Integrating Amazon Alexa into Yamaha Robots

The main goal of this project was to allow a person to communicate with a Yamaha robot using an Amazon Alexa enabled device. To accomplish this project, we used an Amazon Echo Dot, an stable internet connection, a Raspberry Pi 3 B+, a Yamaha RCX240 controller, and a Yamaha robot. You will also need a HDMI monitor, as well as a mouse and keyboard to operate the Pi.

The first step of this project is setting up the Raspberry Pi to run a network tunnel, so that the Amazon Alexa Skill can communicate with the Raspberry Pi. Once the Pi is set up with Raspbian, you must open a terminal session and enter the following commands to ensure we have the proper packages and python libraries:

```
sudo apt-get update && sudo apt-get upgrade -y

sudo apt-get install python2.7-dev python-dev python-pip

sudo pip install Flask flask-ask
```

Now that the Pi is ready, we need to set up the network tunnel that will allow us to run our localhost port 127.0.0.1:5000 on a public domain. We used the service of ngrok to accomplish this. For this service, you can pay $60/year to receive a permeant domain, or you can use the free service and receive a temporary domain that lasts for 8 hours every time you run the service. Go to http://ngrok.com/download and download the most recent Linux version to your home directory /home/pi/. Open a terminal session and enter the following command to open your custom network tunnel https://myapp.ngrok.io:

```
sudo ./ngrok http -subdomain myapp 5000
```

Leave this terminal session open in the background while we continue to work. Now it is time for the python code. Open your python IDE an create a new program and give it a name (program.py). Copy and paste this code into your program and save it. Enter the IP of your RCX240 controller in the HOST variable and enter the Telnet command you would like to use into the s.sendall( ). Change the return statement to what you would like Alexa to say as a response when she executes your command. In the @ask.intent( ) line, change ‘YourIntent’ to what you would like to name your Intent in the Alexa Skill Kit.

```
from flask import Flask
from flask_ask import Ask, statement, convert_errors
import RPi.GPIO as GPIO
import logging
import socket
import getpass
import sys
import telnetlib
import re
import socket
import sys
import time

app = Flask(__name__)
ask = Ask(app, '/')

# The remote host (the IP of the RCX controller)
HOST = 'xxx.xxx.xxx.xxx'
PORT = 23   # The same port as used by the server (don’t change)
s = None

logging.getLogger("flask_ask").setLevel(logging.DEBUG)

@ask.intent('YourIntent', mapping={})
def control_robot():
```

```
for res in socket.getaddrinfo(HOST, PORT, socket.AF_UNSPEC,
socket.SOCK_STREAM):
    af, socktype, proto, canonname, sa = res
    try:
        s = socket.socket(af, socktype, proto)
    except socket.error as msg:
        s = None
        continue
    try:
        s.connect(sa)
    except socket.error as msg:
        s.close()
        s = None
        continue
    break
if s is None:
    print 'could not open socket'
    return statement('Connection could not be established')
    sys.exit(1)

data = s.recv(1024)
print 'Received', repr(data)
data = s.recv(1024)
print 'Received', repr(data)

s.sendall('TELNET COMMAND HERE')
```

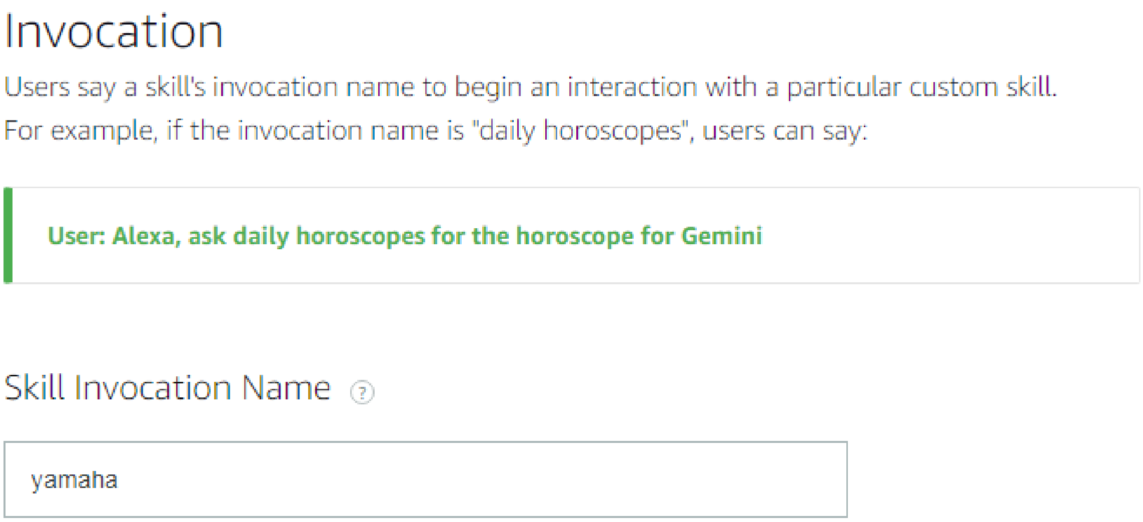
```
    data = s.recv(1024)
    print 'Received', repr(data)

    s.sendall('B')
    s.sendall('Y')
    s.sendall('E')
    s.sendall('\r ')

    time.sleep(1)

    return statement('Return Statement Spoken by Alexa')
if __name__ == '__main__':

    port = 5000
    app.run(host='0.0.0.0',port=port)
```

For a more in depth look at the python code, see our full python code posted at https://www.yrginc.com/integration/?name=Amazon-Echo-Dot-voice-controlled-Yamaha-SCARA-robot-IoT.

Run this program using the following command:

```
sudo python program.py
```

Leave this running in the background while we set up the Alexa Skill. Go to https://developer.amazon.com/ and create an account. Go to the Alexa Skill Kit Developer Console and create a skill. Create a name for the invocation, we used “Yamaha”.



Invocation

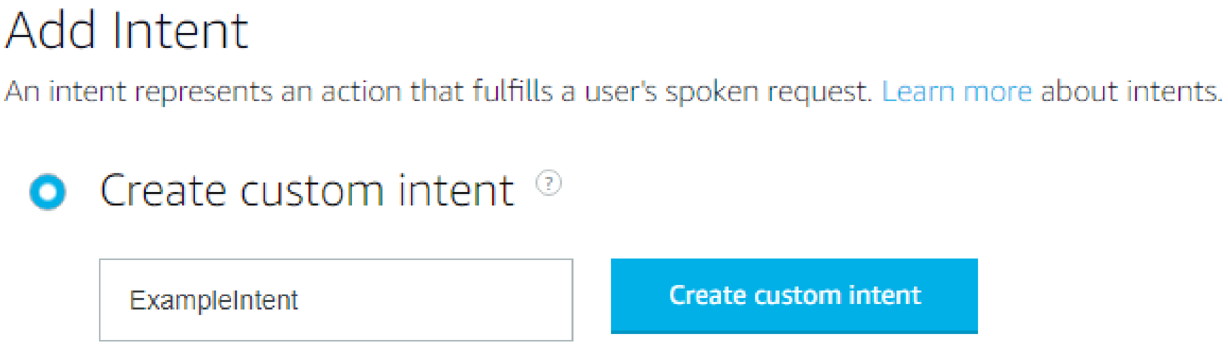
Users say a skill's invocation name to begin an interaction with a particular custom skill. For example, if the invocation name is "daily horoscopes", users can say:

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

Skill Invocation Name ?

yamaha

Now we move on to Intents. Click on the add intent button and give your intent a name related to the task it will trigger.



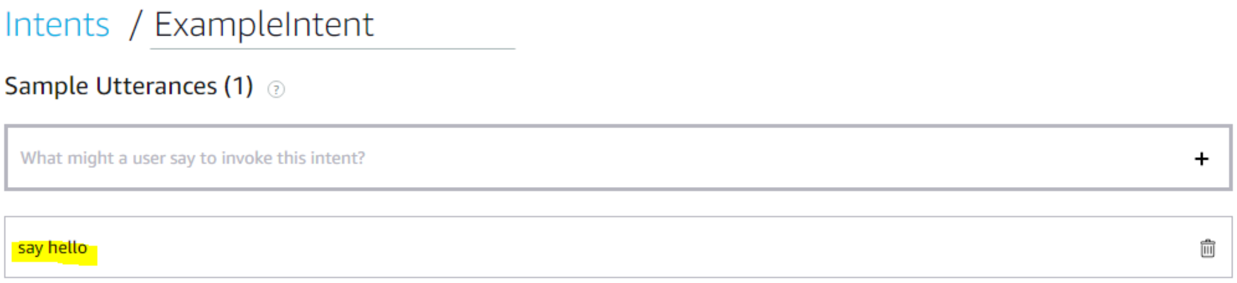
Add Intent

An intent represents an action that fulfills a user's spoken request. Learn more about intents.

● Create custom intent ?

ExampleIntent

Add sample utterances to your intents that you will use to trigger the intent. For example:



Intents / ExampleIntent

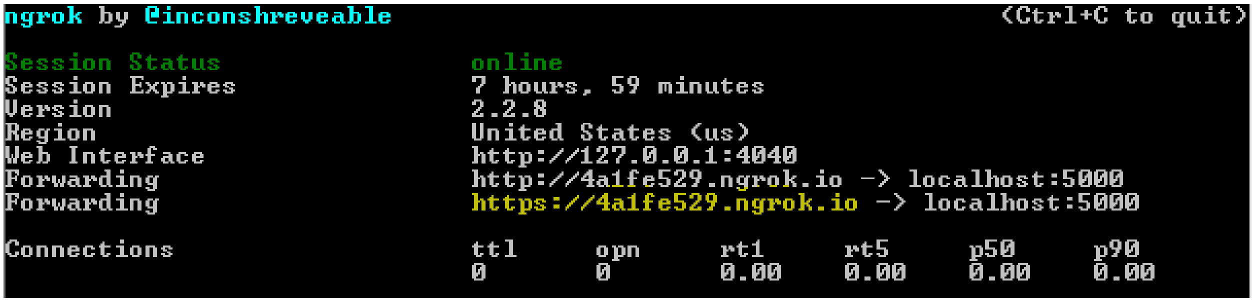
Sample Utterances (1) ?

What might a user say to invoke this intent?

say hello

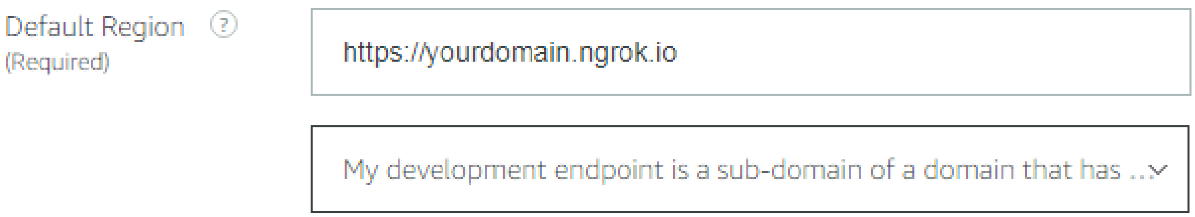
Our invocation word is “Yamaha” and our sample utterance for ‘ExampleIntent’ is “say hello”, so to trigger the ‘ExampleIntent’ intent, you would say “Alexa, tell Yamaha to say hello.” Once your intents are set up, build and save the model. The next step is setting up the endpoint for our

Alexa service, which is our ngrok server that we have running. Ngrok should be displaying something similar to this:



```
ngrok by @inconshreveable                                                  (Ctrl+C to quit)

Session Status                online
Session Expires               7 hours, 59 minutes
Version                       2.2.8
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://4a1fe529.ngrok.io -> localhost:5000
Forwarding                    https://4a1fe529.ngrok.io -> localhost:5000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Under the HTTPS endpoint section on the Alexa Skill Kit, enter the highlighted address into the box and select the option “My development endpoint is a sub-domain of a domain that has a wildcard certificate from a certificate authority.”



Default Region (Required)

https://yourdomain.ngrok.io

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

Now your Alexa Skill is ready to go. You can either test the skill from the “Test” tab on the Alexa Skill Kit, or you can release a Beta version of your skill and test it on your Amazon Echo.

Now that you have ngrok and your python programming running on the same port and your Alexa Skill directing intents to your ngrok server, you should be good to go. If you want to make this permeant, purchase a custom subdomain service from ngrok and run that server. You can also configure the Raspberry Pi to launch both the ngrok server and run the python code upon startup of the Raspberry Pi. Note that the Raspberry Pi must be connected to internet at all times for this to function.