Setup Guide

Raspberry Pi GPIO to Alexa to connection

Team LT United

03.04.2020

Noroff University College

Introduction

This guide is the result of a group project at Noroff University College, started in 2019 and finished in 2020. The purpose of this document is to provide a walkthrough, describing how to create a connection from Alexa to various electric circuits, connected to a Raspberry Pi.

The Raspberry Pi is a small single-board computer created by the Raspberry Pi Foundation, and it has a General Purpose Input Output (GPIO) interface, which can be controlled by different programs written by users.

By connecting electric circuits to the GPIO pins and using the appropriate software, it is possible to use the Amazon Alexa smart assistant to manipulate the electronic.

The intended audience for this project is IT students or anyone who wants to learn more about how IoT, Voice Activation and the Alexa Smart Assistant works.

The inspiration for this project was the Instructable by the user PatrickD126 at URL: https://www.instructables.com/id/Control-Raspberry-Pi-GPIO-With-Amazon-Echo-and-Pyt

This project uses components from the small electronic kit from this URL: https://www.aliexpress.com/item/32968988155.html

Installation - Setting up the Raspbian OS on the micro SD card

Step 0 - Equipment list / prerequisites:

- A Raspberry Pi 4 model b (or similar).
- USB-C Power Supply (original Raspberry Pi unit recommended).
- A micro SD card (size of at least 8 GB, 16 GB or more recommended).
- A computer with a modern Windows OS and micro SD card reader.
- Software: Web browser, 7-Zip, Rufus or Etcher, SSH and VNC Viewer.
- Wi-Fi or RJ45 Network cable.

Step 1 – Download the Raspbian OS archive

First, download the Raspbian OS from the following link:

https://www.raspberrypi.org/downloads/raspbian

Select the latest version with desktop and recommended software, as seen in figure 1.

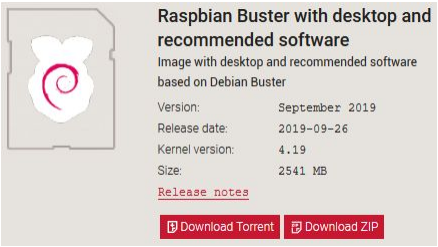

Figure 1. – *OS files*

Step 2 – Validate the SHA-256 Digital Fingerprint (optional)

A SHA-256 hash value signature is listed below the download link as seen in figure 2.

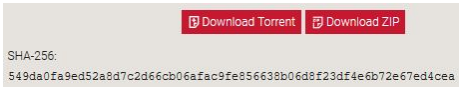

Figure 2. - *SHA-256 hash value signature*

The signature can be used to verify that the file has not been altered or corrupted. To verify it, right-click the file in the Windows File Explorer then select the CRC SHA tool

and choose the SHA-256 option. The result with examples of matching values can be seen below in figure 3.

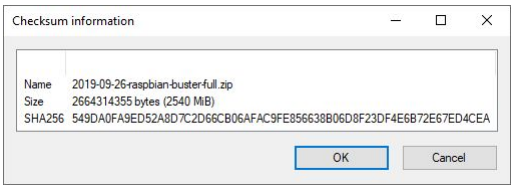

Figure 3. - *SHA-256 value*

Step 3 – Extract the file in the OS archive

The OS files are compressed in a .zip archive and can be extracted with a zip utility like 7-Zip available at https://www.7-zip.org

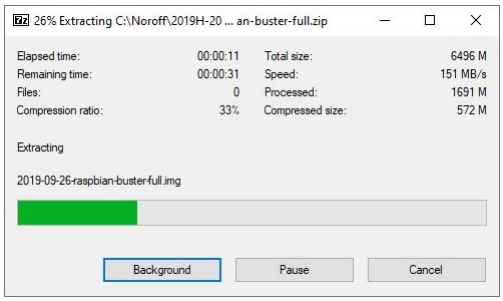

Figure 4. - *Extracting the archive file using 7-Zip*

The result is a file like “2019-09-26-raspbian-buster-full.img” which can then be used to write the OS to the micro SD card.

Step 4 – Cleaning up old partition information (optional)

Before flashing the micro SD card, it might be a good idea to clean up any old partition information on the micro SD card if it has been used previously.

This can be done by running the “diskpart” command in a command prompt window (cmd.exe) with administrative rights.

The “list disk” command can be used to find the correct disk, and then the “select” command can be used to select it. Then the “clean” command can be used to clean up any old partition information. An example is shown in figure 5.

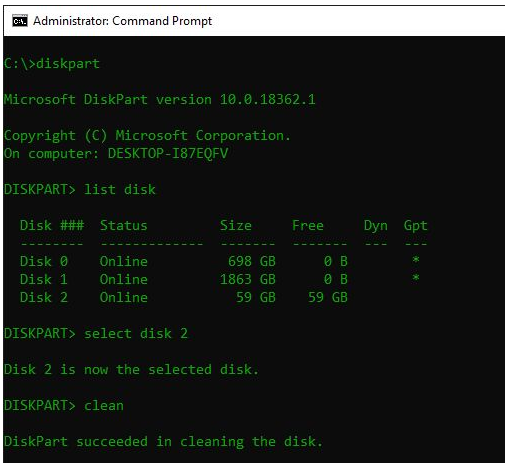

Figure 5. - *Cleaning up old partitions*

Source for more details:

https://www.youtube.com/watch?v=NaRMS7DoSAQ

Step 5 – Prevent Windows 10 from writing metadata to the SD card (optional)

To prevent Windows 10 from automatically writing system volume information to the micro SD card, this can be avoided by enabling a restriction in the OS which is used to write the image to the card.

Run gpedit.msc and browse to “Computer Configuration -> Administrative Templates -> Windows Components -> Search”, then select “Do not allow locations on removable drives to be added to libraries” and enable this as seen in figure 6.

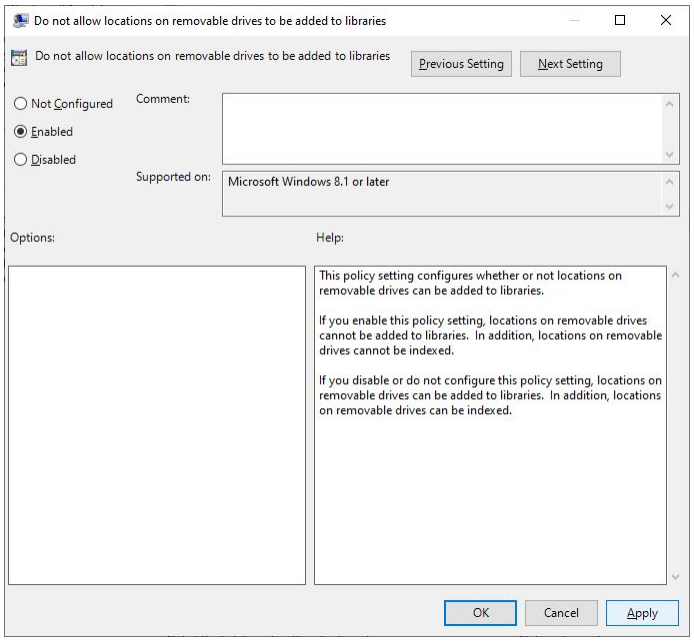

Figure 6. - *Enabling restrictions for removable drives*

This can also be done by using regedit and browsing to “HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows” then add a new DWORD “DisableRemovableDriveIndexing” and set it to 1 if it does not already exist.

Remember to reboot so the changes can take effect!

Some other services might also have to be disabled in Windows 10, like windows search. Consult these sources for more details:

https://www.thewindowsclub.com/prevent-system-volume-information-folder-usb

https://superuser.com/questions/1199823/how-to-prevent-creation-of-system-volume-information-folder-in-windows-10-for

Step 6 – Write the OS image file to the micro SD card

The .img file can now be written to the micro SD card by using a software tool like Rufus, Etcher or Win32DiskImager, available from:

https://rufus.ie
https://www.balena.io/etcher https://sourceforge.net/projects/win32diskimager

First, the image file and destination need to be selected as seen in figure 7. Then the process of writing to the micro SD card can be started as seen in figure 8.

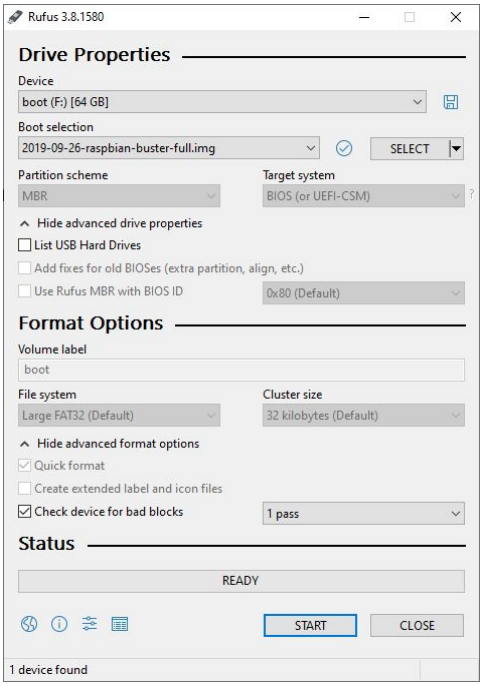

Figure 7. – *Destination and source image file selected*

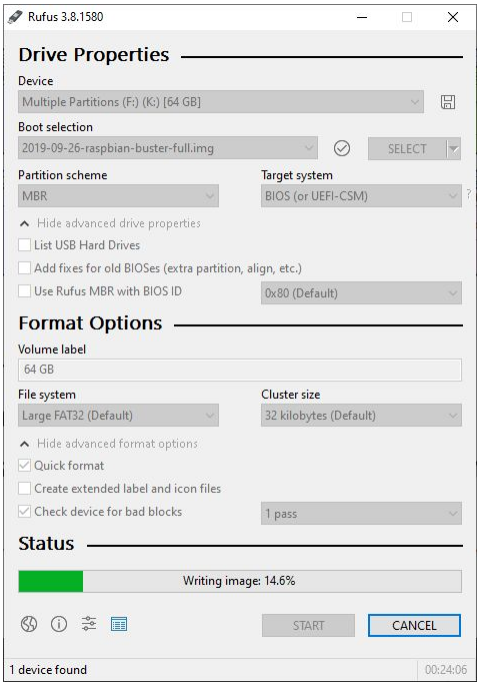

Figure 8. – *Writing files to the micro SD card*

Step 7 – Insert the micro SD card into the Raspberry Pi and boot

The micro SD card can now be inserted into the Raspberry Pi and the device can be turned on to boot the OS.

Configure the Raspbian OS

First boot

Once the SD card is inserted, and the power is turned on, a multi coloured screen should appear, followed by a message about resizing the partition size.

The display will show a welcome screen. Follow the dialogues to configure the Raspbian OS installation.

The Setup Wizard

The setup wizard is a graphical guide to help configure the OS. Follow the dialogue, as shown below in the figures 9 to 15.

Figure 9. – *Setup wizard for the installation*

Select local settings for the OS.

Figure 10. – *Select prefered local language*

It is important to change the password before connecting to the network.

Figure 11. – *Set a new password*

If there is a black edge on the monitor, this can be corrected.

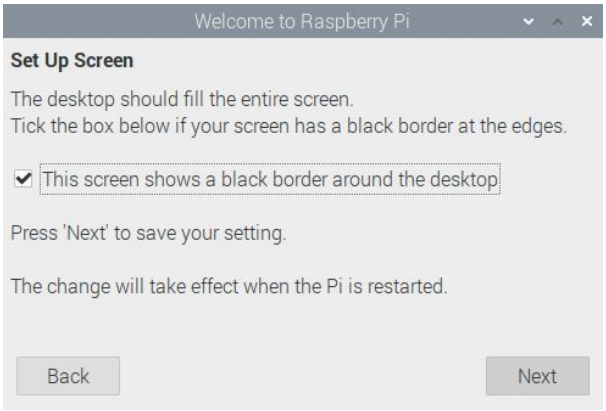

Figure 12. – *preventing the black border around the edge of the screen.*

Select the Wi-Fi to connect the Raspberry Pi to.

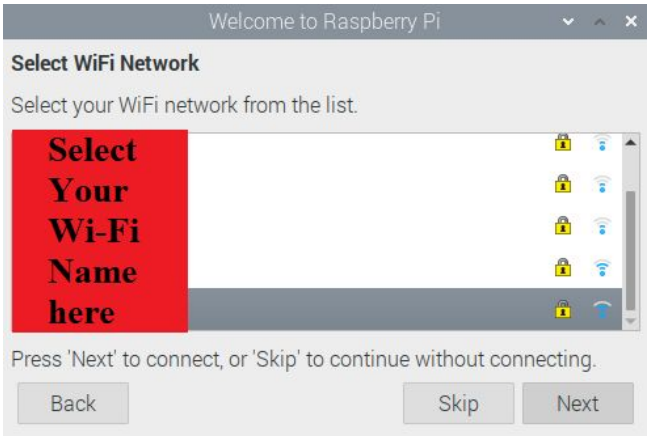

Figure 12. – *Find the local Wi-Fi name on the list where the red box is here.*

Next step is to enter the Wi-Fi passphrase, as seen in figure 13.

Figure 13. – *Enter the password for the Wi-Fi network.*

The wizard will also run an update of the OS, as shown in figure 14. This can also be done manually later, when some days have passed since the last update. This can be done with the ***sudo apt update*** and ***sudo apt upgrade*** commands.

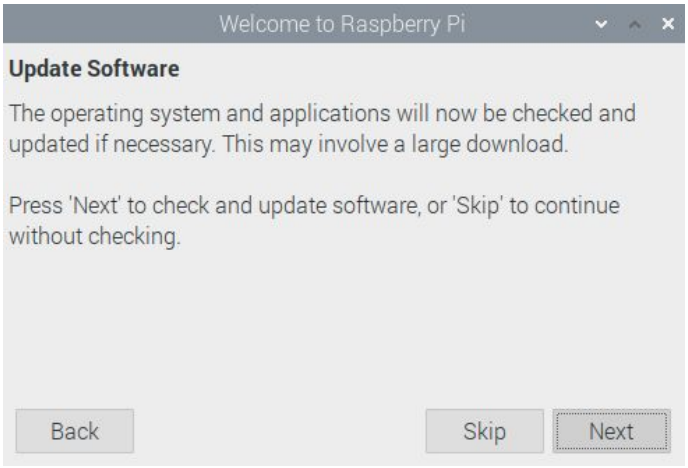

Figure 14. – *Update the OS.*

Once the upgrade is finished, the OS can be rebooted, as shown in figure 15.

Figure 15. – *Restart once updates are finished.*

Setting up dependencies

GitHub Repository

To get the solution to work, there are some dependencies which need to be configured. Start by downloading a copy of the github repository, as seen in figure 16.

Figure 16. – *Download a copy of the project from GitHub.*

Run the installation script with root privileges, as seen in figure 17.

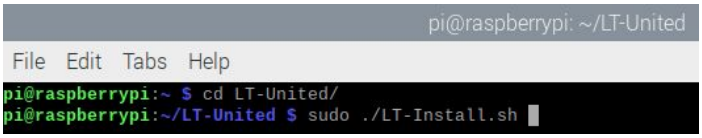

Figure 17. – *Reboot once updates are finished.*

The script will install some software (libffi-dev) needed for calling code written in different programming languages, downgrade the cryptography version (to 2.1.4) for compatibility and add a software interface to Alexa (Flask-ask).

Ngrok

To access the local service running the Python code, Ngrok is used to map it to an external internet URL. To get this software, simply open a web browser and navigate to **www.ngrok.com/download**, and download Ngrok, as seen in figure 18.

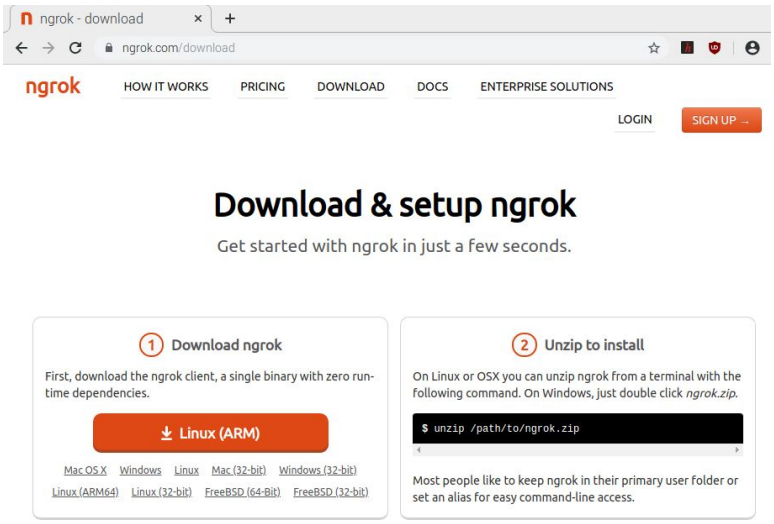

Figure 18. – *Download a copy of ngrok for Linux running on ARM processors.*

Once the download has finished, extract the file from the archive, as seen in figure 19.

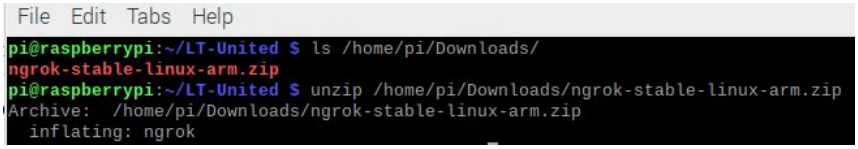

Figure 19. – *Extract the ngrok file from the zip archive.*

Circuit 1 - Raspberry Pi and LED Light

List of components

1. For the connections, it is recommended to use Du-Pont type Male-Female wire connectors, as seen in figure 20.

Figure 20. – *Electrical wires.*

2. 1 x Breadboard. (Used to easily create solderless circuits during prototyping).

3. 1 x Red LED (5mm THT). LED1 on schematics

Figure 21. – *LED Light diode.*

4. 1 x 1kΩ Resistor (THT). R1 on schematics, there are a couple of variations:

Figure 22. – *Resistors.*

5. 1 x Raspberry Pi model 3 or 4 mini computer, as shown in image 23.

Figure 23. – *Raspberry Pi.*

6. 1 x USB Power supply for Raspberry Pi. (Make sure it supplies the recommended power and voltage).

7. 1 x MicroSD card with minimal 16GB capacity. Installed with Rasbian.

8. 1 x USB Keyboard and 1 x USB Mouse. (Used to configure and start the services).

9. 1 x HDMI cable and monitor. (Used to configure and start the services).

Electrical schematic diagram

A logical diagram of the circuit can be seen here in figure 24.

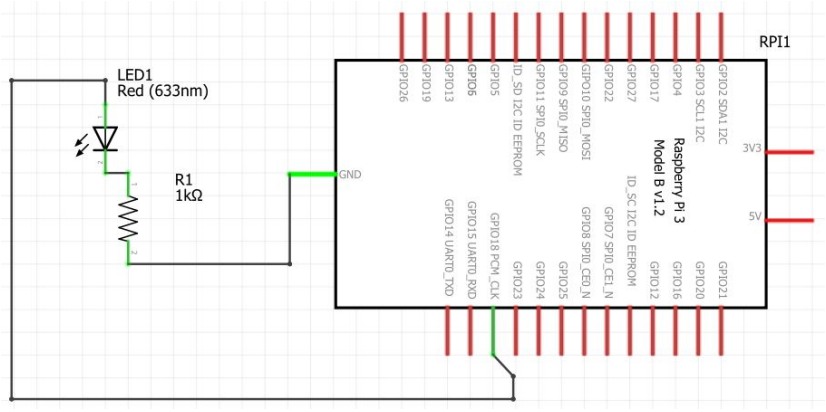

Figure 24. – *Overview of the circuit diagram.*

Breadboard diagram

The physical layout is displayed here in figure 25.

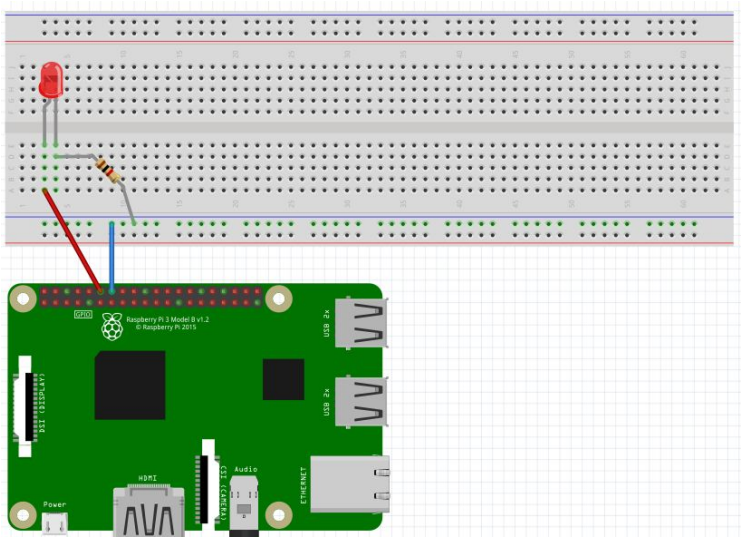

Figure 25. – *Illustration of the physical connection.*

Application 1 - Controlling the LED Light with Alexa

Alexa can be used to control the led light circuit in the previous chapter. To configure and start the application on the Raspberry Pi, there are three steps which need to be carried out.

Starting the Python program

Open a command terminal and navigate to the Code folder. Then start the Python program light.py, as seen in figure 26.

Figure 26. – *Command to start the LED light Python program.*

The service should now be running locally on the Raspberry Pi on port 5001, as seen in figure 27.

Figure 27. – *The local service running on the Raspberry Pi.*

Running Ngrok

To start the external mapping to the local service, open another terminal window and run the ngrok command, as shown in figure 28.

Figure 28. – *Command to start Ngrok.*

Once the program has run for a short time, the output in the terminal window will look similar to what can be seen in figure 29.

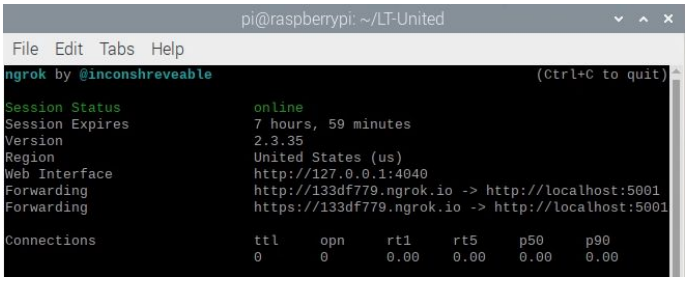

Figure 29. – *Typical Ngrok program output.*

Setting up the Alexa Developer Console

Open a web browser and navigate to https://developer.amazon.com/alexa/console/ask

Create an account if needed and then create a new skill called ***Raspberry Pi***. Use custom settings. Once the name has been set, navigate to JSON and add the code provided in the /JSON/Light.json file in the GitHub Repository.

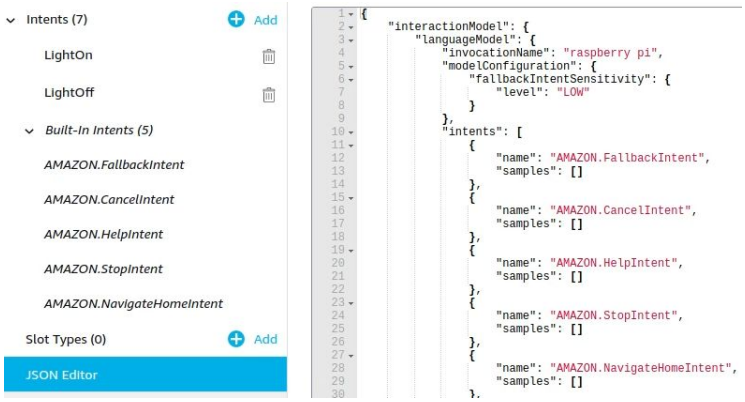

Figure 30. – *The JSON editor.*

The next step is then to update the endpoint with the URL from figure 29. Make sure to use the ***https*** URL, as seen in figure 31.

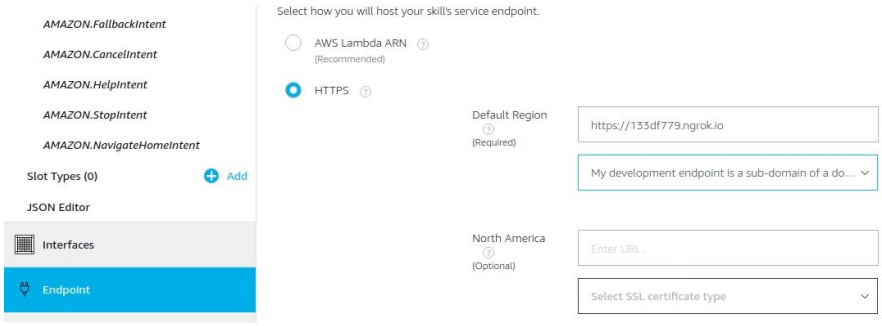

Figure 31. – *Updating the Endpoint with the URL from Ngrok.*

Once the URL is updated, it is essential to remember to save the endpoint. When configuring the solution for the first time, click ***Build Model***, as seen in figure 32, to enable the save button. The endpoint needs to be updated each time the Ngrok program is run.

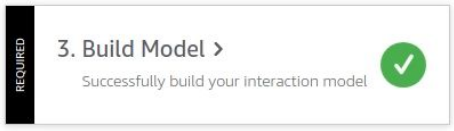

Figure 32. – *The Build Model option in the Amazon developer console.*

The solution should now be finished and can be tested by using an Amazon Echo device, the developer console, or with the Alexa app running on a smartphone.

To trigger the light, say “Alexa, tell Raspberry Pi to turn on the light” or “Alexa, ask Raspberry Pi to turn on light” and similar to turn it off.

Circuit 2 - Alexa and LCD display

List of electrical and hardware components

1. Wires. For Breadboard connections, it is recommended to use Du-Pont type Female-Female wire connectors.

Figure 33 . – *Electrical wires.*

2. 1 x LCD 1602 I2C. LCD1 on schematics

Figure 34. – *LCD display.*

3. 1 x Raspberry Pi model 3 or 4 mini computer. RPI1 on schematics

Figure 35. – *Raspberry pi.*

4. 1 x USB Power supply for Raspberry Pi.
5. 1 x MicroSD card with minimal 16GB capacity.
6. 1 x USB Keyboard. It will be used for the first time Operating system and software installation.
7. 1 x USB Mouse. It will be used for the first time Operating system and software installation.

Electrical schematic diagram

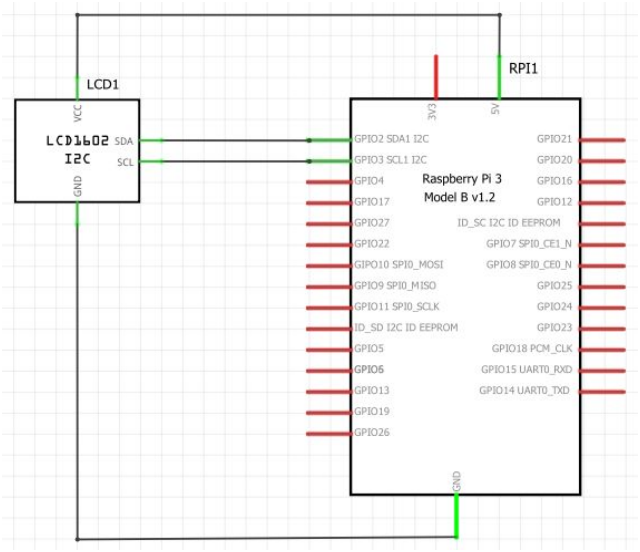

Figure 36. – *Overview of the circuit diagram.*

Physical diagram

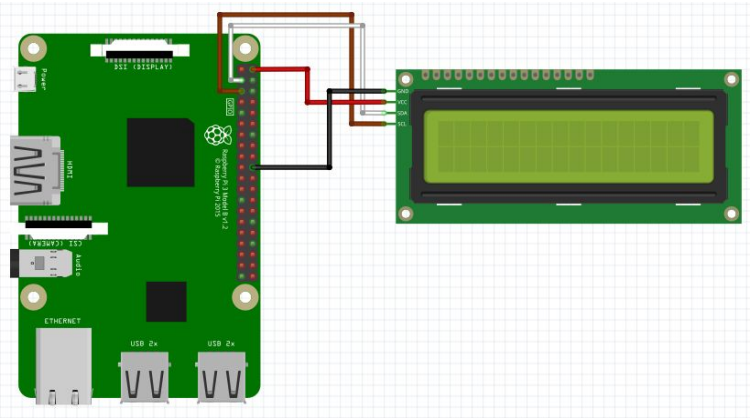

Figure 37. – *Illustration of the physical connection.*

Application 2 - Pomodoro timer with Alexa, Raspberry Pi and LCD Display

In this guide, you will learn how to connect and configure Alexa to communicate with your 16x2 Character l2c LCD Display. This application uses a python library created by “The Raspberry Pi Guy”. All his code you can find at his GitHub repository https://github.com/the-raspberry-pi-guy/lcd. You can explore his code to learn how to use LCD with the Raspberry Pi.

LCD display setup

First, we need to clone the LCD drivers repository from GitHub. For this we open terminal in raspberry pi and run this command:

$ git clone https://github.com/the-raspberry-pi-guy/lcd

Figure 38. – *git clone command.*

After cloning the repository, we navigate to the repository in our disk. For this we use command:

$ cd lcd

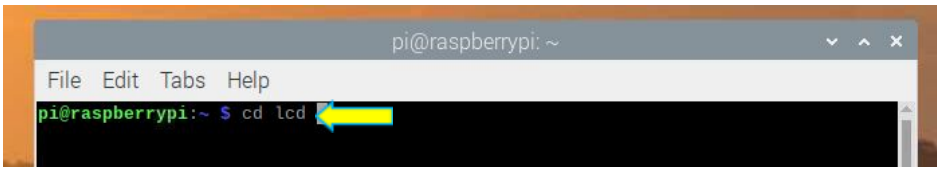

Figure 39. – *navigation using the command line.*

Inside the folder, we need to run the installation script. This will install all the drivers needed to use the LCD display.

$ sudo sh install.sh

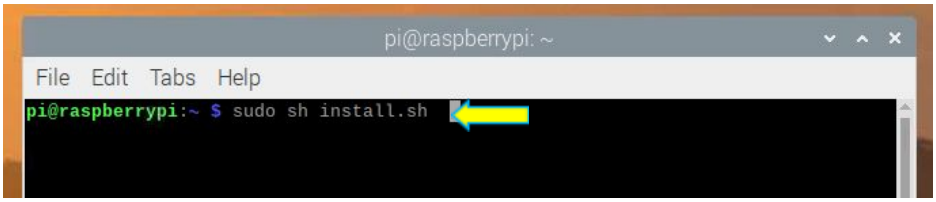

Figure 40. – *launching script.*

When the installation is finished, we need to copy our program code from our repository. For this we use the command:

$ wget https://raw.githubusercontent.com/KarlTrollsaas/LT-United/master/Code/pomodoro.py

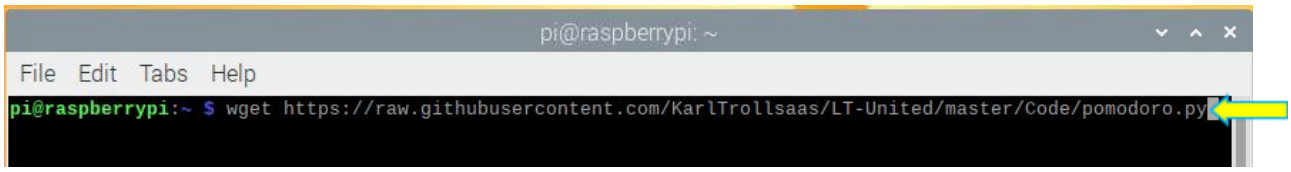

Figure 41. – *downloading pomodoro.py.*

LCD display setup is finished, and now we can run it on the raspberry pi.

First we start Ngrok. It provides a web link to our raspberry pi and allows us to connect to it remotely. (How to setup Ngrok you will find on the Installation section)

$ ngrok http 5001

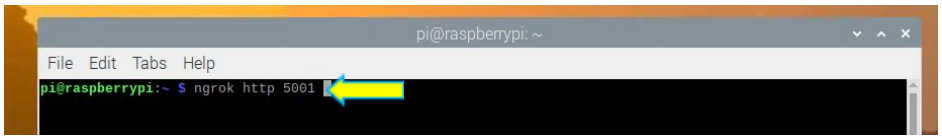

Figure 42. – *starting Ngrok server.*

After running this command you should see a window like this:

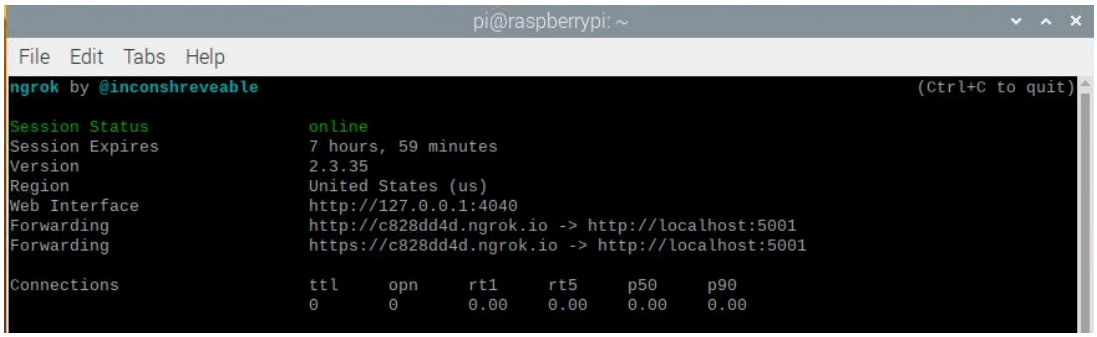

Figure 43. – *Ngrok interface.*

Next, we run the Pomodoro program. Open a new terminal window. Navigate to lcd directory and run python program:

$ python pomodoro.py

It should give a message that it's running.

Figure 44. – *starting pomodoro.py.*

Alexa configuration

We need to teach Alexa how to communicate with our program. For this, we will create a new skill.

In your browser, navigate to https://developer.amazon.com/en-US/alexa/ . Login to your developer's account.

Navigate to Skill Builders -> Developers Console.

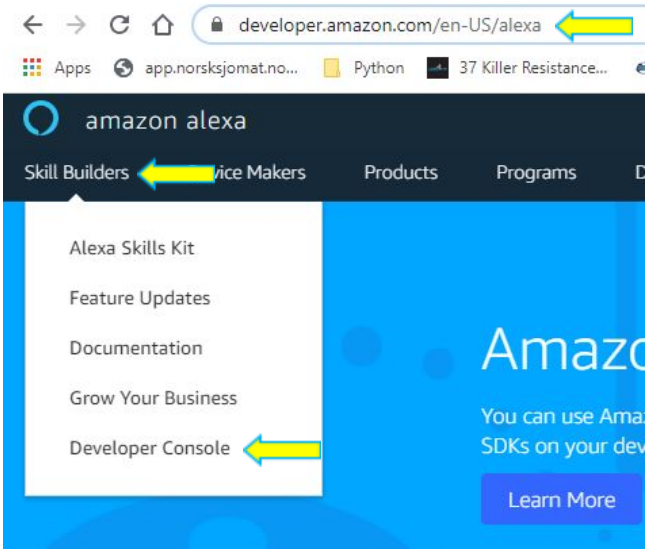

Figure 45. – *navigation to Developer Console.*

This will open Alexa skills windows. Press the Create Skill button.

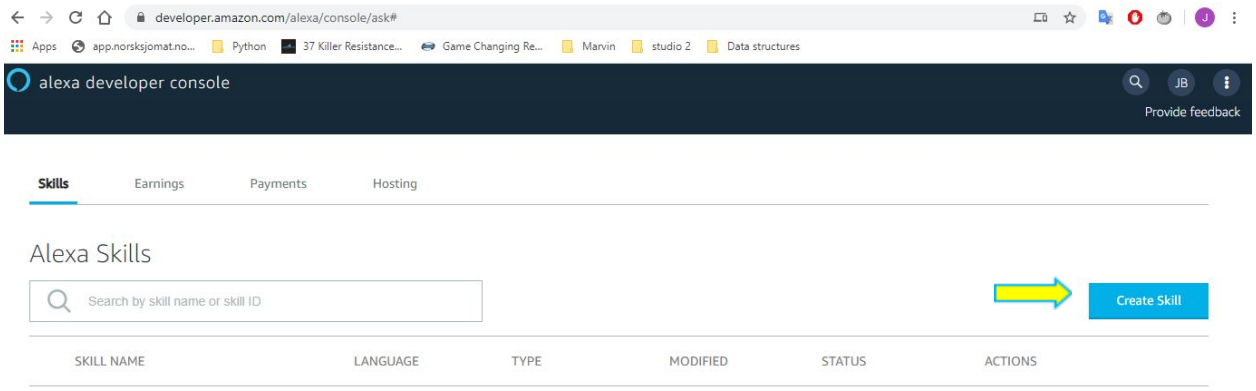

Figure 46. – *Creating new Alexa skill step 1.*

Inside Create a new skill window we give a name to our skill. In this case we call it "Pomodoro", but you can call it differently. It will not change how a program behaves.

For "1. Choose a model to add to your skill" we select "Custom". And for "2. Choose a method to host your skill's backend resources" we select "Provision our own".

Finally we press "Create skill" Button.

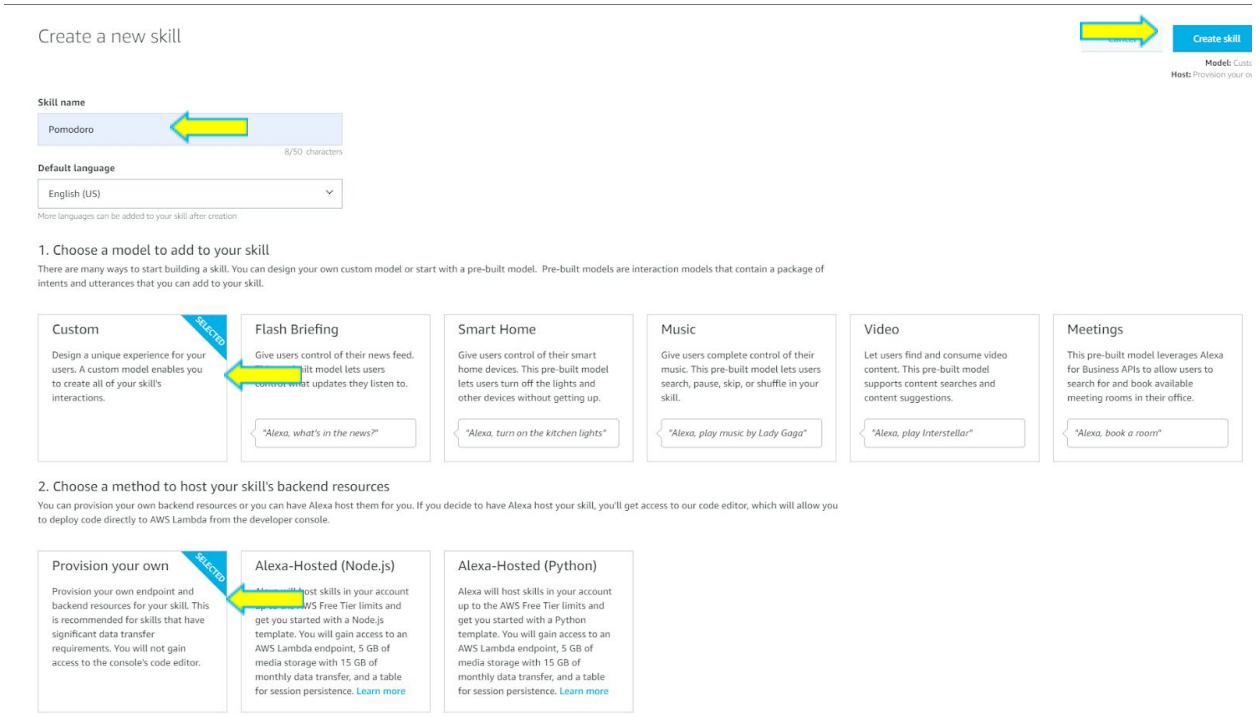

Figure 47. – *Creating new Alexa skill step 2 .*

Now we have to choose a template. For this project, we will "Start from Scratch"

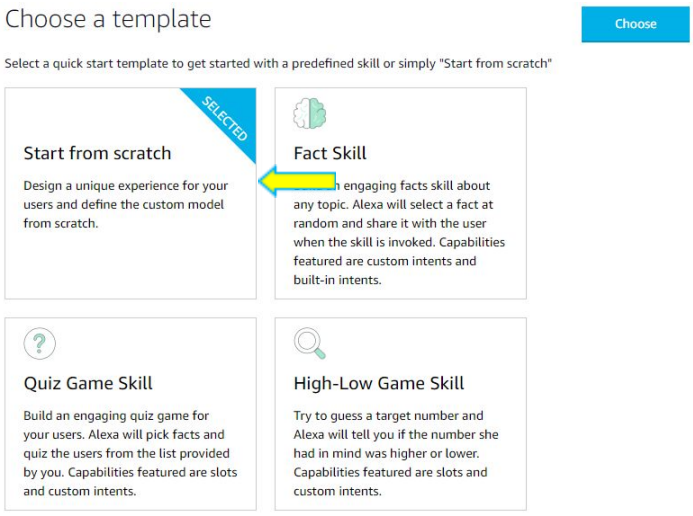

Figure 48. – *Creating new Alexa skill step 3.*

Now we are at the skill build page.

The first step is to set up a phrase to activate our skill. We select "Invocation" and in the field "Skill Invocation Name" we write "pomodoro display". You can change the phrase to a different one if it works better for you.

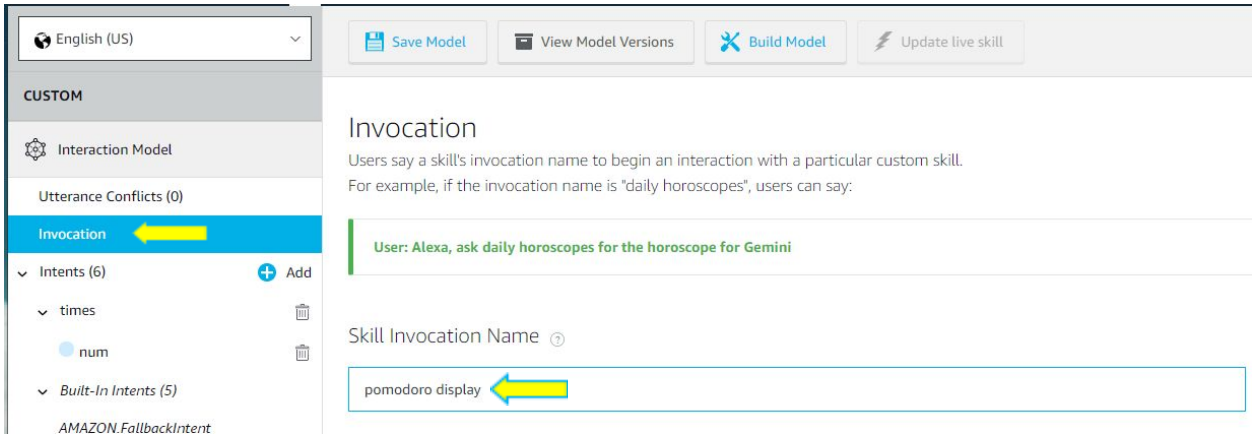

Figure 49. – *Configuring Invocation.*

Now we create an intent which will trigger a specific function in our Python code.

Press the "Add" button and fill up the "Create custom intent" field with the word "times" (This word has to be exactly as it is our function name. If you want you can change it, but don't forget to change it in the code.). Press the "Create custom intent" button.

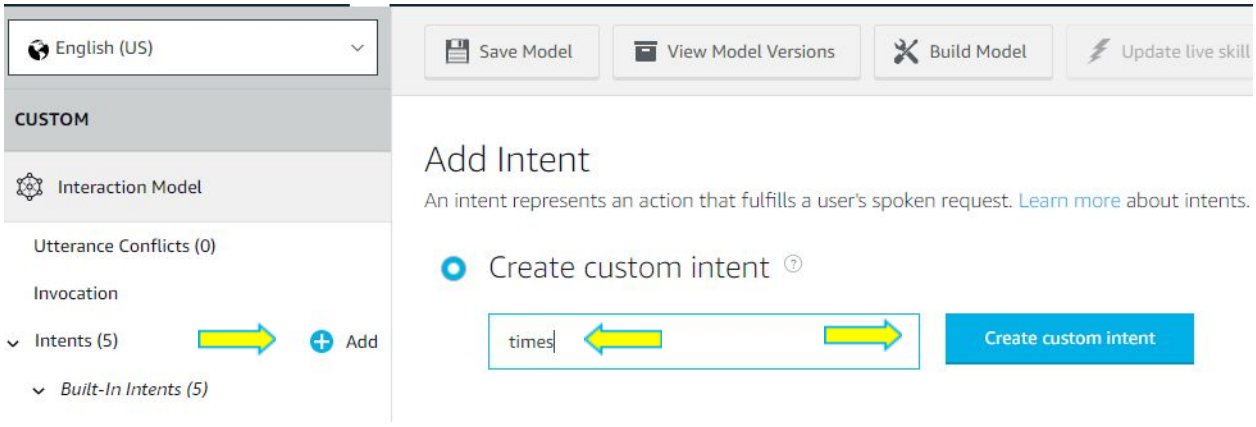

Figure 50. – *Creating new intent.*

Now, as we have created an intent to activate our function, we need to teach Alexa hot to respond to it. The Pomodoro program will need to get an answer to a question "How many sessions do you want to have?". And for this to work we have to send an integer.

In the field "Sample Utterances", we add a couple phrases for Alexa to recognize. It is important that all of them had a keyword marked with the curly brackets. Here we use the "{num}" keyword as it is a parameter name that our function expects to get. This parameter has to be an integer. Because of that in "Intent Slots" for parameter num we set a type "AMAZON.NUMBER". Recommended phrases "{num} sessions", "{num}", "{num} times".

Now we can Build our Model.

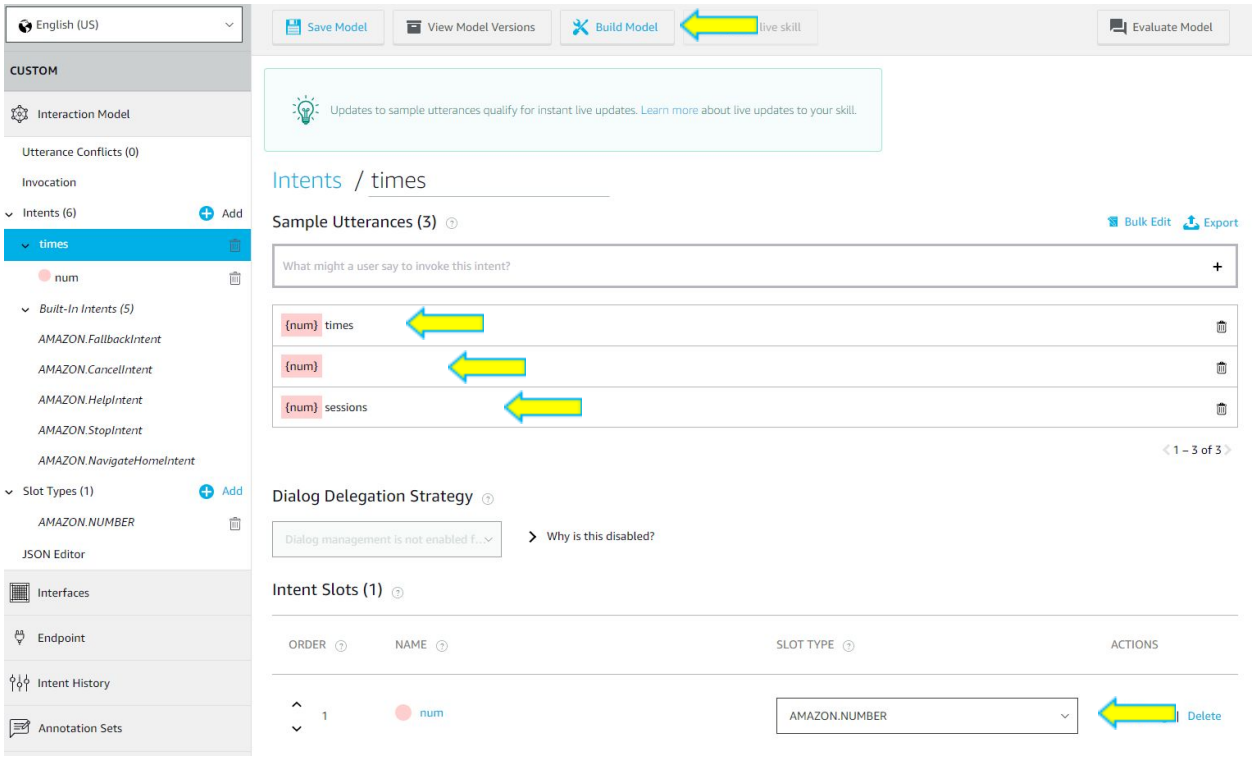

Figure 51. – *Adding phrases to an intent.*

Now we have to set up communication with our Raspberry Pi.

In the menu we select “Endpoint”. Then we select “HTTPS” for Service Endpoint Type.

In the field for Default Region we put our web link from Ngrok and select “My development endpoint is a sub-domain of a domain that has a wildcard certificate from a certificate authority”.

We Save Endpoints.

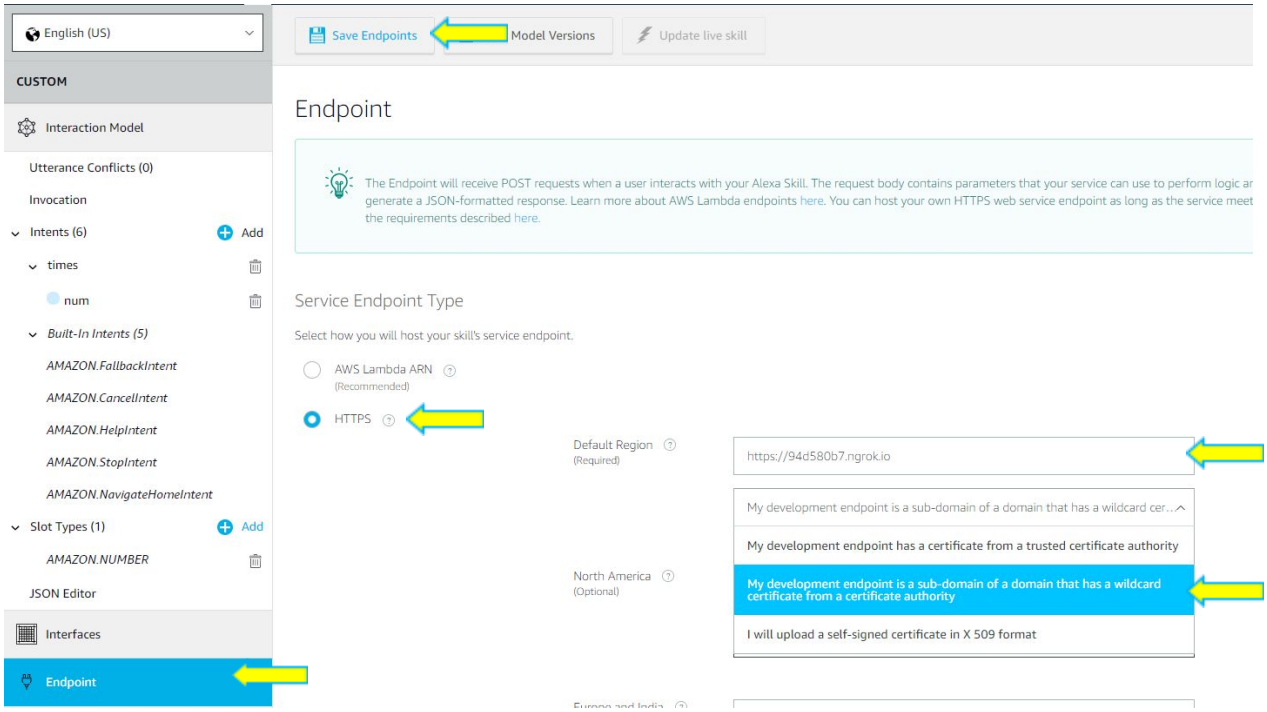

Figure 52. – *configuring the endpoint.*

This step you have to repeat every time you restart Ngrok as it will generate new web links. The necessary link starts with https.

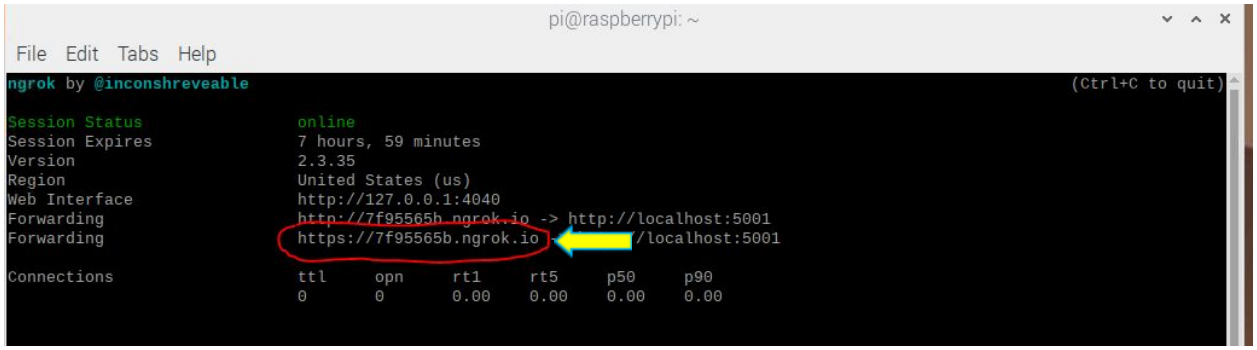

Figure 53. – *Ngrok web link to a Raspberry Pi.*

If you set everything correctly, the new skill should start working.

You can try to activate it using an Alexa enabled device which is connected to the same account as your amazon developer account.

Activation phrase is “Alexa start pomodoro display”. You should get a response phrase “How many sessions do you want to have?”. Your answer should contain a number and “sessions” word.

The counter on the display should start.

Bonus - Weather information on LCD display

This guide will help you to set up an Alexa connection to Raspberry Pi for presenting weather information on the display.

First, we need to get our API key from https://home.openweathermap.org/api_keys. (You have to create an account.)

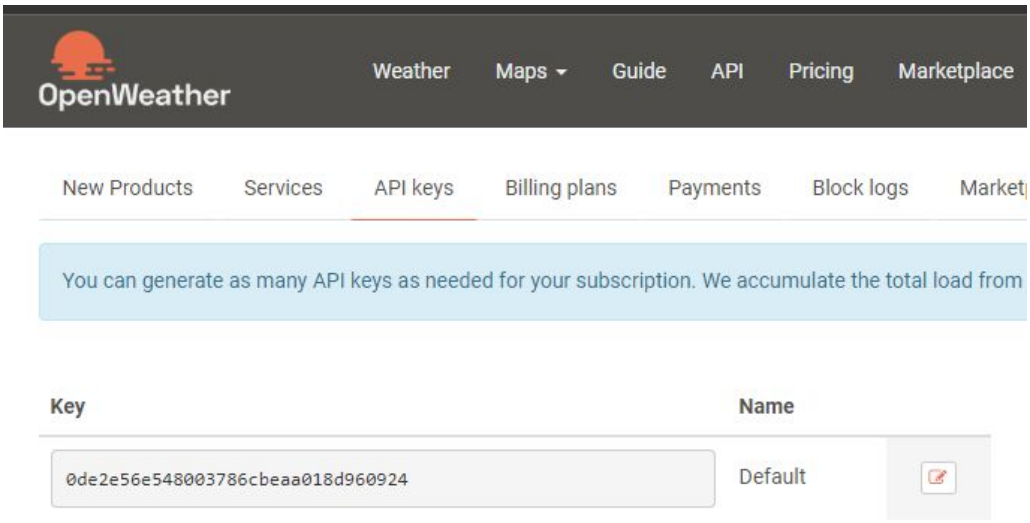

Figure 54. – *API key.*

Copy weather.py file into lcd directory.

$ wget https://raw.githubusercontent.com/KarlTrollsaas/LT-United/master/Code/weather.py

Open the file in the editor and insert your api key as a string.

weather.py *

```
@app.route('/')
def homepage():
    return "Hi"

#starting question when he skill is trigered by Alexa
@ask.launch
def start_skill():
    mes = "Tell me the city name for the wether information"
    return question(mes)



@ask.intent('weather')
def weather(city_name):
    display.lcd_clear()

    api_key = "your api key"

    base_url = "http://api.openweathermap.org/data/2.5/weather?"

    complete_url = base_url + "q=" + city_name + "&units=metric" + "&appid=" + api_key

    response = requests.get(complete_url)

    x = response.json()

    if x["cod"] != "404":

        # store the value of "main"
        # key in variable y
```

Figure 55. – *Code editing.*

Start Ngrok in one terminal and waether.py in the second terminal.

Now create a new Alexa skill and upload weather.json file from github https://github.com/KarlTrollsaas/LT-United/blob/master/Code/weather.py.

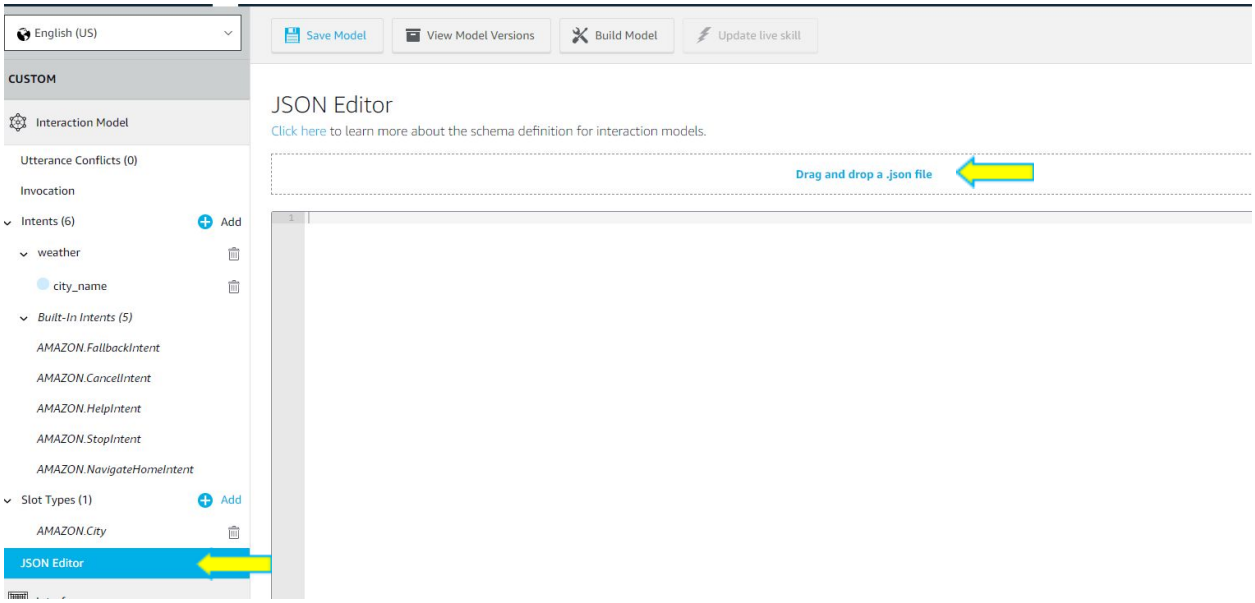

Figure 56. – *Uploading json file.*

Build the model and your skill should start working. To activate the program just use the phrase “weather buddy”. Have fun!

Circuit 3 - Using Alexa to control a servo motor connected to the Raspberry Pi

List of electrical and hardware components:

1. Wires. For Breadboard connections, it is recommended to have Du-Pont type Male-Female wire connectors.

Figure 57. – *Electrical wires.*

2. 1 x Servo motor SG90. SERVO1 on schematics

Figure 58. – *Servo motor.*

3. 1 x Raspberry Pi model 3 or 4 mini computer. RPI1 on schematics

Figure 59. – *Raspberry Pi.*

4. 1 x USB Power supply for Raspberry Pi.
5. 1 x MicroSD card with minimal 16GB capacity.
6. 1 x USB Keyboard. It will be used for the first time Operating system and software installation.
7. 1 x USB Mouse. It will be used for the first time Operating system and software installation.

Electrical schematic diagram

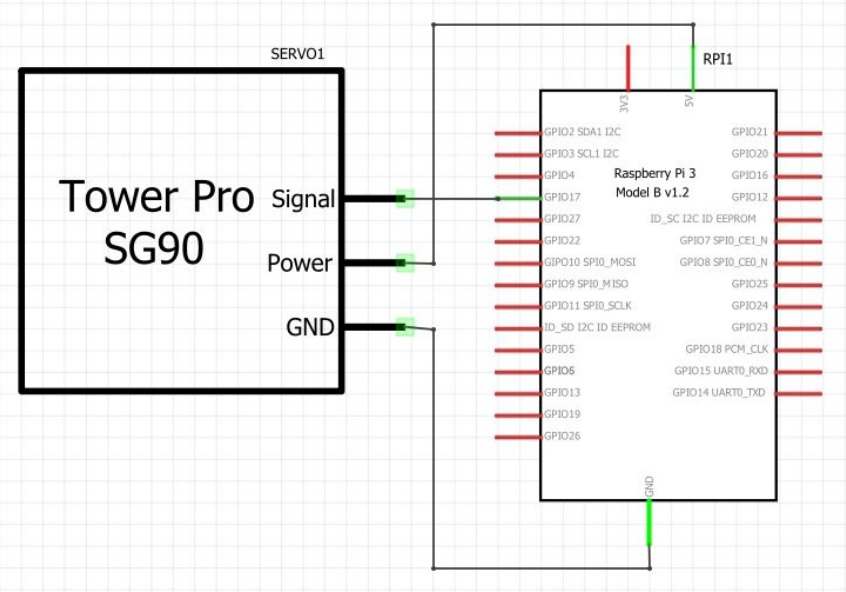

Figure 60. – *Overview of the circuit diagram.*

Breadboard diagram

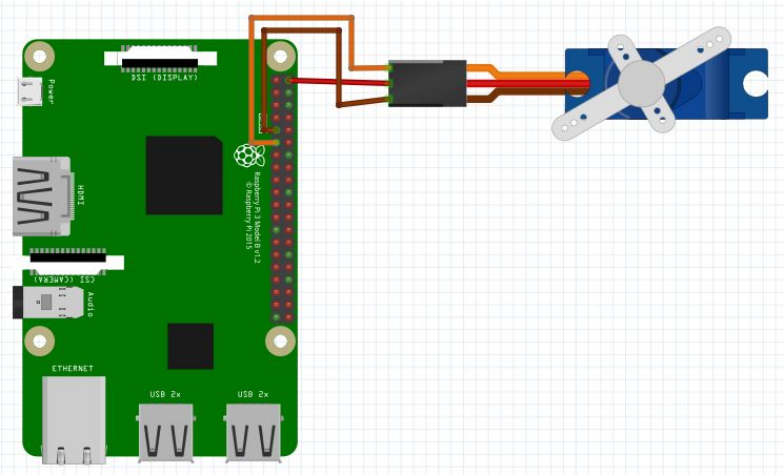

Figure61 . – *Illustration of the physical connection.*

Application 3 - Pet feeder, using Alexa to give a pet food

The servo circuit can be used in various practical applications, such as a pet food dispenser. Parts for this application can be crated with a 3D printer by using the files from another DIY project, available at https://www.thingiverse.com/thing:3761936

There is also a video showing how to assemble the parts https://youtu.be/ulrTSs5H_fA

Circuit 4 - Using Alexa to control all the devices connected to the Raspberry Pi

The whole Alexa Raspberry Pi solution is a versatile base and is compatible to use either one component as LED/LCD/Servo or all in one. Below is the All-in-One solution as an option to have all components connected.

List of electrical and hardware components

1. Wires. For Breadboard connections, it is recommended to have Du-Pont type Male-Female wire connectors.

Figure 62. – *Electrical wires.*

2. 1 x Breadboard.

3. 1 x Red LED (5mm THT). LED1 on schematics

Figure 63. – *LED Light diode.*

4. 1 x 1kΩ Resistor (THT). R1 on schematics
   Variations:

Figure 64. – *Resistors.*

5. 1 x LCD 1602 I2C. LCD1 on schematics

Figure 65. – *LCD display.*

6. 1 x Servo motor SG90. SERVO1 on schematics

Figure 66. – *Servo motor.*

7. 1 x Raspberry Pi model 3 or 4 mini computer. RPI1 on schematics

Figure 67. – *Raspberry Pi.*

8. 1 x USB Power supply for Raspberry Pi.

9. 1 x MicroSD card with minimal 16GB capacity.

10. 1 x USB Keyboard. It will be used for the first time Operating system and software installation.

11. 1 x USB Mouse. It will be used for the first time Operating system and software installation.

Electrical schematic diagram

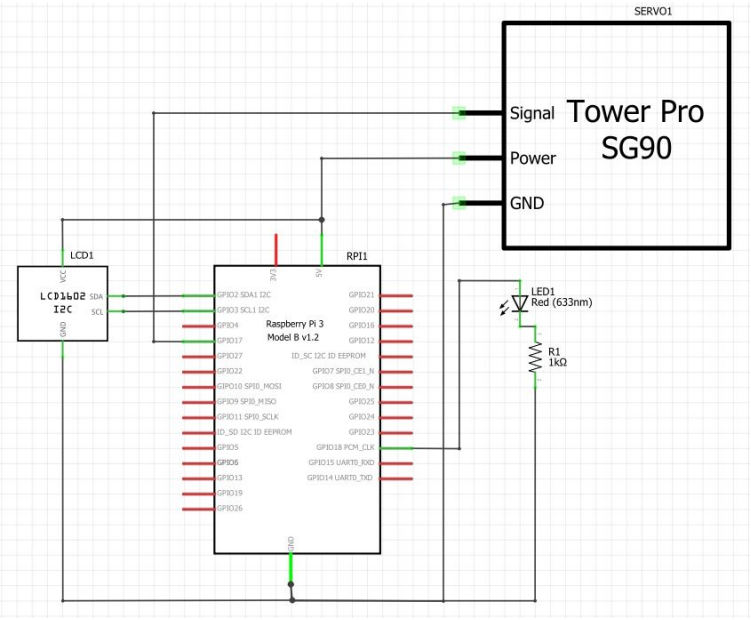

Figure 68. – *Overview of the circuit diagram.*

Breadboard diagram

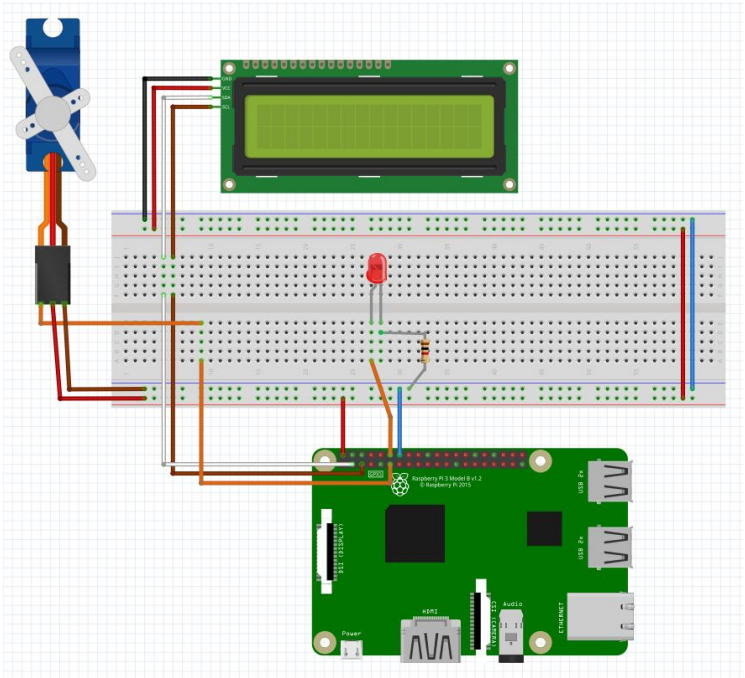

Figure 69. – *Illustration of the physical connection.*