sanchis[2], Ali Ezzeddine[2], Eyob Mengiste[1], Manuel Padilla[1], Sakrit Karmacharya[1]

1 S.M.A.R.T. Construction Research Group, Division of Engineering, New York University Abu Dhabi (NYUAD), Experimental Research Building, Saadiyat Island, P.O. Box 129188, Abu Dhabi, United Arab Emirates

2 Accuracy, Dubai International Financial Centre (DIFC), Dubai, United Arab Emirates

Abstract

This study presents a case study for the proof of concept of developing and integrating a digital twin for an open floor-plan office space in a university campus to assess the (near) real-time monitoring of different elements, including temperature, humidity and occupancy. The system uses various weather sensors and a camera feed as input to a computer vision algorithm to detect (near) real-time occupancy. The goal is to use this platform to provide real-time information about ambient and occupancy information that users and facility managers can use to make buildings more efficient and sustainable by considering users' involvement and feedback.

Peer-review under the responsibility of the scientific committee of the Creative Construction Conference 2023.

Keywords: Blueprint, Game Engine Technology, Industry 5.0, Internet of Things, undergraduate research, Unreal Engine.

1. Introduction

Digital twins and Industry 5.0 aim to optimize manufacturing processes by leveraging cutting-edge technology. Industry 5.0 is the next phase in the industrial evolution and emphasizes collaboration between humans and machines, while digital twins are virtual representations of physical assets or processes, and they can be used to simulate, analyze and optimize the performance of a building throughout its lifecycle [1]. For example, with a digital twin, it is possible to optimize the design and operation of a building to reduce energy consumption, improve indoor air quality, and minimize waste [2]. Digital twins can be used to optimize energy usage, reduce waste, and minimize the environmental impact of manufacturing processes. Industry 5.0 aims to create more sustainable and eco-friendly production processes by leveraging advanced technologies like artificial intelligence and the Internet of Things (IoT). Additionally, digital twins can be used to simulate and analyze the impact of different scenarios, such as changes in weather or occupancy patterns, to identify potential issues and opportunities for improvement [3]. A digital twin can be developed on different platforms. One approach is using game engine technology to provide realistic digital environments. In construction research and related applications, game engines are becoming popular. For example, game engines have been used to create interactive training tools for construction safety applications using various technologies such as VR and AR [4]–[15]. Other applications include Interactive Building Information Anatomy Modeling to allow users to dive into the BIM model using VR and retrieve information about building components, such as material information [16] or the development of a workspace management framework integrating the work planning phase with spatial analysis [17]–[19].

One of the key synergies between digital twins and Industry 5.0 is the ability to create more efficient and effective manufacturing processes. Digital twins can simulate production lines, test new designs, and optimize operations at a relatively low cost. This allows manufacturers to identify potential issues before they occur, reduce downtime, and improve product quality. Industry 5.0 emphasizes the collaboration between humans and machines, and digital twins can support this collaboration by providing workers with real-time data and insights that help them make better decisions [20].

With this background, this study presents a proof of concept for developing and integrating a digital twin in a university campus. This allows to gather information and assess in (near) real-time different elements, including temperature, humidity and occupancy, beyond traditional building management systems. This project created a favorable ecosystem that allowed the involved undergraduate students to get hands-on experience developing the components needed, from sensors to model development and connection between the digital and physical environments. This study was also done in collaboration with the industry to support the students and assist with creating the digital model using game engine technology. The ultimate goal was to create a proof-of-concept platform that allows users and facility managers to visualize (near) real-time information about different components of the space they are in and, ultimately, to make buildings more efficient and sustainable.

2. Literature review

Building Information Modeling (BIM) has changed how information can be generated, stored, and exchanged among various stakeholders in the construction industry. However, digital technologies are fast evolving and integrated with many phases of construction projects. Therefore, BIM should be considered in conjunction with these emerging technologies. Sacks et al. [21] indicate that digital twin models, Automated Project Performance Control (APPC), Construction 4.0, and automated data acquisition technologies remain emerging research areas in the AEC industry.

Although the term “Digital Twin” was first attributed to NASA [22], the concept and model of the digital twin were introduced in 2002 by Grieves [23]. He proposed the digital twin as the conceptual model underlying product lifecycle management (PLM) [24]. Different industries have used digital twins for a while for different applications. For example, the aviation industry has been using twin models of aircraft and airports. The manufacturing industry uses digital twins for small components and large factories. These digital twin models are also used for safety and logistics maintenance to maximize the efficiency of the products [25]. Car manufacturers use this technology to analyze the performance of vehicles before production begins [26].

Implementing digital twins requires careful planning and execution to realize the potential benefits of this technology. One of the common concerns for implementing digital twins is the management of data and physical sensors from a wide range of applications. For instance, some urban planning applications require data collection from waterways, air, and soil, which must be managed and centralized for proper use. Some barriers include misalignment of data interpretation which can hinder the future implementation of digital twins. The inputs, outputs, and feedback vary depending on how digital twins are implemented and ultimately used. For example, for whom it is intended (citizens, public administrators, etc.), during which phase (or phases) it is implemented (planning, construction, O&M), and what type of IoT sensors and data are required (noise pollution, air quality, energy, occupancy etc.) need to be well defined and clearly identified at an early stage. To assist with that, Zhao et al. [27] developed a bottom-up framework to ease the implementation of digital twins to support facility managers during the O&M of buildings. They provided an aggregate landscape of DT applications to manage facilities and a conceptual framework for stakeholders concerned with their FM decision-making processes.

The use of digital twins varies across different sectors and spans different disciplines. For instance, Scientists from the National Oceanic and Atmospheric Administration (NOAA) and the National Climatic Data Center (NCDC) use historical rainfall data from digital twins to virtually test the operation of existing stormwater systems. This addresses outdated stormwater systems buckling under the stress of heavier rainfall and increased flooding caused by climate change [28]. As part of the renovation of the UK rail infrastructure, researchers from Cambridge Centre for Smart Infrastructure and Construction (CSIC) developed a digital twin platform to gather real-time data to provide useful information to support engineers and asset managers to schedule safer and proactive maintenance plans [29],[30].

In 2021, the New York State Department of Transportation (NYSDOT) used digital twins to assess and replace the East 138th Street bridge in New York City’s Bronx borough. Completing the project in such a congested area involved complicated structural design and coordination using a digital twin as the primary construction document. The digital twin may now be continuously updated and used as a tool for asset management and bridge inspection [31].

Bortoloni et al. [3] reviewed applications of digital twins in the field of energy efficiency for buildings. They identified 32 articles (from 2019 to 2022). From the review of those articles, they classified them into four topics related to applications of digital twins (1-Design optimization; 2-Occupants' comfort; 3-Building operation and maintenance; and 4-Energy consumption simulation). They concluded that based on the small number of publications found and how recent they were, the use of digital twins in the field of buildings energy efficiency is still in its infancy.

A few studies highlight the potential of digital twins to improve sustainability in the built environment by optimizing building performance, reducing waste and energy consumption, and improving resource utilization. They also suggest that digital twins have the potential to play an important role in the design, construction, and operation of efficient and sustainable buildings and infrastructure systems.

3. Case study

The case study presented in this article extends the preliminary work conducted for the high-level integration of a digital twin at a university campus. For that project, the Blender GIS addon [32] was used to select the area of interest and generate the 3D context in a quick and accurate manner. After that, the 3D model was exported from Blender in fbx. format and imported into Unreal Engine as the platform to model the 3D environment.

The high-level integration of the digital twin consisted of outdoor weather and environmental data obtained from https://api.openweathermap.org and some of the sensors deployed and maintained by the Center for Interacting Urban Networks (CITIES) as part of their CITIESair project [33]. The obtained data included the outside temperature of the area, humidity, pollutant levels (SO_2, CO_2, PM, etc.), the sun's position, and the level of visibility or amount of dust in the weather. Only real data for indoor pollutant levels was available in one building (shown as green in Figure 1). For the rest, dummy data was generated to show proof-of-concept functionality but could be easily replaced with real data once the required sensors are in place. After obtaining these values in real-time, Unreal Engine actors were used to visualize each of these aspects (updated according to the real-time change in these values). For example, the *Volumetric Cloud* actor in the model changes the intensity of the clouds based on the real-time cloud intensity values obtained from the API. With that information, the *Set Tracing Max Distance* function in Blueprints modifies the intensity of the clouds to show a clear or cloudy sky in the game engine platform. An example of the visualization in the game engine platform of that work displaying a mix of actual and generated data for Fine Particles (PM 2.5) is shown in Figure 1. The building in green has data from an actual sensor installed. The rest of the buildings have dummy data for display purposes only. Information in the upper portion of Figure 1 (e.g., Temp, cloud conditions, humidity, wind speed and direction) is real data from the OpenWeather API at the time of the visualization.

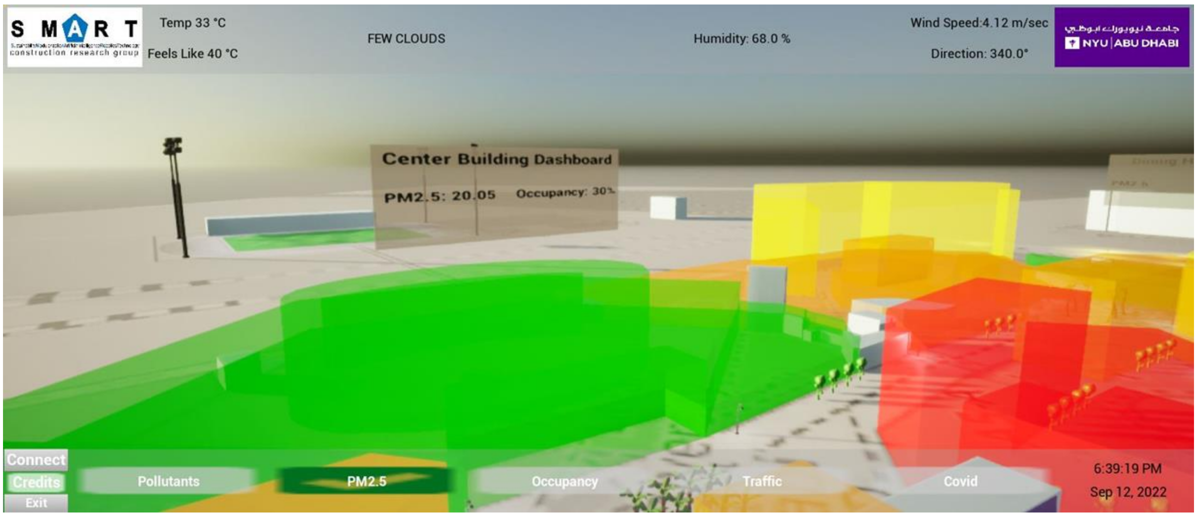


Figure 1: General view of the visualization of different elements considered during the preliminary work for the high-level integration of a digital twin at a university campus.

To expand the previous work, the rest of the study considers a more detailed space (interior of a building) consisting of an open floor-plan office space on a university campus to assess the (near) real-time monitoring of different elements, including temperature, humidity and occupancy, as a proof of concept for developing a digital twin. The process followed for the development of the digital twin involves different steps. The general process for this study is shown in Figure 2.

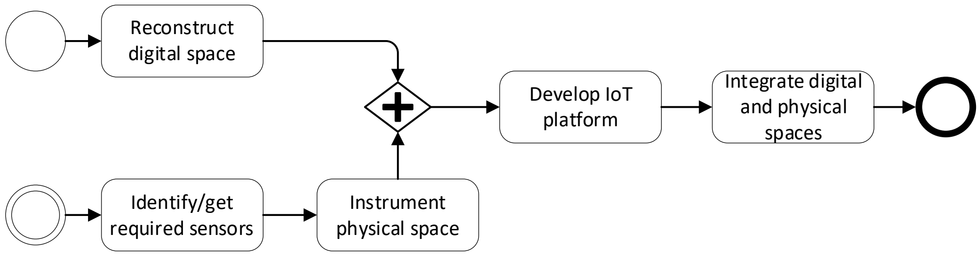


Figure 2: Main steps for the development of the digital twin model for this study

First, it entails reconstructing the digital space (assuming one does not exist already), creating a virtual representation that mirrors the physical environment. Next, real-time data acquisition takes place, capturing information from physical space. This data is then streamed to an Internet of Things (IoT) platform for further processing and analysis. Finally, data streaming is integrated with the digital space, connecting specific elements that are influenced by the (near) real-time data sourced from the physical environment.

3.1. Digital space reconstruction

The space used for this study consists of an open-floor plan office space on a university campus with a total area of approximately 195 m^2 hosting CITIES. Due to the lack of an as-built digital model (e.g., 3D CAD files or BIM) of that space, different tools were evaluated to generate the 3D environment for the digital model. Different instruments were employed to assess their applicability to this case (Table 1). In particular, we utilized three unique types of laser scanning technologies. The Faro Focus laser scanner was primarily used to collect most of the geometric and color data. Nine individual scans were carried out at a consistent height of 1.2 m and four additional scans at a lower height of 0.7 m to accurately depict the area beneath the furniture. A solitary scan at a height of 1.6 m was also performed to encompass the upper sections of tables and shelves.

The Hovermap laser scanner, with its mobile nature, offers a significant advantage over its stationary counterparts like Faro and Leica, as it is capable of scanning areas that are otherwise hidden, thereby reducing occlusion. However, the proficiency of the Hovermap laser scanner mainly lies in scanning geometry, with its ability to capture color data being somewhat limited. Hence, we confined ourselves to using only its positional data.

Ultimately, the Leica scanner was employed to conduct further scans and ensure thorough coverage of the area of interest under various lighting conditions.

Table 1. Main characteristics of the equipment used for reality capture of the space in this study

Equipment	Field of view	3D point accuracy	Range	Point measurement rate	Source
Leica BLK360	360° (horizontal) / 300° (vertical)	± 6mm @ 10m / ± 8mm @ 20m	0.6 - 60 m	Up to 0.36 million points/sec	[34]
Faro Focus S 350 Plus	360° (horizontal) / 300° (vertical)	± 2mm @ 10m / ± 3.5mm @ 25m	0.6 - 350 m	Up to 2 million points/sec	[35]
Emesent Hovermap ST	360° (horizontal) / 290° (vertical) (but moveable)	± 20 mm in general environments ± 15 mm in typical indoor and underground environments ± 5 mm isolated change detection capability	0.4 - 300 m	Single Return Mode: up to 0.3 million points/sec Dual Return Mode: up to 0.6 points/sec	[36]

Different sensors were developed, tested, integrated into the physical space, and linked to the digital model. The digital twin environment was developed in Unreal Engine 4.27 from Epic Games [37]. To test the quality of the point clouds in Unreal Engine, the Lidar Point Clouds plugin [38] was used. The plugin supports different point cloud file formats like E57 and PTS.

One of the challenges when using the point cloud data from the laser scans to the Unreal Engine was that the plugin reduces by default the number of points being visualized to improve the performance of the GPU; however, due to the high number of points included in the scan (over 30,000,000 points), the quality of the scans was reduced to a point where the scans resemble a chaotic (and useless) 3D model. To overcome this issue, this setting can be adjusted to increase the number of points being processed and visualized to improve the quality of the scans. To increase the number of points, users will need to modify the PointBudget variable in the C++ code of the plugin by typing “r.LidarPointCloudBudget” in the command line of the Unreal Engine and increasing this value. We have concluded that any value in the range of 8,000,000 to 10,000,000 points gives an acceptable quality for our model. More complex and bigger scans might need a higher budget to generate a quality 3D environment.

3.2. Real-time data acquisition

The data acquisition system employed in this case study comprises two distinct modules. The first module focuses on the installation of ambient sensors, which collect information including temperature, humidity, and particle concentration. In the second module, real-time detection of space occupancy is taken into consideration. This module utilizes a combination of a static camera and a computer vision algorithm to detect the presence of humans and accurately determine their location within the space.

Figure 2 shows the six locations selected to install two different types of ambient sensors (as described in the next subsection) and a camera for occupancy estimation.

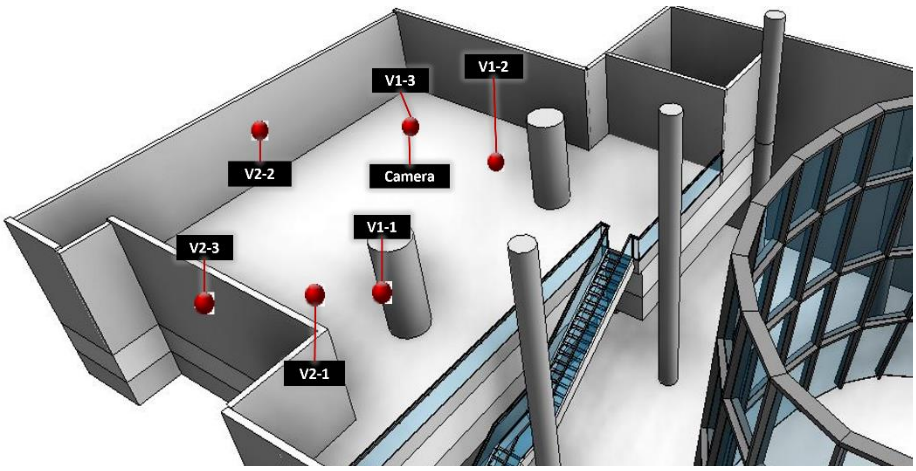


Figure 3: Location of data acquisition sensors used (Sensirion SEN55 (V1-1, V1-2, V1-3), BME280 (V2-1, V2-2 and V2-3) and camera.

3.2.1. Ambient Sensors

Two different types of sensors (Sensirion and BME280) were developed to capture ambient information, including temperature, humidity, and particles (Figure 3). These sensors were specifically selected to actively engage students in the development process. Consequently, the installation of the sensors and the integration of various components were conducted within the laboratory setting. The different elements and main sensors built for this project are summarized next.

The Sensirion SEN55 sensor (Figure 4a) is an all-in-one sensor solution for the accurate measurement of various environmental parameters such as particulate matter, volatile organic compounds (VOCs), oxidizing gases such as nitrogen oxides (NOx), as well as humidity and temperature [39]. It features a 4.5V to 5.5V range, 63mA average current consumption at 5V, a long lifetime, high dust resistance thanks to sheath flow technology, and an I2C interface.

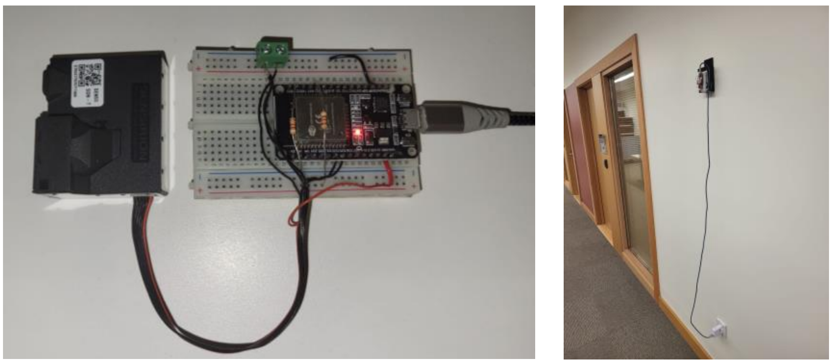
The BME280 (SEN-13676) (Figure 4b) is an atmospheric sensor breakout board that measures barometric pressure, humidity, and temperature readings without too much space. It gives you easy-to-solder 0.1” headers, runs I2C or SPI, takes measurements at less than 1mA and idles less than 5uA [40].

The ESP32 (Figure 4c) is a low-cost microcontroller with integrated Wi-Fi and Bluetooth that can be programmed with the Arduino IDE software [41]. It supports traditional communication protocols and HTTPS and has a crypto-accelerator.

(a) (b) (c)

Figure 4. View of the (a) Sensirion SEN55 [39], (b) BME280 (SEN 13676) [42], and (c) NodeMCU-32S ESP32 [43] used for the development of sensors

The sensors were mounted on breadboards using the ESP32 in Arduino Integrated Development Environment (IDE). An example of the Sensirion circuit connected to the ESP 32 via a breadboard is shown in (Figure 5). The board is powered through the micro USB port either by the laptop while programming or by the battery pack. A similar approach was done for the BME280.



(a) (b)

Figure 5. (a) View of the Sensirion SEN55 circuit mounted on a breadboard with the ESP32, (b) sample view of one of the sensors installed

3.2.2. Occupancy

To estimate the human occupancy in the space investigated, a system able to detect people inside of a room and estimate their positions within the room using a single Monocular Camera was used. Figure 6 shows an overview of the process for estimating the positioning of people in the space.

The occupancy estimation system initiates by capturing the camera feed, which is then subjected to two distinct algorithms for depth estimation and object detection. The TensorFlow object detection API [44] is utilized to identify individuals within the captured feed. Simultaneously, the camera feed is processed in parallel using the MiDaS [45] monocular depth estimation algorithm to estimate depth information. The results from these algorithms are combined, enabling the determination of the (near) real-time relative location of individuals within the video feed. Additionally, the same camera feed undergoes processing to detect the boundaries and center of the captured image. This information plays a crucial role in converting the depth mapping of detected individuals in the feed into global x-y positioning.

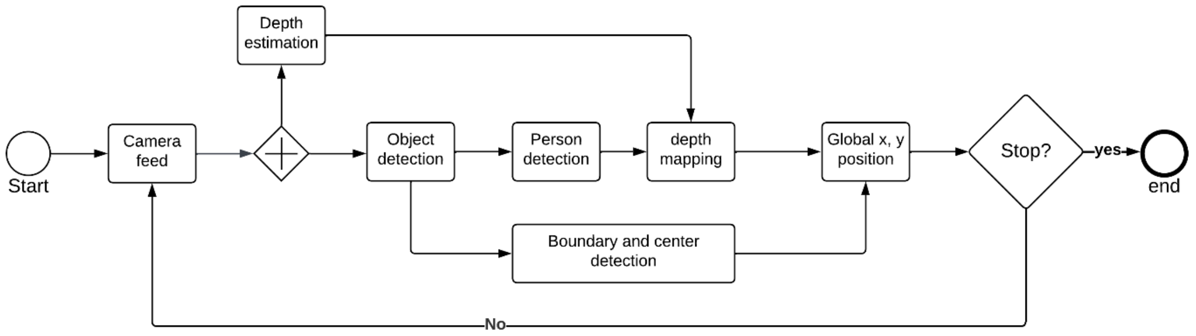


Figure 6. Overview of the process for occupation estimation

3.3. IoT Platform

The real-time data was streamed to the digital twin model using the ThingSpeak [46] IoT platform for the ambient sensors and automatically updating a Google spreadsheet template for occupancy estimation.

In the case of the ThingSpeak platform, a total of six channels representing a total of six sensors were created. The channels contain fields representing each data feed from the sensors. For instance, Sensirion sensors produce six readings: Temperature, Humidity, and particle concentration (PM1, PM2.5, PM4 and PM10) (Table 2). Each reading was registered in a distinct field on a channel. Similarly, the BME280 sensor produces Temperature, Humidity and Pressure data, which are streamed in a distinct channel field.

Table 2. Example of sensor's response data

Field Number	Reading
Field1	Temperature
Field2	Humidity
Field3	PM1.0
Field4	PM2.5
Field5	PM4.0
Field6	PM10.0

On the other hand, the occupancy estimation data was streamed to a Google Spreadsheet, which stored the X and Y coordinates of individuals occupying the space at the given moment. This data is continuously updated, replacing the previous readings every 15 seconds to maintain real-time accuracy.

3.4. Integration of physical space with the digital model

After the physical space has been instrumented, it has to be linked with the digital model. Once the real-time data captured from the sensors is pushed in JSON format, a node-based interface to create gameplay elements from within Unreal Editor (known as Blueprint scripts) was used to get the data in JSON format using the API. The VaRest [30] plugin for Unreal Engine was used in this case. This plugin includes functions allowing to get JSON data from APIs during runtime in the Unreal Engine editor.

Two main Blueprint scripts were developed to implement the flow of information from the sensors and the occupancy estimation. The first Blueprint script was used to obtain the sensor data in JSON format from the ThingSpeak API, while the second Blueprint was used to obtain the occupant's locations from the Google Sheets API.

3.4.1. Blueprint script to obtain the sensor data in JSON format from the ThingSpeak API

The first step to accessing the sensor data was establishing the connection between the Unreal Engine environment and the ThingSpeak API. This was achieved by using the *Call URL* function from the

VaRest plugin. The required input for these functions is the API URL and setting the function to GET the JSON data. The Response Event was used to activate this function and to pull the JSON data from the API. The *Get Response Content as String* function was used to extract the call response in JSON format, which outputs the obtained JSON data in a string format. The second step was to obtain the required data from the JSON response. The JSON response includes an array with six fields, each corresponding to a different reading from the sensor (Table 2).

To obtain the required field or the required reading, a custom function was developed in Blueprints (labeled *JSON Value Getter*). The function has four main inputs and a single output. The single output of the function is the reading value (e.g., temperature, humidity, etc.). The inputs of the function are summarized in Table 3. Figure 7 shows a sample of the Blueprint code to get the temperature reading from one of the sensors. As seen in Figure 7, the first code block (Call URL) is used to obtain the URL of the API with the data. The *Response Event* block initiates the connection between the API and the application. The following code block (*Get Response Content as String*) obtains the API data in JSON format. Finally, the *JSON Value Getter* takes as input the required fields listed in Table 3 and outputs the specific sensor value listed in Table 2.

Table 3. Sensor's Response Data and JSON Function Inputs

Input Variable	Description
Array Name	The name of the array in the JSON response as obtained from ThingSpeak
Array Index	The index of the array in case there are several arrays (0 for our case as we are obtaining the first array)
Field Name	The field name of the reading we want as described in Table 2
Response Node	The JSON request event which holds the JSON data object (Response Event)

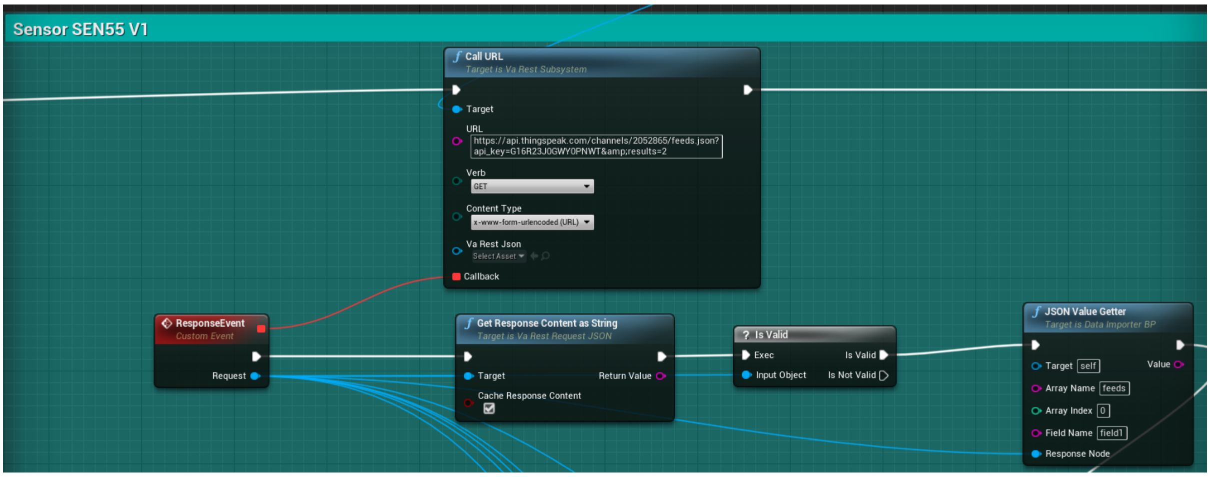


Figure 7. Sensor Data Blueprint

3.4.2. Blueprint to obtain the occupant's locations from the Google Sheets API

The main role of this function is to obtain the number of occupants currently in the room and their specific coordinates within the room on the Google Sheet. The first step is the same as explained above, and the same functions were used to obtain the data in JSON format from the Google Sheet. The only difference was that the URL was replaced by the Google Sheet JSON URL. The second step required a different approach since each row in the Google Sheet is stored as a different array in the JSON object. However, the challenge is that each row does not have a unique field name to obtain the data from in the same way as the sensor data. Each row of data is stored in a single array, and all arrays are stored in a general array having a field name as values. To obtain each row's X and Y coordinate values (each row represents a person in the office space), the *For Each Loop* function was used to loop over each row and obtain the X and Y values from it. The Blueprint script to obtain the Google Sheet data is shown in Figure 8.

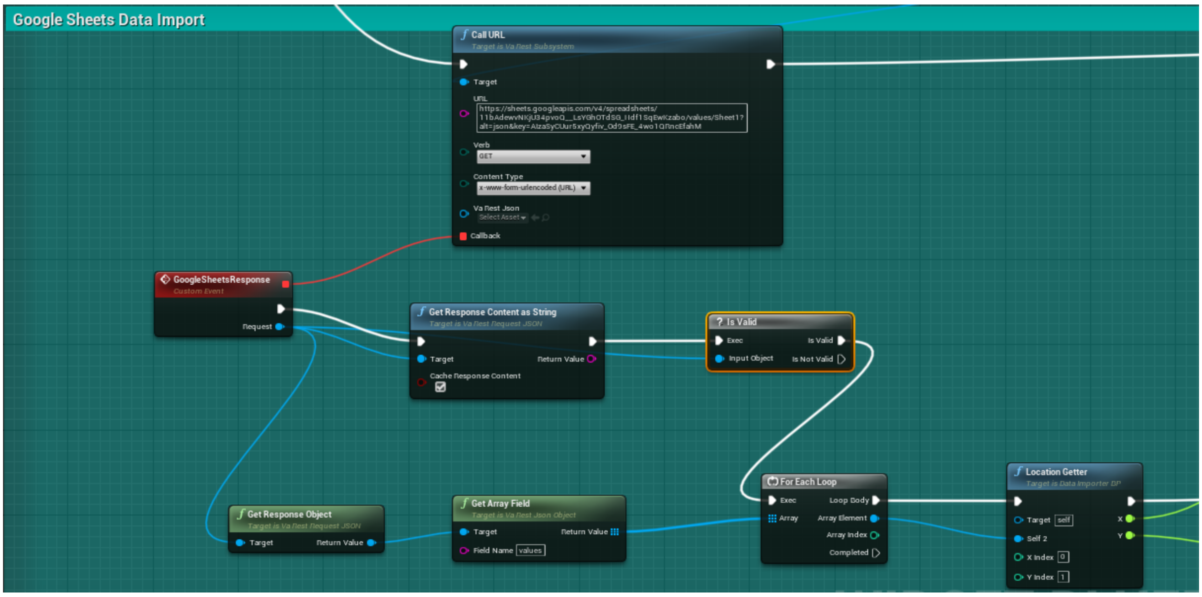


Figure 8. Google Sheet Data Blueprint

3.5. Results: Data visualization

After securing the connection and flow of data from the sensors to the Unreal Engine environment using the Blueprint scripts, the final step for the digital twin development is to visualize the data (i.e., use it as a cockpit or dashboard with information collected from the sensors). This is a key element since it will allow a better understanding of the real-time information of the different elements monitored.

An example of the visualization of the data is shown in Figure 9 (for temperature), Figure 10 (for air quality), and Figure 11 (for occupancy).

Three different sensors were used to monitor and display the temperature, each located in a zone within the space considered for this study. To ensure the occupants' comfort, a value for the office temperature was set to stay between 22 and 23 degrees Celsius. Any value above or below this range was considered out of the comfort range of the occupants and thus was color-coded based on its value. Green was used to indicate an optimum working temperature in the office, orange was used to indicate a change in the temperature, and red was used to indicate a large change (more than 2 degrees from the comfort range defined for this study) in the office temperature.

Based on the real-time data captured from the sensors, the different zones in the office space were color-coded. That can be seen in Figure 9, where Zone 3 is color-coded in orange, meaning there is a slight increase in the ideal temperature in that specific zone, while Zones 1 and 2 are color-coded in red, indicating that both zones have more than 2 degrees of change in temperature from the comfort range. In addition to the color coding of the zones, a bar chart is updated based on the real-time data and integrated into the model to visualize the change in temperature values among the predetermined zones.

Figure 9. Screenshot of the visualization in the Unreal Engine displaying the Office Zoning Color Coding based on temperature readings from the sensors

For the monitoring and display of the air quality as indicated by the PM values, a simple dashboard summarizing the readings along with the safe values for each reading was added to the digital twin (Figure 10). Also, since PM2.5 is considered the most dangerous to human health, a line chart showing the changes in the PM2.5 value was added to the digital twin.

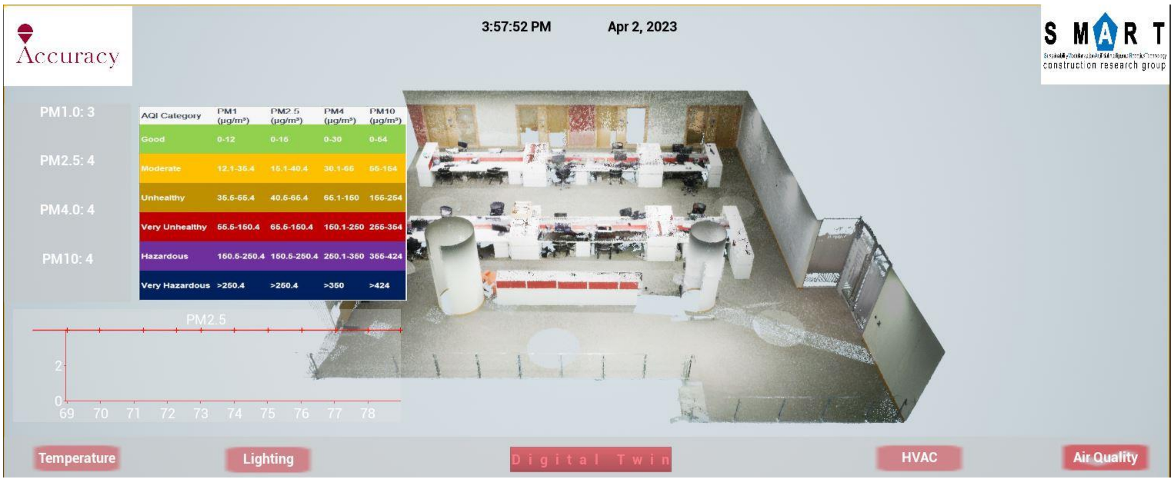


Figure 10. Screenshot of the visualization in the Unreal Engine related to the air quality information

The script for the visualization of the occupancy rate is shown in Figure 8. The first step is to obtain each occupant's location (X and Y coordinates) in the room. This information is sent to a Google Sheet. Afterward, the script spawns or adds a game object, in this case, the 3D avatars shown in Figure 10, to the coordinates. This code is refreshed every 5 to 10 seconds to update the occupants' locations in real-time.

4. Discussion and limitations

Digital twins offer a valuable tool for the AEC industry to optimize the performance of buildings through their lifecycle. By creating a digital representation of physical spaces, building managers can simulate various scenarios to improve energy efficiency, air quality and waste reduction. This can lead to more sustainable building systems.

This study demonstrated the construction of a digital twin model of an indoor space using various reality capture technologies, student-designed and built sensors and an IoT platform. The utilized laser scanner technologies to capture the geometric and color data of the physical environment are selected based on their specific capabilities and advantages, such as reducing occlusion or capturing color data.

Although this process allows to accurately represent the space, processing dense point clouds becomes computationally heavy and working with Unreal Engine was challenging.

Figure 11. Screenshot of the visualization in the Unreal Engine showing the occupancy in the area of interest

Another significant aspect discussed in the study is the real-time data acquisition process. The installed ambient sensors generally collect information such as temperature, humidity, and particle concentration. For a broader understanding of energy efficiency or air quality, a given space can be improved by adding sensors with more parameters (i.e., carbon monoxide concentration, and electric power consumption). However, the data streaming platform utilized in the study, ThingSpeak, provides a fixed number of channels that limit the system's scalability.

The study further discusses the integration of the physical space with the digital model, particularly the use of Unreal Engine, plugins, and Blueprint scripts to link the real-time data obtained from the sensors and occupancy estimation with the digital twin model. They demonstrated obtaining sensor data in JSON format from the ThingSpeak API and occupant location data from the Google Sheets API. This integration enables real-time data visualization within the digital twin, providing a comprehensive and interactive representation of the physical space.

In this proof-of-concept study, a key objective was to engage students in the development of the sensor systems and their integration with the digital model. Therefore, instead of utilizing commercial solutions to perform sensing, students were able to gain a full learning experience by building sensors using basic modules. Initially, the aim was to create sensors that could operate on battery power. This had some limitations primarily related to power source optimization, but it was a great learning experience for the students involved. To address this limitation, the sensors were strategically placed where power outlets were available, allowing them to be powered directly.

5. Conclusions and outlook

Digital twin technology has gained much attention in recent years as it provides a virtual replica of a physical asset or system that can be used to simulate and optimize performance. However, implementing digital twins in the built environment can be challenging for several reasons. For example, digital twins rely on data from sensors and other sources to accurately represent the physical system. However, this data can be difficult to collect and may not always be accurate, leading to potential errors in the digital twin model. Digital twins must be integrated with existing systems for real-time monitoring and analysis. This can be challenging, as many legacy systems were not designed to work with digital twins. Operating digital twins requires significant computational resources, and scaling up to represent larger systems can be challenging. Therefore, careful planning and optimization are needed to ensure the digital twin can handle the required workload. Security becomes important since digital twins can contain sensitive data about the physical system, making them a potential target for cyber-attacks [47]. Security measures must be implemented to protect the digital twin from unauthorized access. Digital twins need to be able to work with other systems and software to provide a comprehensive view of the physical system. This requires standards and protocols to ensure data can be shared between systems. Finally, implementing digital twins can be expensive. They might require significant investments in hardware, software, and personnel to develop and maintain the system.

Future work includes adding other features/sensors and visualization. For example, lighting and HVAC. For lighting, Light Actors can be placed in the scene, which are turned on or off based on their real-time status in the office. As for the HVAC system, the visualization also turns on and off based on the actual status of the system. To visualize the intensity and airflow of the HVAC system, the Niagara system in Unreal Engine can be used. For instance, a colored air flow moving from the location of the HVAC register output (usually in the ceiling or upper portion of a wall) increases in intensity based on the airflow value set in the office. An example of how the visualization of the airflow would be seen in the digital twin is shown in Figure 12.

Figure 12. Airflow Visualization

Acknowledgments

The project was partially funded by Accuracy. This study has benefited from the collaboration with the NYUAD Center for Interacting Urban Networks (CITIES), funded by Tamkeen under the NYUAD Research Institute Award CG001. Thanks to Nabiha Siddique for his involvement in the early stages of this project as part of her NYUAD's Spring 2023 Visiting Student Undergraduate Research Assistantship award sponsored by CITIES. Thanks to Amanda Ramón Constantí for her assistance with the instrumentation of the physical space and generation of CAD files. Thanks to Nikolaos Giakoumidis from the NYUAD Kinesis CTP Lab for his assistance in scanning the physical space.

References

[1] European Commission. Directorate General for Research and Innovation., *Industry 5.0: human centric, sustainable and resilient.* LU: Publications Office, 2020. Accessed: May 10, 2023. [Online]. Available: https://data.europa.eu/doi/10.2777/073781

[2] H. Feng, Q. Chen, and B. Garcia de Soto, "Application of digital twin technologies in construction: an overview of opportunities and challenges," presented at the 38th International Symposium on Automation and Robotics in Construction, Dubai, UAE, Nov. 2021. doi: 10.22260/ISARC2021/0132.

[3] R. Bortolini, R. Rodrigues, H. Alavi, L. F. D. Vecchia, and N. Forcada, "Digital Twins' Applications for Building Energy Efficiency: A Review," *Energies*, vol. 15, no. 19, p. 7002, Sep. 2022, doi: 10.3390/en15197002.

[4] J. K. Dickinson, P. Woodard, R. Canas, S. Ahamed, and D. Lockston, "Game-based trench safety education: Development and lessons learned," *Electronic Journal of Information Technology in Construction*, vol. 16, no. March 2010, pp. 118–132, 2011.

[5] H. Li, G. Chan, and M. Skitmore, "Visualizing safety assessment by integrating the use of game technology," in *Automation in Construction*, Elsevier, Mar. 2011, pp. 498–505. doi: 10.1016/j.autcon.2011.11.009.

[6] L. K.Y, R. A. G., L. C.H, and D. L. L. J, "Developing 3D Safety Training Materials on Fall Related Hazards for Limited English Proficiency and Low Literacy Construction Workers," in *Computing in Civil Engineering*, 2012, pp. 1–25.

[7] L. Yuan-Ling, L. Ken-Yu, L. Min, and C. Nai-Wen, "Learning Assessment Strategies for an Educational Construction Safety Video Game," in *Construction Research Congress*, 2012, pp. 2091–2100.

[8] Y. Fang, J. Teizer, and E. Marks, "A Framework for Developing As-Built Virtual Enviroments to Advance Training of Crane Operators," in *Construction Research Congress 2014*, 2014, pp. 140–149. doi: 10.1061/9780784413517.176.

[9] Q. Le, A. Pedro, C. Lim, H. Park, C. Park, and H. Kim, “A framework for using mobile based virtual reality and augmented reality for experiential construction safety education,” *The International journal of engineering education*, vol. 31, no. 3, pp. 713–725, 2015.

[10] A. Pedro, Q. T. Le, and C. S. Park, “Framework for Integrating Safety into Construction Methods Education through Interactive Virtual Reality,” *Journal of Professional Issues in Engineering Education and Practice*, vol. 142, no. 2, pp. 1–10, 2016, doi: 10.1061/(ASCE)EI.1943-5541.0000261.

[11] Z. Liang, K. Zhou, and K. Gao, “Development of Virtual Reality Serious Game for Underground Rock-Related Hazards Safety Training,” *IEEE Access*, vol. 7, pp. 118639–118649, Aug. 2019, doi: 10.1109/access.2019.2934990.

[12] C. Zhang and B. Chen, “Enhancing Learning and Teaching for Architectural Engineering Students uing Virtual Building Design and Construction,” *Higher Education Studies*, vol. 9, no. 2, p. 45, 2019, doi: 10.5539/hes.v9n2p45.

[13] R. Eiris, M. Gheisari, and B. Esmaeili, “Desktop-based safety training using 360-degree panorama and static virtual reality techniques: A comparative experimental study,” *Automation in Construction*, vol. 109, no. May 2019, p. 102969, 2020, doi: 10.1016/j.autcon.2019.102969.

[14] O. Golovina, C. Kazanci, J. Teizer, and M. König, “Using Serious Games in Virtual Reality for Automated Close Call and Contact Collision Analysis in Construction Safety,” in *Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC)*, 2019. doi: 10.22260/isarc2019/0129.

[15] S. Bükrü, M. Wolf, O. Golovina, and J. Teizer, “Using Field of View and Eye Tracking for Feedback Generation in an Augmented Virtuality Safety Training,” in *Construction Research Congress 2020: Safety, Workforce, and Education*, American Society of Civil Engineers Reston, VA, 2020, pp. 625–632.

[16] C. S. Park, Q. T. Le, A. Pedro, and C. R. Lim, “Interactive Building Anatomy Modeling for Experiential Building Construction Education,” *Journal of Professional Issues in Engineering Education and Practice*, vol. 142, no. 3, pp. 1–12, 2016, doi: 10.1061/(ASCE)EI.1943-5541.0000268.

[17] L. Messi, B. García de Soto, A. Carbonari, and B. Naticchia, “Spatial conflict simulator using game engine technology and Bayesian networks for workspace management,” *Automation in Construction*, vol. 144, p. 104596, Dec. 2022, doi: 10.1016/j.autcon.2022.104596.

[18] L. Messi, B. García de Soto, A. Carbonari, and B. Naticchia, “Intelligent BIM-based spatial conflict simulators: A comparison with commercial 4D tools,” presented at the 39th International Symposium on Automation and Robotics in Construction, Jul. 2022. doi: 10.22260/ISARC2022/0078.

[19] K. Wu and B. García de Soto, “Spatio-temporal planning simulation of temporary elevators in construction using a game engine,” presented at the 38th International Symposium on Automation and Robotics in Construction, Dubai, UAE, Nov. 2021. doi: 10.22260/ISARC2021/0042.

[20] A. Ezzeddine and B. García de Soto, “Connecting teams in modular construction projects using game engine technology,” *Automation in Construction*, vol. 132, pp. 103887–103887, Dec. 2021, doi: 10.1016/j.autcon.2021.103887.

[21] R. Sacks, I. Brilakis, E. Pikas, H. S. Xie, and M. Girolami, “Construction with digital twin information systems,” *DCE*, vol. 1, p. e14, 2020, doi: 10.1017/dce.2020.16.

[22] Mike Shafto *et al.*, “Modeling, Simulation, Information Technology and Processing Roadmap,” National Aeronautics and Space Administration (NASA), Washington, DC 20546, DRAFT, Apr. 2012. Accessed: May 10, 2023. [Online]. Available: https://www.nasa.gov/sites/default/files/501321main_TA11-ID_rev4_NRC-wTASR.pdf

[23] M. W. Grieves, “Virtually Intelligent Product Systems: Digital and Physical Twins,” in *Complex Systems Engineering: Theory and Practice*, S. Flumerfelt, K. G. Schwartz, D. Mavris, and S. Briceno, Eds., Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2019, pp. 175–200. doi: 10.2514/5.9781624105654.0175.0200.

[24] S. Greengard, “Digital Twins Grow Up,” Aug. 06, 2019. https://cacm.acm.org/news/238642-digital-twins-grow-up/fulltext (accessed May 10, 2023).

[25] V. J. Mawson and B. R. Hughes, “The development of modelling tools to improve energy efficiency in manufacturing processes and systems,” *Journal of Manufacturing Systems*, vol. 51, pp. 95–105, Apr. 2019, doi: 10.1016/j.jmsy.2019.04.008.

[26] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, “Shaping the digital twin for design and production engineering,” *CIRP Annals*, vol. 66, no. 1, pp. 141–144, 2017, doi: 10.1016/j.cirp.2017.04.040.

[27] J. Zhao, H. Feng, Q. Chen, and B. Garcia de Soto, “Developing a conceptual framework for the application of digital twin technologies to revamp building operation and maintenance processes,” *Journal of Building Engineering*, vol. 49, p. 104028, May 2022, doi: 10.1016/j.jobe.2022.104028.

[28] OECD, “Climate-resilient infrastructure,” OECD Environment Policy Papers 14, Dec. 2018. doi: 10.1787/4fdf9eaf-en.

[29] Anonymous, “Research Profile - Staffordshire Bridges – long-term performance monitoring using fibre optic sensors,” Sep. 22, 2022. https://www.cdbb.cam.ac.uk/news/research-profile-staffordshire-bridges-long-term-performance-monitoring-using-fibre-optic (accessed May 19, 2023).

[30] Anonymous, “Digital Twin Journeys: smart infrastructure for safety and reliability across the rail network,” Nov. 22, 2021. https://www.cdbb.cam.ac.uk/news/digital-twin-journeys-smart-infrastructure-safety-and-reliability-across-rail-network (accessed May 19, 2023).

[31] J. H. Flynn, B. Crudele, and J. M. Fuda, “East 138th Street Bridge Replacement Project: A Marriage of ABC and Digital Delivery at NYSDOT | Accelerated Bridge Construction.” https://abc-utc.fiu.edu/mc-events/east-138th-street-bridge-replacement-project-a-marriage-of-abc-and-digital-delivery-at-nysdot/?mc_id=789 (accessed May 10, 2023).

[32] domlysz, “Blender GIS.” May 10, 2023. Accessed: May 10, 2023. [Online]. Available: https://github.com/domlysz/BlenderGIS

[33] Center for Interacting Urban Networks (CITIES), “CITIESair.” https://citiesair.com/ (accessed May 20, 2023).

[34] “Leica BLK360 Imaging Laser Scanner.” https://leica-geosystems.com/products/laser-scanners/scanners/blk360 (accessed May 20, 2023).

[35] “FARO Focus Laser Scanner | FARO.” https://knowledge.faro.com/Hardware/Focus/Focus/Technical_Specification_Sheet_for_the_Focus_Laser_Scanner (accessed May 20, 2023).

[36] “Hovermap Series,” *Emesent*. https://emesent.com/hovermap-series/ (accessed May 20, 2023).

[37] Unreal Engine, “Unreal Engine | The most powerful real-time 3D creation tool,” *Unreal Engine*. https://www.unrealengine.com/ (accessed May 10, 2023).

[38] Unreal Engine, “LiDAR Point Cloud Plugin,” *Unreal Engine 5.0 Documentation - LiDAR Point Cloud Plugin*. https://docs.unrealengine.com/5.0/en-US/lidar-point-cloud-plugin-for-unreal-engine/ (accessed May 10, 2023).

[39] Sensirion, “Smart sensor solutions,” *SEN55: Environmental sensor node for PM, RH/T, VOC, NOx measurements*. https://sensirion.com/products/catalog/SEN55/ (accessed May 20, 2023).

[40] “SparkFun BME280 Arduino Library.” SparkFun Electronics, May 01, 2023. Accessed: May 20, 2023. [Online]. Available: https://github.com/sparkfun/SparkFun_BME280_Arduino_Library

[41] “Arduino core for the ESP32, ESP32-S2, ESP32-S3 and ESP32-C3.” Espressif Systems, May 20, 2023. Accessed: May 20, 2023. [Online]. Available: https://github.com/espressif/arduino-esp32

[42] "BME280 Combined humidity and pressure sensor: Final data sheet." BOSCH, May 07, 2015. Accessed: May 20, 2023. [Online]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/4/1/9/BST-BME280_DS001-10.pdf
[43] "NodeMCU-32S - Waveshare Wiki." https://www.waveshare.com/wiki/NodeMCU-32S (accessed May 20, 2023).
[44] Hongkun Yu Chen Chen, Xianzhi Du, Yeqing Li, Abdullah Rashwan, Le Hou, Pengchong Jin, Fan Yang, Frederick Liu, Jaeyoun Kim and J. Li, "TensorFlow Model Garden." 2020. [Online]. Available: https://github.com/tensorflow/models
[45] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, 2022.
[46] "IoT Analytics - ThingSpeak Internet of Things." https://thingspeak.com/ (accessed May 22, 2023).
[47] B. García de Soto, A. Georgescu, B. Mantha, Ž. Turk, A. Maciel, and M. Semih Sonkor, "Construction cybersecurity and critical infrastructure protection: new horizons for Construction 4.0," *ITcon*, vol. 27, pp. 571–594, Jun. 2022, doi: 10.36680/j.itcon.2022.028.