

Vodka powered by Żubrberry Pi



mgr inż. Michał Gałka

michal.galka@harman.com

dr inż. Maciej Polańczyk

maciej.polanczyk@stxnext.pl

Kim jesteśmy

Michał Gałka

Kontakt:

- michal@galka.cc
- [@michal_galka](https://twitter.com/michal_galka) (Twitter)
- **None** (Facebook)

Zawodowo:

- Principal Software Engineer @Harman
(C, Python, Linux, Embedded)
- Trener @Sages (Python)



Kim jesteśmy

Maciej Polańczyk

Kontakt:

- maciej.polanczyk@gmail.com

Zawodowo:

- Python Developer @STXNext

PyCaribbean



Idealny asystent

- zawsze słucha
- podaje alkohol

Idealny asystent

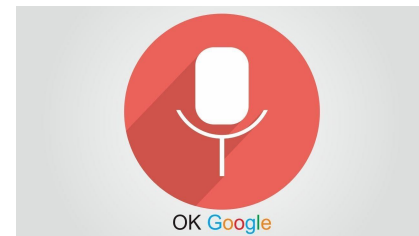
Asystent który zawsze słucha?

Dostępne rozwiązania

- Amazon Alexa
- Apple Siri
- Google Assistant



amazon alexa

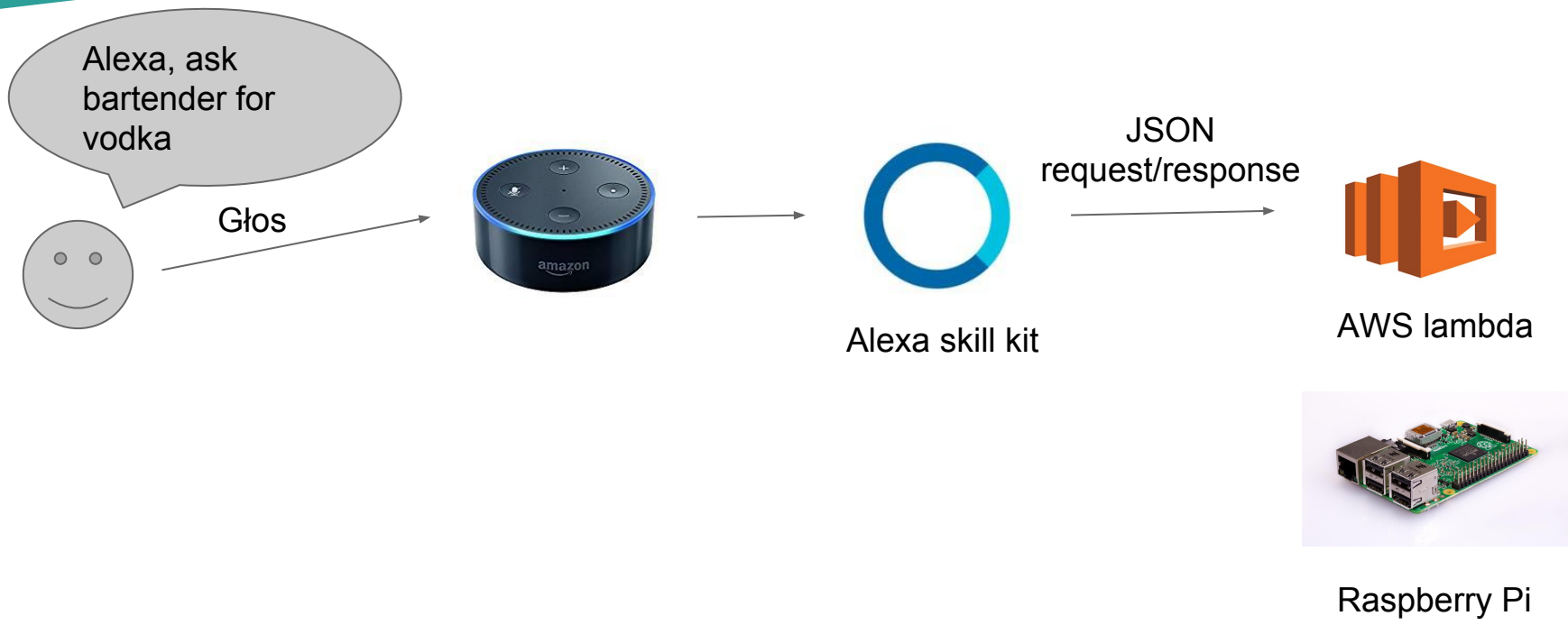


Wybraliśmy



amazon alexa

Alexa - Flow chart



Alexa - Requests

Alexa,

bartender

start bartender

launch bartender

open bartender

...



/

Alexa - Requests

Alexa, **ask**

bartender to pour

vodka

use

tell

...



/pour_vodka

Alexa - Responses

Alexa, ask bartender to pour vodka ->

<- vodka is ready

Alexa, ask bartender to pour whisky ->

<- whisky is ready

Alexa - Responses

Alexa, ask bartender to pour vodka ->

<- with juice?

no, I'm from Poland ->

<- vodka is ready

Nauka polewania - Amazon

- założenie konta w <https://developer.amazon.com/>
- dodanie nowego skill'a
- skonfigurowanie skill'a
- routing za pomocą ngrok

Nauka polewania - Amazon



Sign in

Email (phone for mobile accounts)

Password

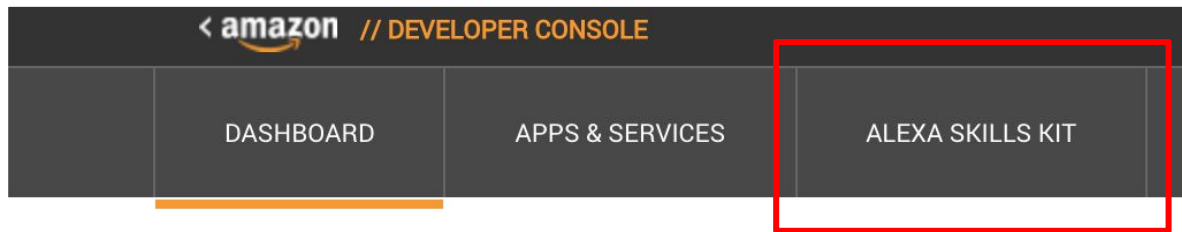
[Forgot your password?](#)

Sign in

— New to Amazon Developer? —

Create your Amazon Developer account

Nauka polewania - Amazon



Notifications

All	Critical
No Notifications.	

Nauka polewania - Amazon

Create Skill

Nauka polewania - Amazon

Create a new skill

SkillName

Nauka polewania - Amazon

Create a new skill

bartender

Nauka polewania - Amazon



Custom

Design a unique experience for your users. A custom model enables you to create all of your skills interactions.

Unselect

Selected

Nauka polewania - Amazon



Skill builder checklist

Complete these steps to be able to test your skill using the simulator in the test tab, or with your echo device.

REQUIRED

1. Invocation Name >

Enter an invocation name for your skill



Nauka polewania - Amazon

Invocation

Users say a skill's invocation name to begin an interaction with a particular custom skill. For example, if the invocation name is "daily horoscopes", users can say:

User: Alexa, ask daily horoscopes for the horoscope for Gemini

Skill Invocation Name [?](#)

Nauka polewania - Amazon



Skill builder checklist

Complete these steps to be able to test your skill using the simulator in the test tab, or with your echo device.

REQUIRED

1. Invocation Name >

Enter an invocation name for your skill



REQUIRED

2. Intents, Samples, and Slots >

Add at least one intent and one sample utterance



Nauka polewania - Amazon

```
{
  "interactionModel": {
    "languageModel": {
      "invocationName": "bartender",
      "intents": [
        {
          "name": "AMAZON.CancelIntent",
          "samples": []
        },
        {
          "name": "AMAZON.HelpIntent",
          "samples": []
        },
        {
          "name": "AMAZON.StopIntent",
          "samples": []
        }
      ],
      "types": []
    },
    "name": "pour_vodka",
    "slots": [],
    "samples": [
      "vodka",
      "pour me vodka",
      "pour me a vodka",
      "pour vodka"
    ],
    "types": []
  }
}
```

Nauka polewania - Amazon

REQUIRED

1. Invocation Name >

Enter an invocation name for your skill



REQUIRED

2. Intents, Samples, and Slots >

Add at least one intent and one sample utterance







REQUIRED

3. Build Model >

Successfully build your interaction model



Nauka polewania - Amazon

- REQUIRED** **1. Invocation Name >**
Enter an invocation name for your skill 
- REQUIRED** **2. Intents, Samples, and Slots >**
Add at least one intent and one sample utterance 
- REQUIRED** **3. Build Model >**
Successfully build your interaction model 
- REQUIRED** **4. Endpoint >**
Set a web service endpoint to handle skill requests 

Nauka polewania - Amazon

HTTPS ?

Default Region ?
(Required)

https://77cb89e4.ngrok.io

My development endpoint is a sub-domain of a domain that has a wildc...▼

Nauka polewania - Amazon

The screenshot shows the Amazon Alexa Developer Console interface for testing a skill named "bartender". The top navigation bar includes "Your Skills", "Build", "Test" (which is the active tab), "Launch", and "Measure". Below the navigation bar, there is a toggle switch for "Test is enabled for this skill" which is turned on, and a "Skill I/O" section with a checkmark. The main content area has three tabs: "Alexa Simulator" (selected), "Manual JSON", and "Voice & Tone". Under the "Alexa Simulator" tab, there is a language dropdown menu set to "English (US)", a text input field with the placeholder "Type or click and hold the mic" and a microphone icon, and a large empty text area for the response. On the right side, there is a "Skill I/O" panel with a "JSON Input" section.

amazon alexa

< Your Skills bartender Build Test Launch Measure

Test is enabled for this skill Skill I/O

Alexa Simulator Manual JSON Voice & Tone

English (US) v Type or click and hold the mic

Skill I/O

JSON Input

Nauka polewania - server

- server oparty o Flask - Ask

<https://github.com/johnwheeler/flask-ask>

```
pip install Flask-Ask certifi
```

Nauka polewania - server

bartender.py:

```
from flask import Flask
```

```
from flask_ask import Ask, statement, question, session
```

```
app = Flask(__name__)
```

```
ask = Ask(app, '/')
```

```
@app.route('/')
```

```
def homepage():
```

```
    return 'hi there, how ya doin?'
```

Nauka polewania - server

```
@ask.launch
```

```
def start_skill():
```

```
    message = 'What kind of drink would you like?'
```

```
    return question(message)
```

```
@ask.intent('pour_vodka')
```

```
def vodka_intent():
```

```
    message = 'Vodka is ready'
```

```
    return statement(message)
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

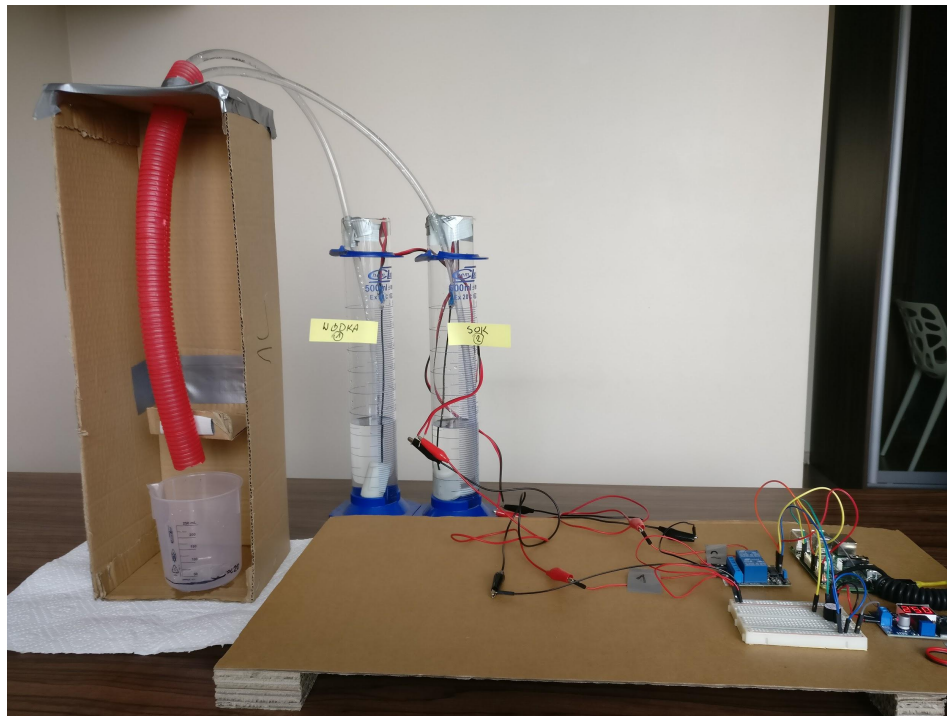

Nauka polewania - server

Uruchamiamy server: `python bartender.py`

Wódkopolewacz 2000

Skład:

- Tektura - 80%
- Srebrna taśma - 17%
- Drobną elektroniką - 2%
- Raspberry Pi - 1%

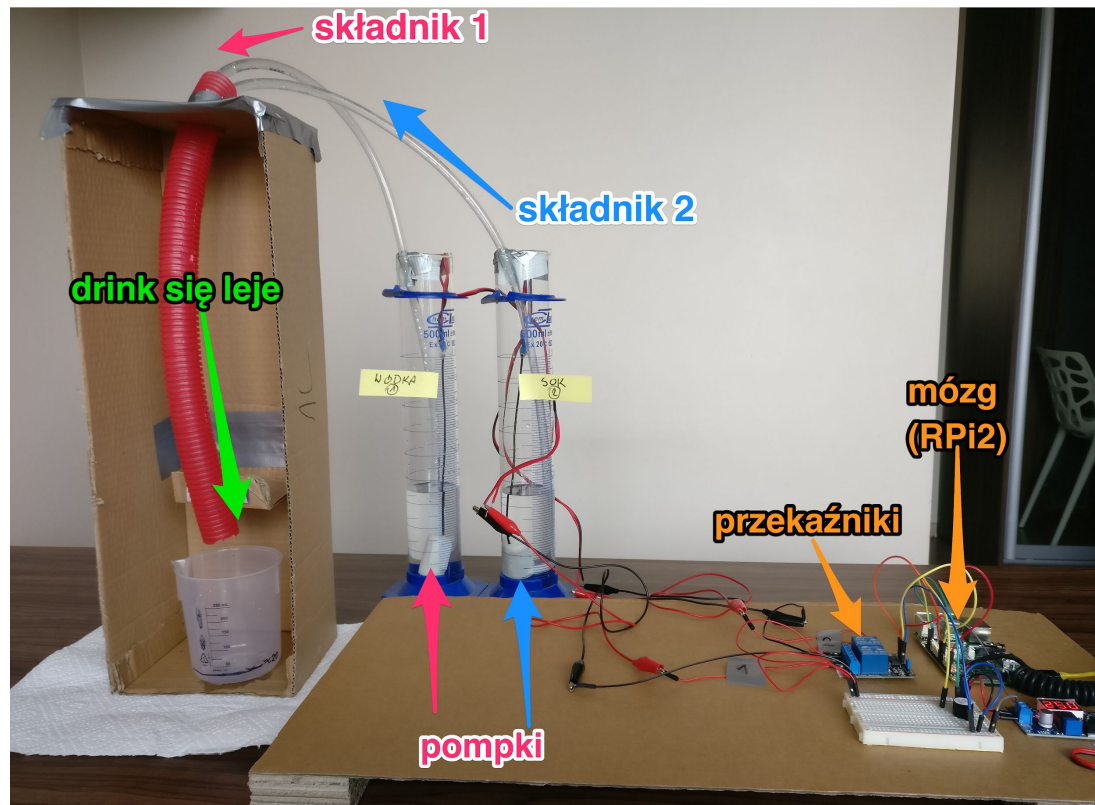


Nauka polewania - Raspberry pi

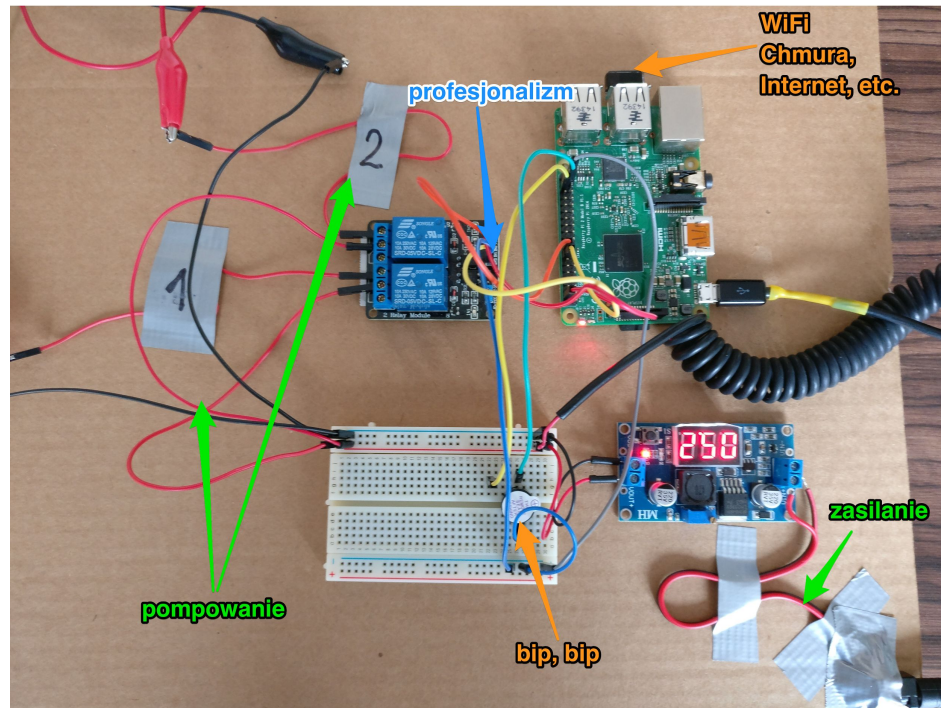
- Mózg wódkopolewacza
- Raspberry Pi2
 - Flask + Flask Ask
 - RPi.GPIO



Wódkopolewacz 2000



Wódkopolewacz 2000



Nauka polewania - sterowanie

```
import RPi.GPIO as GPIO
```

```
class Dispenser:
```

```
    def __init__(self, relay, time_per_10ml=1.4):
```

```
        self.relay = relay
```

```
        self.time_per_10ml = time_per_10ml
```

```
    def start_pouring(self):
```

```
        GPIO.output(self.relay, GPIO.LOW)
```

```
    def stop_pouring(self):
```

```
        GPIO.output(self.relay, GPIO.HIGH)
```

Nauka polewania - sterowanie

```
def pour_time(self, seconds):
```

```
    self.start_pouring()
```

```
    time.sleep(seconds)
```

```
    self.stop_pouring()
```

```
def pour_amount(self, ml):
```

```
    seconds = (self.time_per_10ml * ml) / 10
```

```
    self.pour_time(seconds)
```

Nauka polewania - sterowanie

```
class Drinkomatic:
```

```
    ...  
    def _init_rpi(self):  
        GPIO.setmode(GPIO.BOARD)  
        for dispenser in self.dispensers:  
            GPIO.setup(dispenser.relay, GPIO.OUT)  
            GPIO.output(dispenser.relay, GPIO.HIGH)  
        GPIO.setup(self.buzzer.pin, GPIO.OUT)  
        GPIO.output(self.buzzer.pin, GPIO.LOW)  
    ...
```

```
dear_assistant = drinkomatic.Drinkomatic(VODKA_RELAY, JUICE_RELAY, BUZZER_PIN)
```

```
drink_recipe = {
```

```
    0: 20,
```

```
    1: 80
```

```
}
```

```
dear_assistant.prepare_drink(drink_recipe)
```


Testy integracyjne



Demo

“Alexa, my dear assistant”

Demo

“It was great”

Demo

“Alexa, my dear assistant”

Demo

“It was hard”

Dziękujemy

Pytania?

