Let Us Create an Alexa-Enabled IoT Device Using C#, AWS Lambda, and ESP Module

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

Area/Section: Computer Science.
Type of the Paper: Experimental Paper.
Type of Review: Peer Reviewed as per |C|O|P|E| guidance.
Indexed in: OpenAIRE.
DOI: https://doi.org/10.5281/zenodo.8260291
Google Scholar Citation: IJMTS

How to Cite this Paper:
Chakraborty, S., & Aithal, P. S. (2023). Let Us Create an Alexa-Enabled IoT Device Using C#, AWS Lambda and ESP Module. *International Journal of Management, Technology, and Social Sciences (IJMTS), 8*(3), 256-261. DOI: https://doi.org/10.5281/zenodo.8260291

International Journal of Management, Technology, and Social Sciences (IJMTS)
A Refereed International Journal of Srinivas University, India.

CrossRef DOI: https://doi.org/10.47992/IJMTS.2581.6012.0301

Received on: 04/06/2023
Published on: 18/08/2023

Let Us Create an Alexa-Enabled IoT Device Using C#, AWS Lambda and ESP Module

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: *Voice-operated query is becoming popular. Alexa Such a device that is now widely used. Several vendors provide built-in Alexa services to their devices. In our fast day-to-day life, it is the most relevant Technology. It is safe and convenient also. Doing multiple work at a time is a common scenario. We can quickly get weather updates while cooking. We use our voice to operate AC, Fan, light, and electronic gadgets. Several tech giants like Amazon made Technology easy for us. Our non-Alexa devices can easily convert into Alexa enabled. We can operate our device or electric equipment just over voice. Here we provide the procedure for creating an Alexa-enabled device quickly. Using several AWS services like Alexa Skill, Lambda, and IoT, We can quickly build an Alexa-enabled custom device for us.*
Design/Methodology/Approach: *The Alexa-enabled devices can be built following a series of processes. To create the Alexa-enabled device, we develop Alexa skills inside the Alexa developer console, write and upload command processor Lambda function using C# language inside the Microsoft visual studio, and configure the IoT profile inside the AWS IoT cloud server. Furthermore, finally, create ESP firmware to operate the equipment. To send the voice command, we need any Alexa-enabled devices. It may be an echo dot, mobile phone, or PC application.*
Findings/Result: *Combining Amazon Alexa with IoT devices presents a potent synergy, bringing immense potential for improved user experiences and expanded automation and control capabilities. Developing an Alexa-enabled IoT device with C# and AWS Lambda is relatively uncomplicated, provided you possess some programming skills and familiarity with cloud computing. The device gains internet connectivity by leveraging an ESP module, facilitating seamless communication with the Alexa Voice Service, and empowering voice-based commands and responses. The versatility of this device spans a wide array of applications, encompassing the management of lights, switches, appliances, and more, all of which can be further tailored to specific user requirements.*
Originality/Value: *Several documents are available to create Alexa-enabled devices. Here we provide the procedure with a block diagram and data flow so that the researcher can easily create and integrate it into their research work. All information is practical-oriented. We can quickly build Alexa-enabled devices using the practical example following this work.*
Paper Type: *Experimental-based Research.*
Keywords: Alexa, IoT (Internet of Things), C#, AWS Lambda, ESP module, Voice-activated devices, Smart home technology, Cloud computing, Remote control, Home automation.

1. INTRODUCTION :

The Internet of Things (IoT) has rapidly transformed how we interact with Technology, becoming an integral part of our daily lives, from smart homes to wearable devices. One exciting aspect of IoT is that voice-controlled devices provide seamless, hands-free experiences. Among the popular voice assistants, Amazon's Alexa shines for its versatility and extensive capabilities. This project will explore creating an Alexa-enabled IoT device using C#, AWS Lambda, and an ESP module. Combining these

technologies allows us to tap into Alexa's robust voice recognition to control our IoT devices effortlessly.

To create an Alexa-enabled device need to follow several steps. First, we develop Alexa skills in the AWS Alexa developer console. We prepare our utterances under custom Intents according to the equipment triggers. After that, we prepare the AWS Lambda function to execute the request. Add Lambda ‘s Endpoint to Alexa skill so the text token can dispatch to the Lambda function. After the lambda function, we need to create an AWS IoT profile. Create a Shadow register so that Lamda can update the command on the Alexa voice command. Finally, we need an ESP hardware module. Using Arduino IDE AWS MQTT communication made easy. Add IoT and Wifi credentials into ESP firmware. Run the system, and talk in front of Alexa. It converts into a voice token and sends it to the Alexa cloud services. Alexa services parse it, convert it into text, and send it to the Lambda Cloud services. The lambda services update the shadow register. The IoT services update the ESP module over the MQTT protocol.

2. RELATED WORKS :

In this study, Ho et al. focus on the forensic analysis of intelligent homes and identifying Distributed Denial of Service (DDoS) attack patterns on IoT devices [1]. Arora et al. explore the integration of artificial intelligence (AI) and virtual assistant technologies. They present a working model that leverages AI techniques to enhance the capabilities of virtual assistants [2]. The review by Ilyas et al. examines the use of virtual and remote labs as immersive pedagogy tools in educational settings [3]. Noorani et al. present a study on automatic malware detection on an Alexa-Pi IoT device. The authors propose a framework that leverages machine-learning techniques to detect and mitigate malware threats on IoT devices [4]. Arya and Patel discuss the implementation of Google Assistant and Amazon Alexa on the Raspberry Pi platform [5]. Mitev et al. propose "Leakypick," an IoT audio spy detector system to identify potential eavesdropping attacks in intelligent home environments [6]. Chung and Lee discuss the capabilities of intelligent virtual assistants in understanding and responding to user needs. They explore integrating AI technologies with virtual assistants, enabling them to adapt to user preferences [7]. Sanchez examines the legal implications and privacy concerns surrounding Alexa-enabled intelligent home devices. The author discusses the concept of "digital curtilage" and its application to smart home devices [8]. Einarsson et al. propose SmarthomeML, a domain-specific modeling language (DSML), for creating intelligent home applications [9]. Mitev et al. investigate skill-based man-in-the-middle (MITM) attacks on virtual assistants such as Alexa. The authors demonstrate the vulnerability of virtual assistants to skill-based attacks, where malicious skills intercept and manipulate user commands [10]. This literature review overviews various research studies and articles on IoT devices, virtual assistants, and innovative home technologies.

3. OBJECTIVES :

To explore the capabilities and limitations of using an ESP module for Internet of Things (IoT) devices. We discussed the benefits of integrating Amazon Web Services (AWS) Lambda with an ESP module for IoT applications. We examined the C# programming language for developing Alexa-enabled IoT devices. To design and develop a functional prototype of an Alexa-enabled IoT device using C#, AWS Lambda, and an ESP module. We evaluate the performance and user experience of the developed prototype. We provide insights and recommendations on improving the design and implementation of Alexa-enabled IoT devices using C#, AWS Lambda, and an ESP module.

4. APPROACH AND METHODOLOGY :

(1) **IoT hardware development using ESP Module**: to trigger the electrical equipment using Iot, we need some hardware with wifi to connect with the Internet. The famous chip at a low cost is the ESP module. It has several variations. According to our requirements, we can purchase any module. Here we will demonstrate ESP32. Among all ESP modules, it Is balanced like pin count and speed aspect. Using the Arduino IDE, we can program it. There are several documents available to program it. We have one paper demonstrating how to create Iot using the ESP module, which link was added later.

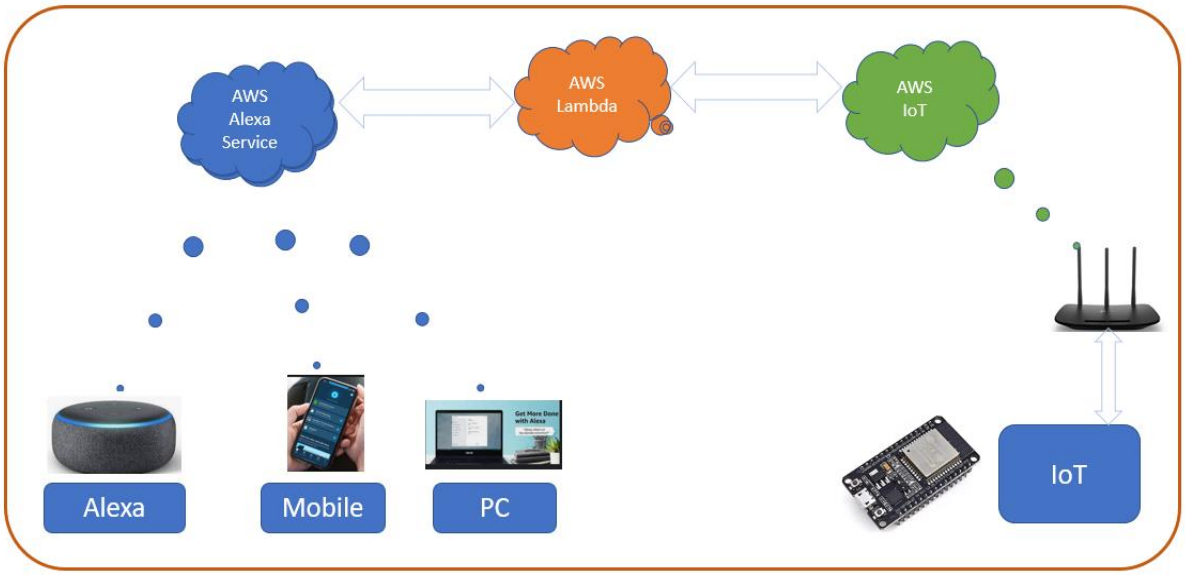

Fig. 1: Research methodology used

Fig. 1 depicts the research methodology. It consists of several parts. We are discussing one by one now.

(2) **IoT device creation in the AWS server**: Once the Iot program is over, The next step is to create an Iot account in the AWS server. Suppose an account is not available need to make it. AWS might be required a credit card for account creation. For new accounts, a one-year free version is available. After one year, we can switch to any suitable plan. The plan can be customizable on data traffic basics. After IoT creation, we can test it using the AWS cloud MQTT test client option. We must subscribe to or publish the topic to send or receive the data. When we create IOT inside the AWS cloud, we create shadow devices. We can update the shadow devices from any application using an IoT ARN. When the shadow devices update, AWS triggers the device means AWS sends the update topic to the connected IoT device, i.e., the ESP module. When an IoT device needs to send the updated device register to the AWS cloud, it sends an update topic to the AWS cloud. This topic updates the device's shadow value. For that, all connected clients will be notified by AWS IoT services.

(3) **Aws lambda function development**: The lambda function bridges Alexa skill and IoT devices. When it receives the command from Alexa, it process and parses it. Then it forwards the command to the IoT Services, which resides in the cloud. The lambda function is widespread nowadays. It is payable as they execute the processes.

(4) **Alexa skill development**: The Alexa skill is the input of the whole system. Over the voice command, we can trigger the electrical equipment. We must create an Alexa skill in the AWS cloud Alexa console to operate over the voice command. First, we need to create an AWS account. Navigate the Alexa developer console. Create a new skill and add slots. Once the skill is developed. We need to add Endpoints so that it can trigger the lambda function. Once the configuration is over, there is a test interface to test the Alexa console. We can have Alexa echo dot, or just through our mobile, we can send the voice command. At first, you need to invoke the skill name along with the "Open" command. Then say any device trigger command. It will trigger the load, either on /off. Or some analog-type value, like fan speed 50% like that.

4. EXPERIMENT :

(1) **ESP32 Development Board**: We need one physical ESP module to execute experiments on Alexa enabled device. Here we selected the ESP32S module, available in online stores like Amazon. We need to create an AWS account in the AWS IoT server. The paper on IoT device creation Inside the AWS: https://doi.org/10.5281/zenodo.7726980 [11]. Create a shadow register to update the command, which acts like a bridge between Alexa and ESP module. The article to create a physical IoT Device: https://doi.org/10.5281/zenodo.7779097 [12].

(2) **Alexa Developer Console**: There is an Alexa developer console. Inside the console, we need to create one skill to take the input from voice, convert the token, and send it to the AWS server. The server process the command and send it to the AWS Lambda. The paper to make Alexa Skill in the AWS cloud server: https://doi.org/10.5281/zenodo.7940237 [13].

(3) **AWS Lambda**: When we talk in front of Alexa. Alexa creates a voice token and sends it to the AWS Alexa server. The server process the command. Convert to text and send to the Lambda. Aws lambda do two process. It processes the command and back response to the Alexa Endpoint. It also updates the IoT shadow register. The lambda function is developed using the C# language. After development, we can able to deploy to the cloud. AWS provided their SDK, freely available from their website and Nuget package manager. The article to create a Lambda function inside the AWS server: https://doi.org/10.5281/zenodo.7995727 [14].

(4) **Amazon Echo Dot**: Once the application is developed. It is time to test the command. For that, we need either any Alexa eco dot or online. Inside the Alexa developer console, there is a test interface. Through that, we can send the command to our device.

5. RECOMMENDATIONS :

- The below-enlisted papers can be treated as a reference for IoT-based research work. All papers are practical-oriented.
- A good youtube link on Lambda in C#: https://www.youtube.com/watch?v=BPWmaYJIjCk&t=632s
- The researcher interested in IoT using Sinric Pro can Navigate: https://www.srinivaspublication.com/journal/index.php/ijcsbe/article/view/1980 [15].
- One good reference on Virtual IoT device Creation using Sinric Pro: https://www.srinivaspublication.com/journal/index.php/ijaeml/article/view/2093/817 [16].
- The article Create Multiple IoT Device Controller: https://doi.org/10.5281/zenodo.7857660 [17].

6. CONCLUSION :

Alexa-enabled devices are becoming popular. Using the voice command, we can operate any equipment. It has several advantages over voice commands. There is no chance of getting shocked by the operated devices. Moreover, when both hands are busy, we can trigger the electrical equipment using our voice. Not only the local machines but we can also operate any devices from remote locations. Here, through a step-by-step procedure, we demonstrate how to create an Alexa-enabled IoT device. For this, we used AWS IoT as an IoT server, and to trigger the devices wirelessly, we used a popular ESP Wifi module. The researcher trying to integrate Alexa into their research can get some reference information from this work.

REFERENCES :

[1] Ho, S., Greeson, H., & Karabiyik, U. (2022). Smart Home Forensics: Identifying Ddos Attack Patterns on IoT Devices. 01-13. Google Scholar↗

[2] Arora, S., Athavale, V. A., Maggu, H., & Agarwal, A. (2020). Artificial intelligence and virtual assistant—working model. In *Mobile Radio Communications and 5G Networks: Proceedings of MRCN 2020* (pp. 163-171). Singapore: Springer Singapore. Google Scholar↗

[3] Ilyas, Z., Rihawi, Z., Azmat, F., & Al Sebae, A. (2019). Laboratory Experience as an Immersive Pedagogy Tool Using Virtual and Remote Labs: A Review. *Excellence in Engineering Education for the 21st Century: The Role of Engineering Education Research*, 261. Google Scholar↗

[4] Noorani, M., Mancoridis, S., & Weber, S. (2019). Automatic malware detection on an Alexa-Pi IoT device. In *35th Annual Computer Security Applications Conference (ACSAC 2019)*. Google Scholar↗

[5] Arya, S. D., & Patel, D. (2020). Implementation of Google Assistant & Amazon Alexa on Raspberry Pi. *arXiv preprint arXiv:2006.08220*. Google Scholar↗

[6] Mitev, R., Pazii, A., Miettinen, M., Enck, W., & Sadeghi, A. R. (2020, December). Leakypick: Iot audio spy detector. In *Annual Computer Security Applications Conference* (pp. 694-705). Google Scholar↗

[7] Chung, H., & Lee, S. (2018). An intelligent virtual assistant knows your life—*arXiv preprint arXiv:1803.00466*. Google Scholar↗

[8] Sanchez, J. (2020). A New Era: Digital Curtilage and Alexa-Enabled Smart Home Devices. *Touro L. Rev.*, *36*, 663. Google Scholar↗

[9] Einarsson, A. F., Patreksson, P., Hamdaqa, M., & Hamou-Lhadj, A. (2017, June). SmarthomeML: Towards a domain-specific modeling language for creating innovative home applications. In *2017 IEEE International Congress on Internet of Things (ICIOT)* (pp. 82-88). IEEE. Google Scholar↗

[10] Mitev, R., Miettinen, M., & Sadeghi, A. R. (2019, July). Alexa lied to me: Skill-based man-in-the-middle attacks on virtual assistants. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security* (pp. 465-478). Google Scholar↗

[11] Sudip Chakraborty, & P. S. Aithal. (2023). Let Us Create An IoT Inside the AWS Cloud. *International journal of case studies in business, IT, and education (IJCSBE), 7*(1), 211–219. https://doi.org/10.5281/zenodo.7726980.

[12] Sudip Chakraborty, & P. S. Aithal. (2023). Let Us Create A Physical IoT Device Using AWS and ESP Module. *International journal of management, technology, and social sciences (IJMTS), 7*(1), 224–233. https://doi.org/10.5281/zenodo.7779097.

[13] Sudip Chakraborty, & P. S. Aithal. (2023). Let Us Create an Alexa Skill for Our IoT Device Inside the AWS Cloud. International journal of case studies in business, IT, and education (IJCSBE), 7(2), 214–225. https://doi.org/10.5281/zenodo.7940237.

[14] Sudip Chakraborty, & P. S. Aithal. (2023). Let Us Create A Lambda Function for Our IoT Device In The AWS Cloud Using C#. *International journal of management, technology, and social sciences (IJMTS), 8*(2), 145–155. https://doi.org/10.5281/zenodo.7995727.

[15] Chakraborty, S., & Aithal, P. S. (2023). Let Us Create Our Desktop IoT Soft-Switchboard Using AWS, ESP32 and C#. *International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7*(3), 185-193. DOI: https://doi.org/10.5281/zenodo.8234036

[16] 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.

[17] Sudip Chakraborty, & P. S. Aithal, (2023). Let Us Create Multiple IoT Device Controller Using AWS, ESP32 And C#. *International journal of applied engineering and management letters (IJAEML), 7*(2), 27–34. https://doi.org/10.5281/zenodo.7857660.

****