Virtual IoT Device in C# WPF Using Sinric Pro

Sudip Chakraborty [1] & P. S. Aithal [2]
[1] D.Sc. Researcher, Institute of Computer Science and Information sciences, Srinivas University, Mangalore-575 001, India,
OrcidID: 0000-0002-1088-663X; E-mail: sudip.pdf@srinivasuniversity.edu.in
[2] Vice Chancellor, Srinivas University, Mangalore, India,
OrcidID: 0000-0002-4691-8736; E-Mail: psaithal@gmail.com

Subject Area: Computer Science.
Type of the Paper: Experimental Research .
Type of Review: Peer Reviewed as per |C|O|P|E| guidance.
Indexed In: OpenAIRE.
DOI: https://doi.org/10.5281/zenodo.7473766
Google Scholar Citation: IJAEML

How to Cite this Paper:
Chakraborty, S., & Aithal, P. S., (2022). Virtual IoT Device in C# WPF Using Sinric Pro. *International Journal of Applied Engineering and Management Letters (IJAEML)*, *6*(2), 307-313. DOI: https://doi.org/10.5281/zenodo.7473766

International Journal of Applied Engineering and Management Letters (IJAEML)
A Refereed International Journal of Srinivas University, India.

Crossref DOI: https://doi.org/10.47992/IJAEML.2581.7000.0160

Received on: 27/11/2022
Published on: 22/12/2022

Virtual IoT Device in C# WPF Using Sinric Pro

Sudip Chakraborty [1] & P. S. Aithal [2]

[1] D.Sc. Researcher, Institute of Computer Science and Information sciences, Srinivas University, Mangalore-575 001, India,

OrcidID: 0000-0002-1088-663X; E-mail: sudip.pdf@srinivasuniversity.edu.in

[2] Vice Chancellor, Srinivas University, Mangalore, India,

OrcidID: 0000-0002-4691-8736; E-Mail: psaithal@gmail.com

ABSTRACT

Purpose: *IoT is becoming the leading player in the industrial automation environment. In most scenarios, we experiment with IoT using a physical device. We can also research using a virtual device that can perform as real hardware. Without buying any physical hardware, we can visualize the status of the operating load or device which is being triggered through the IoT server or client endpoint. The researcher who is not from an electronics background or does not have sufficient knowledge to continue IoT research can do experiments using it. This paper will show how we can create such kinds of virtual instruments or devices. We use the C# client application adopted from Sinric Pro. We simplified the client module for new researchers in the IoT field for easy understanding. We added some Graphical user elements to display the status of the devices in real-life we see. The code is available for customization.*

Design/Methodology/Approach: *We create and configure the device inside the Sinric Pro IoT server. Download the C# client and customize it for better understanding. We added some graphical elements to display the status of the load. Then we trigger the load from any endpoint like Alexa, the server dashboard, or a mobile phone application.*

Findings/Result: *We get better load status visibility using the GUI element and a minimalistic code structure to send or receive the data to and from the IoT server. Through this concept, we made IoT development or demonstration easy. With slight modifications, we can use this procedure to communicate with any IoT server.*

Originality/Value: *Various clients of fake load for IoT are available. Here we are experimenting more realistic way. Fetching the status and triggers the load using visual indication as the real world does. So it will be more understandable to us, and also, we can trigger the load from our GUI, which has the feature to send the value to the respective load.*

Paper Type: *Experimental-based Research.*

Keywords: Virtual IoT device, Sinric Pro C# client, IoT Demonstration

1. INTRODUCTION :

From morning till night, we are surrounded by Electronics gadgets. Most of the time, we need to be aware of the gadgets' status or need to read the data from devices frequently. IoT, or the internet of things, has become popular to make life more manageable. Using IoT, we can get the status of the device very quickly. Not only that getting the status or triggering something is also a worry less task for us. It makes life easier. However, to experiment with IoT, most of the time, we use hardware. It has some constraints. First, we must purchase the hardware locally or order from an online store. We have to arrange a proper power supply and associated hardware. To work on it, we have to gather some knowledge with a learning curve or hire some help from an expert. Then we will execute our experiment. Also, if any hardware is not performing as expected, we need to debug it, which is time-consuming.

So we need to adopt a better solution. The better solution is to experiment with virtual hardware. Everything is the same, except we use virtual hardware instead of physical. After configuring all the software, our light will glow in the virtual light when we trigger. Here through this work, we will see how we can test the IoT framework without hardware.

Here, we will create and test the virtual IoT hardware through a simple step-by-step procedure. In this experiment, the prerequisite is to be familiar with account creation at the Sinric pro website. We can follow the reference [11] paper for account creation on the Sinric Pro Website. In our GUI, we use picture box control. We can also use animated 2D/3D graphics for a more realistic view. This effort is primary only, and many features might be added to project requirements.

2. RELATED WORKS :

Simiscuka, A. A. et al. introduces a novel social VR-IoT environment, which allows users to share and control local or remote IoT devices in a virtual platform. Their MQTT generates less delay and data traffic than REST [1]. Molina Zarca et al. present and evaluate a novel policy-based and cyber-situational awareness security framework for continuous and dynamic authentication management [2]. Nkenyereye, L. et al. propose an elastic computing algorithm of virtual IoT slice services functions (IoT-SSFs) resources at the NFVI so that the IoT applications and underlying IoT resources can access IoT-SSFs at the network's edge [3]. In their paper, Furfaro A. et al. describe an approach based on exploiting virtual environments and agent-based simulation to evaluate cybersecurity solutions for the next generation of IoT applications in realistic scenarios [5].

3. OBJECTIVES :

This paper's primary purpose is to provide the researcher with reference information on IoT virtual devices. And the other is to show, using a minimalistic approach, how we can achieve the software stack for client node execution. The virtual device can be used when availability and complexity arise on real hardware. We use the same software stack as we use inside the physical hardware. For demonstration purposes, it is excellent. It will help those who are trying to build virtual IoT devices.

4. APPROACH AND METHODOLOGY :

The central part is the C# main UI thread. It coordinates among various objects. When the application opens, the C# Web socket client communicates with the Sinric Pro IoT server by sending the available device's id. The servers acknowledge by sending the time stamps. After a while, the server sends all the last available device statuses to the client. Our virtual devices update the device status accordingly and keep them up to date. Now our virtual device is ready to receive commands. It is called active listening mode. If the command reaches the server, it will parse, and the load of the virtual device will change its current status. Figure 1 depicts the project's logical workflow diagram.

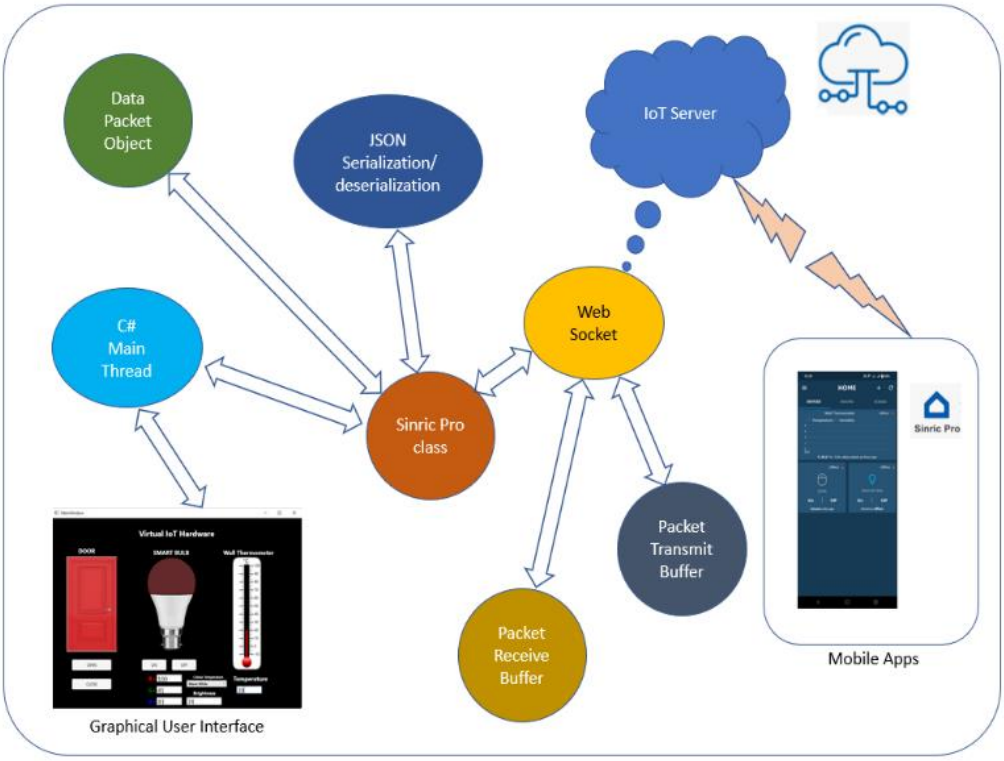

Fig. 1: Block diagram of project logical workflow

The client Web Socket object maintains two buffers. The Packets received and sent buffer. If it gets any packet from the server, it is pushed to the receive buffer. It periodically checks the transmit buffer; any data present inside it is immediately sent to the destination and deque from the buffer.

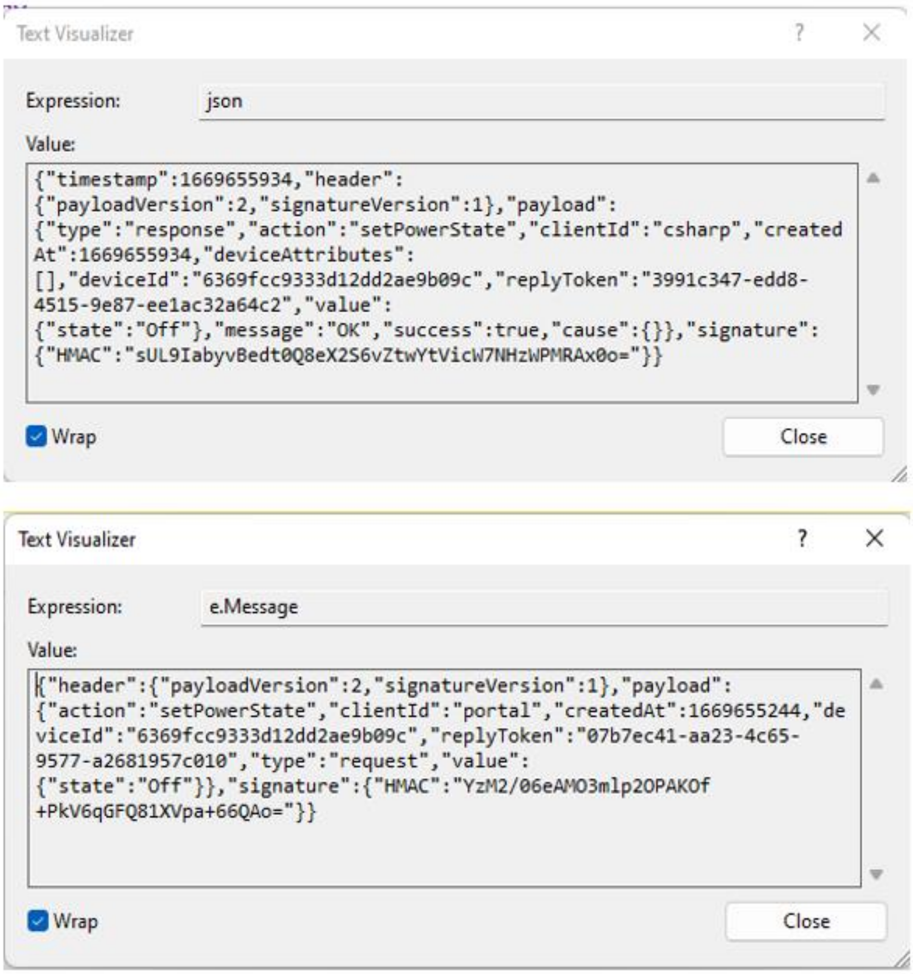

Fig. 2: Sample Communication Packet

The application layer periodically checks for packet arrival. If any packet receives, it is deserialized using JSON object and parses the data. Reading the device ID changes the status of the corresponding devices. When the user changes any device status, the Sinric Pro object forms a packet, filling a blank object. After packet formation, it pushes to the web sockets transmit buffer. The WebSocket periodic poll senses the outgoing queue and sends it to the cloud server. Figure 2 depicts sample packets in JSON format.

Using the Sinric Pro cloud server account, we can install and operate the device from android mobile apps. Download the apps and install them from the Google play store. Login using the cloud credentials. Change the device status and observe the device status has changed in our application which is our virtual device.

5. EXPERIMENT :

Now we will do some experiments. We need to follow the step-by-step procedure below to do our experiment.

Create a cloud account: open the Sinric Pro website. Create an account. If it does not exists, Create it. YouTube videos can guide us. The paper "How to make IoT in C# using Sinric Pro [11] may be a helpful reference.

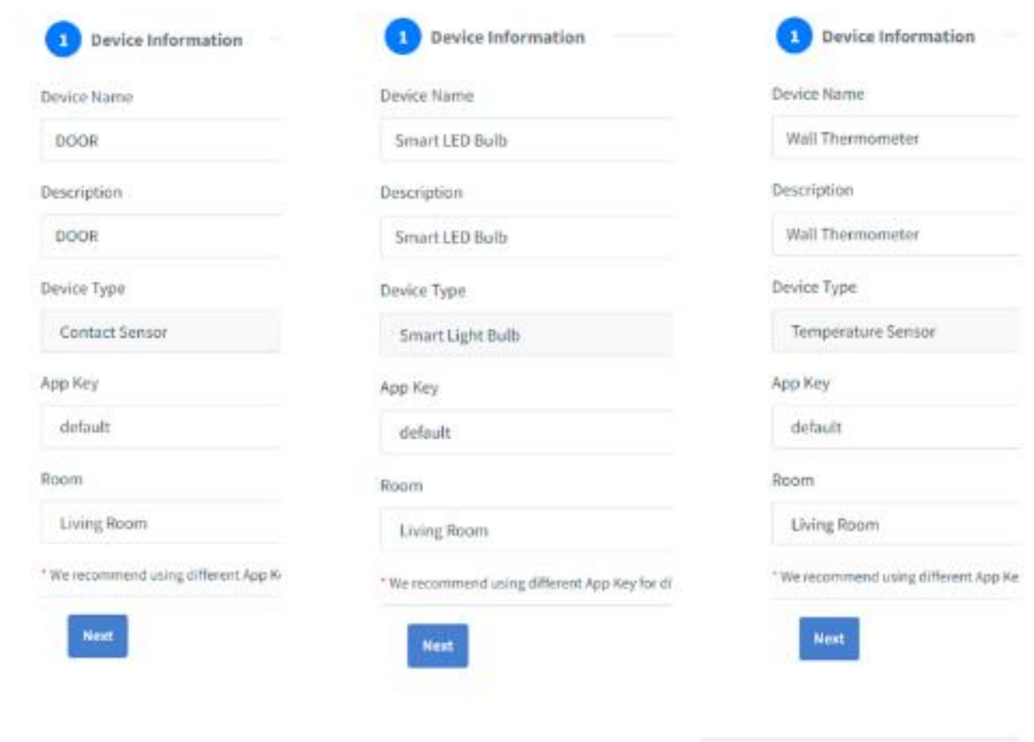

Fig. 3: Devices in the Cloud Server

After creating an account, we need to make three devices, as depicted in figure 3. We can create any number of devices, but the first three are free. After device creation, keep a local copy of the App key, a Secret key, and three device IDs. Inside the cloud, work is done.

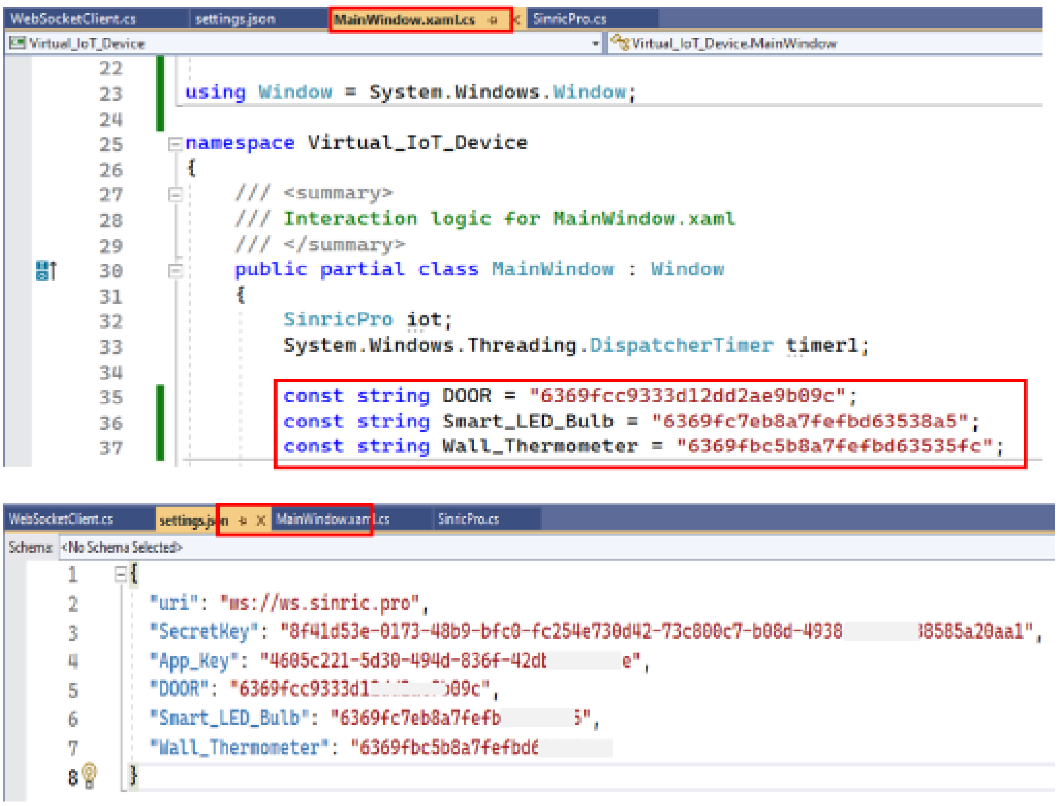

Fig. 4: Devices in the Cloud Server

Project Build: Now clone the project from Github. Before building the project, we need to follow some procedures. Navigate inside the project folder, Open MainWindow.XAML.cs and the settings.json file add the device credential, as depicted in figure 4.

Project Execution: Now, build the project. It should build successfully. If not, debug it. Maybe some dependency issues need to solve by installing it using the Nuget package manager. If the application runs properly, it should look as depicted in figure 5. Now we interact with a button. Press the button under the door "open." the door will open locally, and the status will be sent to the remote IoT server. Open the Sinric Pro website and keep logged in. We observe the popup message is showing when we click the button. Open the Sinric Pro mobile app and change the switch position. We will watch that the IoT server dashboard status is changing, and our virtual load status is also updating.

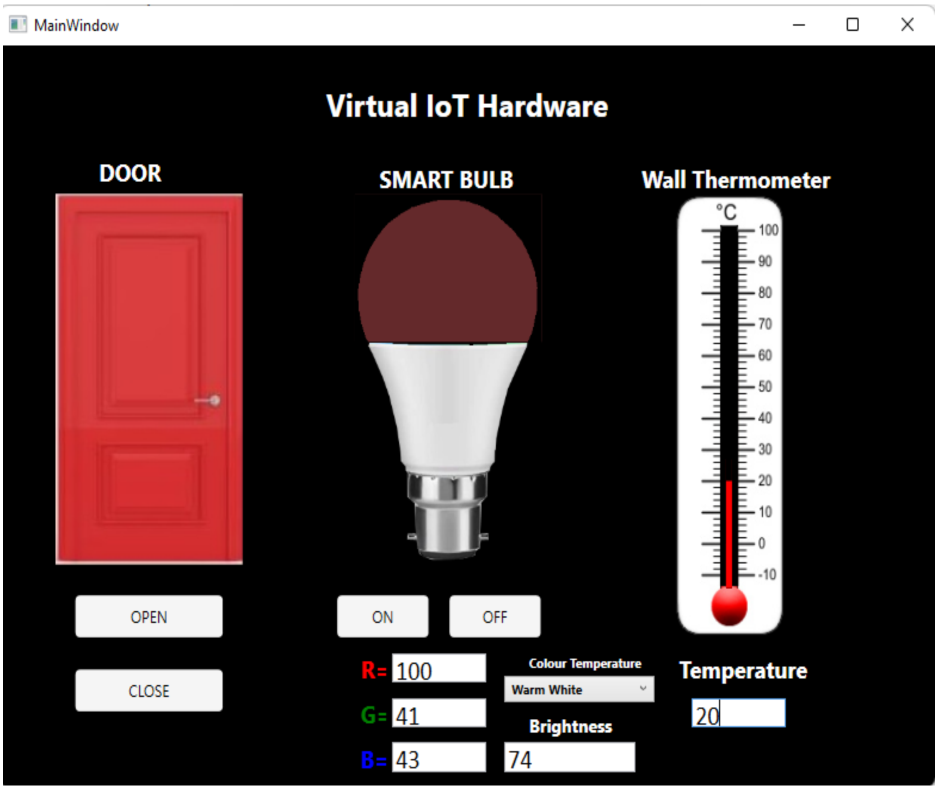

Fig. 5: Virtual Device Interface

6. RECOMMENDATIONS :

- The GitHub code link: https://github.com/sudipchakraborty/Virtual_IoT_Device.git
- Sinric Pro Sample message https://github.com/sinricpro/sample_messages
- For sinric pro account set up: https://www.srinivaspublication.com/journal/index.php/ijcsbe/article/view/1980
- With minimal hardware (without Wifi on Board), it can also be triggered to the physical load from the application. We can send the current reading to the less efficient controller board to activate the mechanical contact relay through any communication channel like RS232.
- We can add any number of virtual devices. We showed the procedure here using three devices.

7. CONCLUSION :

Throughout the paper, we discussed creating an IoT virtual device that acts like the actual hardware. It can be used for the experiment, protocol or algorithm development, demonstration, and many more purposes without engaging physical hardware. It has no risk of shocking hazards or other dangers like real hardware. It is entirely safe to use. We use the Sinricpro website as an IoT server. It helps to trigger the load over the internet. Following the procedure described, we can utilize minimal effort to activate our load with different IoT servers.

REFERENCES :

[1] Simiscuka, A. A., Markande, T. M., & Muntean, G. M. (2019). Real-virtual world device synchronization in a cloud-enabled social virtual reality IoT network. *IEEE Access*, *7*, 106588-106599. Google Scholar↗

[2] Molina Zarca, A., Garcia-Carrillo, D., Bernal Bernabe, J., Ortiz, J., Marin-Perez, R., & Skarmeta, A. (2019). Enabling virtual AAA management in SDN-based IoT networks. *Sensors*, *19*(2), 295, 1-24. Google Scholar↗

[3] Nkenyereye, L., Hwang, J., Pham, Q. V., & Song, J. (2021). Virtual IoT service slice functions for multiaccess edge computing platform. *IEEE Internet of Things Journal*, *8*(14), 11233-11248. Google Scholar↗

[4] Ullah, I., Ahmad, S., Mehmood, F., & Kim, D. (2019). Cloud-based IoT network virtualization for supporting dynamic connectivity among connected devices. *Electronics*, *8*(7), 742, 1-28. Google Scholar↗

[5] Furfaro, A., Argento, L., Parise, A., & Piccolo, A. (2017). Using virtual environments for the assessment of cybersecurity issues in IoT scenarios. *Simulation Modelling Practice and Theory*, *73*, 43-54. Google Scholar↗

[6] Vettel, A., & Clayton, R. (2019, November). Homeware: A virtual honeypot framework for capturing CPE and IoT zero days. In *2019 APWG Symposium on Electronic Crime Research (eCrime)* (pp. 1-13). IEEE. Google Scholar↗

[7] Tang, S., Shelden, D. R., Eastman, C. M., Pishdad-Bozorgi, P., & Gao, X. (2019). A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends. *Automation in Construction*, *101*, 127-139. Google Scholar↗

[8] Simiscuka, A. A., & Muntean, G. M. (2018, June). The synchronization between real and virtual-world devices in a VR-IoT environment. In *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)* (pp. 1-5). IEEE. Google Scholar↗

[9] Marquez, J., Villanueva, J., Solarte, Z., & Garcia, A. (2016). IoT in education: Integration of objects with virtual academic communities. *New advances in information systems and technologies* (pp. 201-212). Springer, Cham. Google Scholar↗

[10] Li, H., Ota, K., & Dong, M. (2019). LS-SDV: Virtual network management in large-scale software-defined IoT. *IEEE Journal on Selected Areas in Communications*, *37*(8), 1783-1793. Google Scholar↗

[11] Chakraborty, S., & Aithal, P. S., (2022). How to make IoT in C# using Sinric Pro. *International Journal of Case Studies in Business, IT, and Education (IJCSBE)*, *6*(2), 523- 530. Google Scholar↗

****