DESIGN AND DEVELOPMENT OF AN IOT DEVICE CONTROLLED BY AMAZON'S ALEXA VOICE ASSISTANT

Angel Toshkov

Abstract: This article shows how we can build our own IoT device using a memory-limited type of AI compatible directly with Amazon's built-in artificial intelligence voice assistant Alexa. For this purpose, a hardware model with a microcontroller from the ESP32 series was used, with the help of which an Alexa compatible IoT device was realized. The hardware module consists of a microcontroller ESP32-Device Kit, five different colored LEDs, five resistors limiting the current through the diodes to the appropriate level. All these elements are mounted on one solderless mini board. The LEDs are connected to a general purpose GPIO. In the examples, the LEDs are used to indicate the control action resulting from the AI decision based on the data received from or intended for the IoT device. In a real situation, the LED can be replaced by another executive device - for example, a relay or relay matrix, a motor control unit, etc.

Keywords: IoT, ESP32, Alexa, voice control, AI, Echo Dot

ПРОЕКТИРАНЕ И РАЗРАБОТКА НА IoT УСТРОЙСТВО, УПРАВЛЯВАНО ОТ ГЛАСОВИЯ АСИСТЕНТ ALEXA НА AMAZON

Ангел Тошков

Абстракт: Тази статия показва как можем да изградим собствено IoT устройство, използвайки ограничен в паметта тип AI, съвместим директно с вградения изкуствен интелект на Amazon, гласов асистент Alexa. За целта е използван хардуерен модел с микроконтролер от серията ESP32, с помощта на който е реализирано Alexa съвместимо IoT устройство. Хардуерният модул се състои от микроконтролер ESP32-Device Kit, пет различни цветни светодиода, пет резистора, ограничаващи тока през диодите до подходящото ниво. Всички тези елементи са монтирани на една мини платка без спойки. Светодиодите са свързани към GPIO с общо предназначение. В примерите светодиодите се използват за указване на контролното действие, произтичащо от решението на AI въз основа на данните, получени от или предназначени за IoT устройството. В реална ситуация светодиодът може да бъде заменен с друго изпълнително устройство - например реле или релейна матрица, блок за управление на двигателя и др.

Ключови думи: IoT, ESP32, Alexa, гласов контрол, изкуствен интелект, Echo Dot

Introduction

To get an idea of the use of AI in IoT systems, we first need to familiarize ourselves with some terminology.

What is AI (artificial intelligence)?

There are many attempts to define what artificial intelligence is. Some of them, for example, are:

- Artificial intelligence is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence.

- AI is a class of applications that interpret conditions and make decisions, similar to the way people respond to their senses, but without requiring direct human intervention.

According to some specialists, we can classify artificial intelligence into 4 distinct types. [1].

Reactive AI - This type of AI is good for simple tasks, such as classifying and recognizing images or sounds; works well when all possible input parameters are known; can do calculations very quickly.

It cannot handle scenarios involving lossy information or those requiring historical data.

Memory-limited AI - This type of AI can handle complex classification tasks. May use historical data to make decisions or make predictions; Capable of complex tasks such as self-driving cars, but still vulnerable to non-standard data or deliberate actions.

This is assumed to be the current state of AI.

AI using Theory of Mind - These are AI's capable of understanding human motives and reasoning.

It can provide a personalized experience to everyone based on their motivations and needs; Able to learn with fewer examples because he understands motive and intent; Considered the next milestone in Al's evolution

Self-aware - Human-level intelligence that can resemble our intelligence.

What is an IoT?

The Internet of Things or IoT is a system of interconnected computing devices, mechanical and digital objects, machines, living objects that are equipped with a set of sensors

and a unique identifier (UID). They are the ability to transfer data over a network without requiring human-to-human or human-computer interaction.

How does IoT work?

An IoT ecosystem consists of intelligent devices with network communication capability and access to web-based applications that use embedded systems, such as processors, sensors, and communication hardware, to collect, send, and act on the data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other end device, where the data is sent to the cloud to be analyzed or analyzed locally. Sometimes these devices communicate with other connected devices and act on the information they receive

from each other. The devices do most of the work without human intervention, although people can interact with the devices—for example, set them up, give them instructions, or access the data.

IoT with AI.

The use of AI in IoT systems is leading to the emergence of IoT with AI. These are intelligent devices capable of collecting information and processing it using AI technologies.

In this material, we will look at the possibility of creating and using basic IoT with AI.

The most common use cases at the moment are IoT devices using Reactive AI and Memory-limited AI.

In this document, microcontrollers from the company Espressif, especially ESP32 series were used. ESP32 is an accessible and multi-purpose platform for IoT applications.

In order to be able to use the power of the Intelligent Agents of Google and Amazon, we can used intermediaries such as Adafruit.io, IFTT or Synrik Pro. These and other intermediaries offer a variety of professional services that we can use depending on the goals of our projects.

However, in this example we will show how we can create our own IoT device using Memory-limited type of AI, compatible directly with the voice assistant with embedded artificial intelligence - Amazon's voice assistant - Alexa voice assistant.

For this purpose, it is necessary to have a hardware mockup, for example with a microcontroller from the ESP32 series, with the help of which we can implement our Alexa compatible IoT device.

For this example, we will use our hardware scheme, shown below Fig. 1.

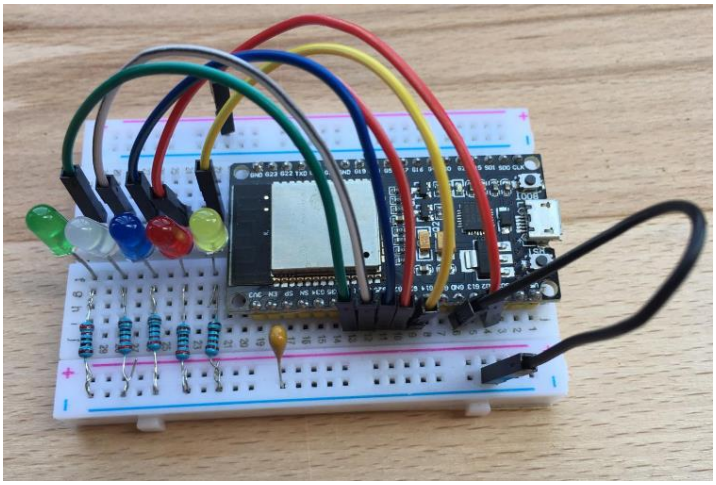

Fig.1 Test hardware circuit

Implementation of the project

The hardware module consists of a microcontroller ESP32-Device Kit [2], five different colored LEDs, five resistors limiting the current through the diodes to the appropriate level. All these elements are mounted on a single solderless mini board. The LEDs are connected to a general purpose GPIO. In the examples, the LEDs are used to indicate the control action resulting from the

AI decision based on the data received from or intended for the IoT device. In a real situation, the LED can be replaced with other executive devices - for example, with relays or a relay matrix, a motor control unit, or others.

Before start, we need to make the following preparations:

1. To have Arduino IDE installed, latest version. [3]

2. To add and select in the IDE the board you are working with, in the case ESP32 DevKit.

3. To have included one of the above links in the "Additional URLs field" in the board manager. You can add multiple URLs by separating them with commas. Fig.2

https://raw.githubusercontent.com/espressir/arduino-esp32/gh-pages/package_esp32_dev_index.json

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

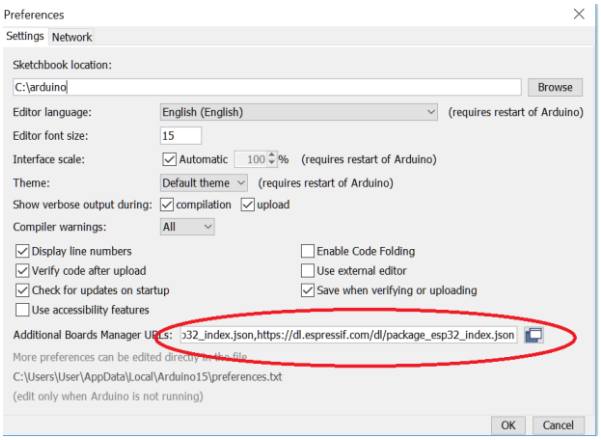

Fig.2 You can add multiple URLs by separating them with commas.

4. To add the library Espalexa [4] from Tools/Manage Libraries Fig. 3

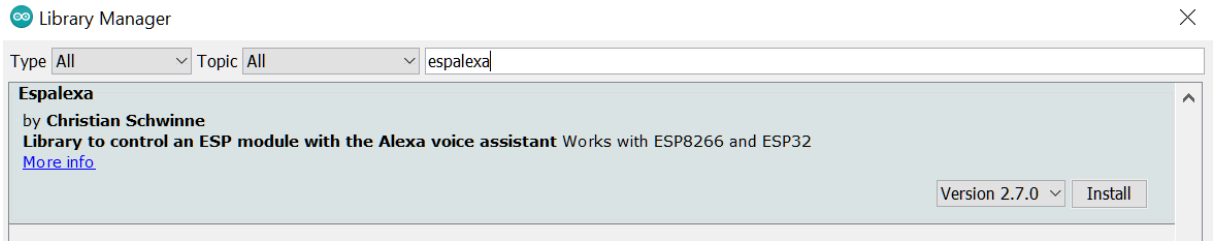

Fig. 3 Arduino IDE Library manager

5. Create and load the code into your test hardware mockup.

When building the code, we define the libraries and connect to WiFi:

#ifdef ARDUINO_ARCH_ESP32

#include <WiFi.h>

```
#else
#include <ESP8266WiFi.h>
#endif
#include <Espalexa.h>
```

define the devices – for example:

```
  // Define your devices here.
  espalexa.addDevice("Device 1", firstLED_Channel);       //simplest definition, default state off
  espalexa.addDevice("Device 2", secondLED_Channel, 255);  //third parameter is beginning state from 0 to 255. (here fully on)
      // Second way to define new device
  device3 = new EspalexaDevice("Device 3", thirdLED_Channel);
      //you can also create the Device objects yourself like here
  espalexa.addDevice(device3);
      //and then add them
  device3->setValue(128);
       //this allows you to e.g. update their state value from 0 to 255 at any time!
  espalexa.begin();
```

And we use callback functions – for example:

```
void firstLED_Channel(uint8_t LED1_State) {
  // You can do what you need to do here
  //  Example for "light1"
  Serial.print("Device 1 changed to ");
  if (LED1_State) {
    Serial.print("ON, LED1_State ");
    Serial.println(LED1_State);
    digitalWrite(light1, HIGH);
  }
  else {
    Serial.println("OFF");
    digitalWrite(light1, LOW);
  }
```

}

6. Will be need to create or have an account in Amazon.com. Fig.4

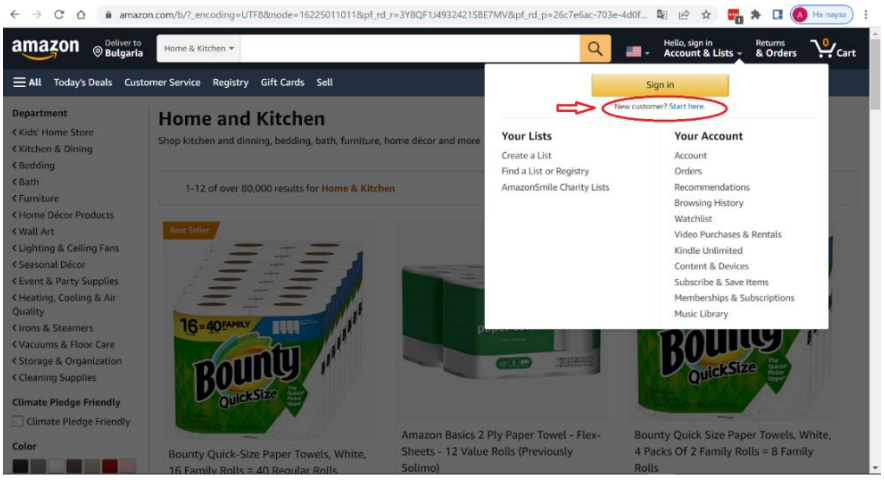

Fig.4 Amazon's main page

This is your general Amazon account for purchases and other activities.6. To have Alexa app installed on your smartphone.

7. To have Alexa Echo Dot or compatible with Amazon Alexa Voice Assistant device Fig. 5. For example:

Amazon Echo Dot 3, Alexa

Set Up the Amazon Echo Dot [5]

Note: Menus and figures may differ depending on whether you are using iOS or Android, which version and type of your device.

Devices from the Amazon DOT series, such as the Alexa Amazon Dot 3 are smart enough, but I still recommend that you read and follow the instructions provided by the manufacturer.

Typically, the setup goes like this: When you turn on the Echo and a blue light will appear, then it will turn orange and Alexa will greet you and tell you how to complete the setup using the app. Make sure you enable the Bluetooth on your smart device - phone or tablet.

If Echo's Alexa doesn't prompt you for set-up, go to the App and select "Devices" in the lower right corner. Then choose "Add Device" with either the plus sign in the top right corner or the More menu (the three horizontal lines) in the lower right corner Fig. 6.

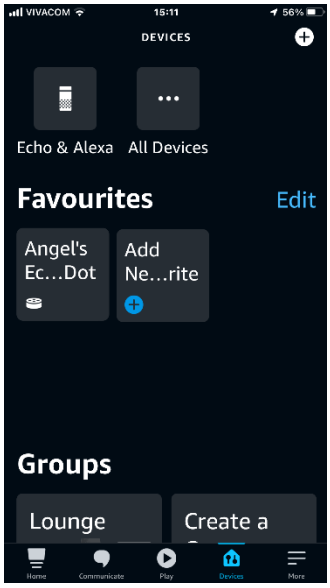

Fig. 6 "Devices" page in Alexa

Follow the prompts on the App to choose your device. For this example choice "Other" and next "Discover Devices".

Alexa Dot will usually detect the new devices and invite you to add them in app. When the devices are added and ready Fig. 7, you will see something like this:

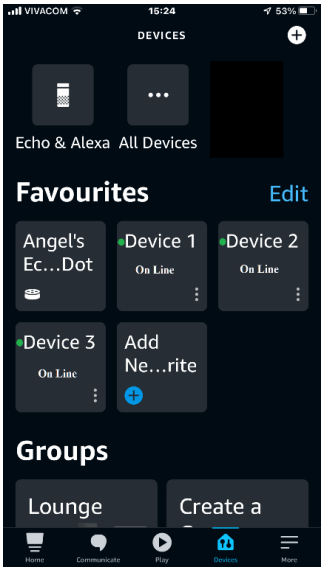

Fig. 7 When the devices are added and ready

You can view the status of each device from the "Menu" by selecting the appropriate device Fig.8.

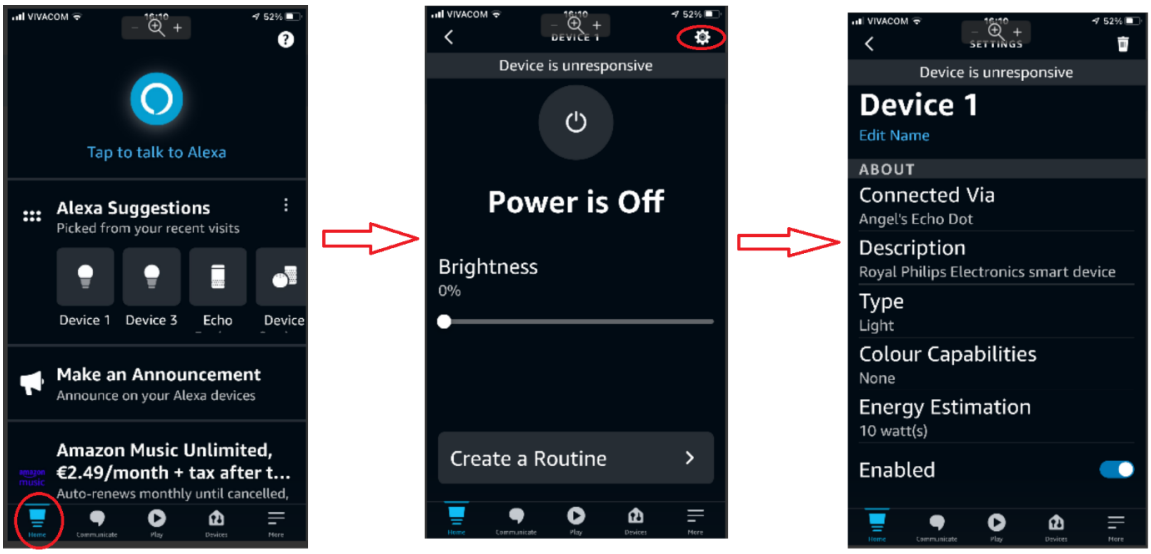

Fig.8 Status of each device from the "Menu"

Device parameters are usually predefined in the source code. For example for “Device 1”:

// Define your devices here.

espalexa.addDevice("Device 1", firstLED_Channel); // default state off

8. Set up WiFi

For your Alexa DOT device to work properly, you need to choose the right Wi-Fi network and enter a password to this Wi-Fi. You can do this by selecting the Alexa Echo Dot device and entering the same network and password that your smart IoT devices are connected to.

9. Give room name

You can also select a room where your Echo Dot is located and give that room a name.

Now you can control your devices through the Alexa app, either with voice commands to the Alexa Dot or through the voice assistant in the Alexa app on your smartphone.

Sample commands”

„Alexa, turn on Device 1“

“Alexa, turn off Device 1”

“Alexa, tell me if Device 1 is on”

You can try other commands. For more commands, search and read Amazon Alexa commands.

Conclusions

The article proposed a way to create your own IoT device using the capabilities of the intelligent voice assistant Alexa with artificial intelligence. This approach allows us to develop smart IoT without

limitation in the field of application, according to its own design and designed to solve specific tasks. The wide distribution and availability of hardware components, as well as the constant improvement of their parameters at an affordable price, gives us the opportunity to solve a wide range of control and regulation problems and tasks.

References

1. (https://www.bmc.com/blogs/artificial-intelligence-types/)
2. https://www.espressif.com/
3. https://www.arduino.cc/en/software
4. https://github.com/Aircoookie/Espalexa
5. https://www.amazon.com/gp/help/customer/display.html?nodeId=202016340