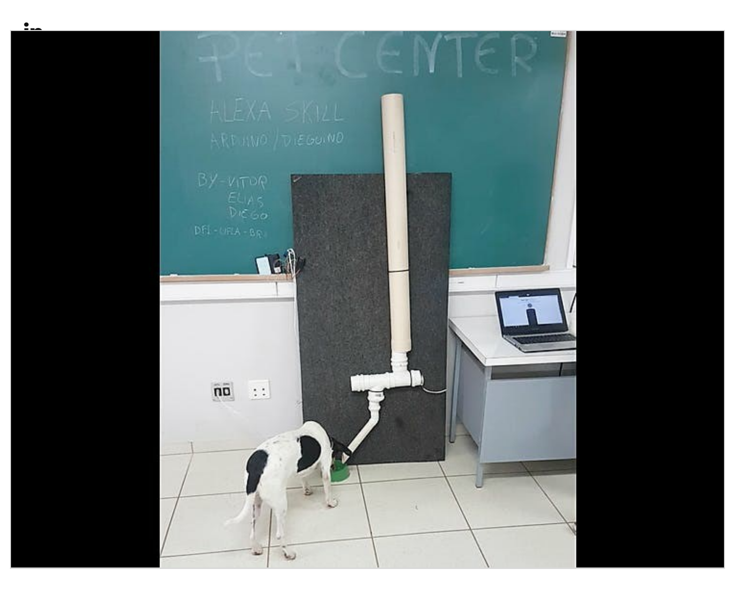
Vítor Barbosa
Published February 25, 2018

Pet Center - Feed and Entertain Your Pet

A simple and effective Alexa-based system to provide food and fun to your dog or cat.

Intermediate · Full instructions provided · 2 days · 4,934



RELATED CHANNELS AND TAGS

internet of things
lasers
mqtt
pets
smart appliances

RELATED PROJECTS

Arduino-Powered Smart Light (Works with Amazon Echo)

DIY Air Humidifier with Backlight Controlled by Alexa

IoT Pet Feeder

Home Automation Using Google Assistant

Voice Activated Media Appliances Using Arduino and Alexa

View more related projects

Things used in this project

Hardware components

Arduino Nano R3	× 1
Espressif ESP8266 ESP-01	× 1
SG90 Micro-servo motor	× 2
Pan/Tilt Camera Module	× 1
RobotGeek Relay	× 1
5mW Red dot laser module	× 1
3.3V Regulator	× 1

Software apps and online services

- Arduino IDE
- Amazon Alexa Alexa Skills Kit
- Google Compute Engine
- Eclipse IoT Broker

Hand tools and fabrication machines

- FT232RL USB-Serial Converter

Story

Intro

When I saw the Alexa and Arduino Smart Home challenge, I thought it was a great opportunity to build something to make people's life better. So I talked to my friends Diego and Elias an we went through a brainstorm to choose the project in which we would spend the next week.

There were lots of ideas of smart plant monitors, lamps and health related things, but we chose to stick with one the would make both people and pet lives a little better, and then Pet Center was born.

Hardware

The electronic parts (see main circuit schematic below) used aren't expensive and can be easily found.

ESP-01 is a bit tricky to program, and you will probably need an USB to Serial converter like FT232RL, and some 10K resistors. The text in orange indicates the corresponding FT232RL pins. These resistors are recommended for trouble-free programing, even though in the main schematic I've included just the ones required to make the chip run after code has been uploaded (GPIO0 and CH_PD).

I know, not the best drawing in the world

Pet entertainment part is done using a camera pan/tilt module, two SG90 servos and a red laser diode. We warn you though, it's much more effective with cats than dogs!

As for the smart pet feeding system, we assembled it with PVC pipes. A handwound steel wire spring and an old microwave oven 120VAC motor do the heavy lifting: as the spring rotates with the motor shaft, food goes down the pipe and reaches the bowl.

The bigger pipe is 75mm wide, and is where pet food is stored

Software

This part was the most difficult, but we ended up developing the endpoint in Python with Flask-Ask. The endpoint had to be always ready to provide a response, and there are many ways to achieve this, but our choice was Google Cloud due to their always free virtual machine instance. We also used localtunnel.me to get a public https url. Code is available below too.

After we got a working endpoint, creating an Alexa skill with the new Skills Kit Console was a nice experience, and no coding was required.

A simple VUI was designed :

Voice User Interface (VUI) Diagram

Alexa, upon user command, makes a request to our endpoint, which provides the correct reply to the user (through Alexa) and publishes a MQTT message to the public broker *iot.eclipse.org* under *petcenter* topic.

ESP-01 is subscribed to this topic under the same broker and relays the message to Arduino Nano, which activates either the entertainment servos or the food serving mechanism, depending on the user request.

See it in action!

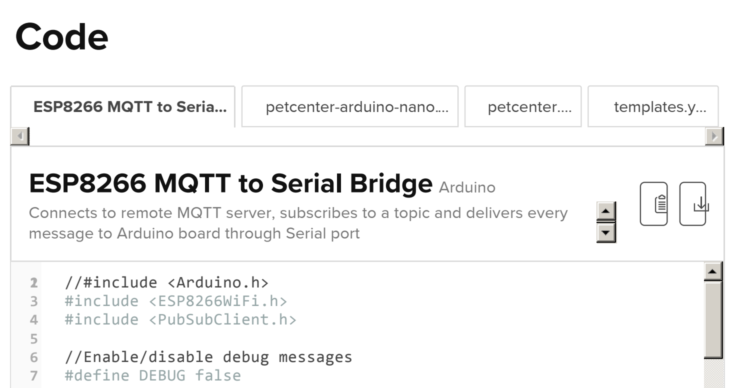
That pretty dog is called Belinha!

Showcase of all skill options

Schematics

Main circuit

View of the main circuit. The AC motor is the pet feeding motor



Code

ESP8266 MQTT to Seria... | petcenter-arduino-nano... | petcenter.... | templates.y...

ESP8266 MQTT to Serial Bridge
Arduino

Connects to remote MQTT server, subscribes to a topic and delivers every message to Arduino board through Serial port

```
//#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

//Enable/disable debug messages
#define DEBUG false


// Update these with values suitable for your network.
const char* ssid = "SSID";
const char* password = "PASS";
const char* mqtt_server = "iot.eclipse.org";
const int mqtt_port = 1883;

//topic to subscribe
const char* inTopic = "petcenter";
//topic to publish
const char* outTopic = "petcenter";

//Serial baud rate
const int baud = 115200;

WiFiClient espClient;
```

Credits

Vítor Barbosa
1 project • 1 follower
Follow
Contact

Comments

Please log in or sign up to comment.