



SMS Texting Pet Food Dish

Created by Tony DiCola



Last updated on 2018-08-22 03:38:54 PM UTC

Guide Contents

Guide Contents	2
Overview	3
Hardware	4
Parts	4
(https://adafru.it/cZx)Wiring	4
Pet Food Dish	6
Amazon SNS Setup	8
Software	11
Future Work	13

Overview

Is checking and filling your pet food dish a constant daily chore? Have you ever wished you could be notified on your phone when the dish is empty? With this project you'll learn how to build a pet food dish that does exactly that by sending an SMS text message when the dish is empty! This is way better than giving your pet a smartphone: our cat MOSFET ran up a huge texting bill and then refused to pay it.

This project is a great example of using an Arduino, [CC3000 Wi-Fi module](http://adafru.it/1469) (<http://adafru.it/1469>), and the [Amazon Simple Notification Service](https://adafru.it/cXo) (<https://adafru.it/cXo>) to send notifications such as text messages. You can even use the knowledge and code from this project to add notification capabilities to your own Arduino projects!

Before you begin it will be helpful to review these guides for more background:

- [Adafruit CC3000 WiFi](https://adafru.it/cRT) (<https://adafru.it/cRT>)
- [IR Sensor](https://adafru.it/cJH) (<https://adafru.it/cJH>)
- [WiFi Candy Bowl Monitor](https://adafru.it/cVI) (<https://adafru.it/cVI>) (this SMS pet food dish project uses the same electronics as the candy bowl monitor project)



Continue on to learn about the hardware used in this project.

Hardware

The electronics for this project are exactly the same as the [WiFi candy bowl project \(https://adafru.it/cZv\)](https://adafru.it/cZv). The only difference with this project is that the electronics are attached to a pet food dish instead of a candy bowl.

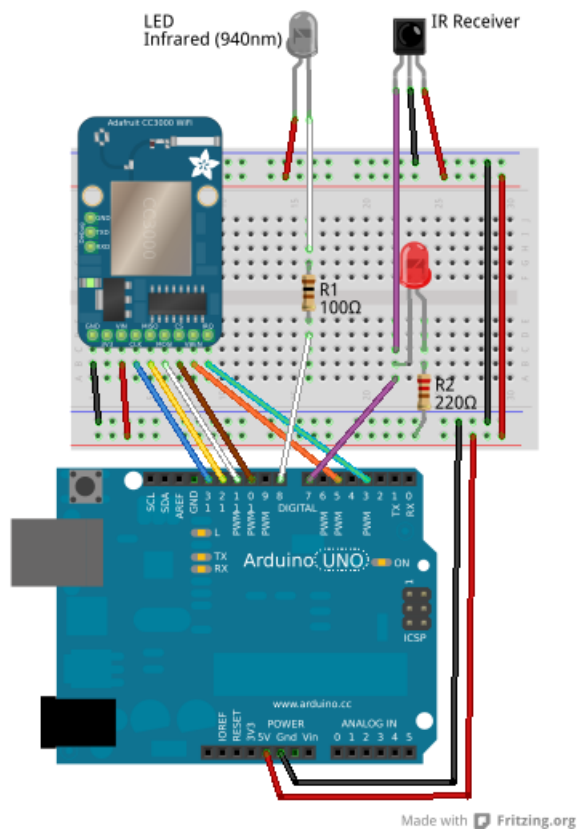
Parts

To build this project you'll need the following hardware and tools:

- [Arduino Uno \(http://adafru.it/50\)](http://adafru.it/50), [Mega \(http://adafru.it/191\)](http://adafru.it/191), or Nano
 - You need to use an Arduino which is compatible with the CC3000--check the [product page \(http://adafru.it/1469\)](http://adafru.it/1469) for the latest compatibility information. An Uno is best
- [CC3000 shield \(http://adafru.it/1491\)](http://adafru.it/1491) or [breakout \(http://adafru.it/1469\)](http://adafru.it/1469)
- [Infrared light sensor \(http://adafru.it/157\)](http://adafru.it/157) and [Infrared LED \(http://adafru.it/387\)](http://adafru.it/387)
 - Make sure the IR sensor and LED are matched to the same light wavelength (usually 940nm). The IR sensor should be one tuned to detect 38khz signals.
- [LED \(http://adafru.it/777\)](http://adafru.it/777) (any color) as an indicator of the IR sensor receiving signals.
- Two 1/4 watt resistors:
 - R1: ~100 to ~1000 ohm resistor to limit the current to the IR LED.
 - R2: ~200 to ~1000 ohm resistor to limit the current to the indicator LED.
- Power source for the Arduino, such as a [9-volt battery \(http://adafru.it/67\)](http://adafru.it/67) or [wall wart \(http://adafru.it/63\)](http://adafru.it/63).
- [Solid core hookup wire \(http://adafru.it/1311\)](http://adafru.it/1311)
- [Breadboard \(http://adafru.it/64\)](http://adafru.it/64) or [perf-board \(http://adafru.it/571\)](http://adafru.it/571)
- Pet food dish you can mount the IR sensor and LED within.
 - Because you need to drill into the bowl to mount the sensor and LED, a [plastic bowl \(https://adafru.it/cZw\)](https://adafru.it/cZw) will be the easiest to use with this project.
- Drill to make mounting holes in the bowl for the sensor and LED.
- Hot glue to secure the sensor and LED inside the bowl.
- [Soldering tools \(http://adafru.it/136\)](http://adafru.it/136) to attach wire to the IR LED and sensor

[Wiring](https://adafru.it/cZx)

Connect your hardware as follows:



For the CC3000, connect it to the Arduino in the [same way as this CC3000 tutorial \(https://adafru.it/cRU\)](https://adafru.it/cRU):

- Arduino 5V to CC3000 VIN
- Arduino ground to CC3000 ground
- Arduino digital pin 13 to CC3000 CLK
- Arduino digital pin 12 to CC3000 MISO
- Arduino digital pin 11 to CC3000 MOSI
- Arduino digital pin 10 to CC3000 CS
- Arduino digital pin 5 to CC3000 VBEN
- Arduino digital pin 3 to CC3000 IRQ

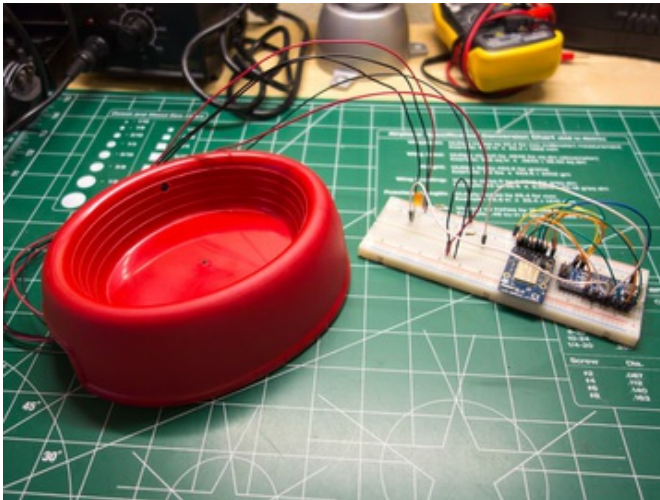
For the IR LED, solder long wires (long enough to reach from the bowl to the breadboard) to each leg--don't forget which leg is positive (longer) and negative! Connect the positive wire from the IR LED through to 5V power from the Arduino. Connect the negative wire from the IR LED through the R1 ~100 ohm resistor to Arduino digital pin 8.

For the IR sensor, again solder long wires to each leg. Connect the Vs leg to 5V power, the ground leg to ground, and the output leg to both Arduino digital pin 7 and the negative leg of the indicator LED. [Check the datasheet \(https://adafru.it/aKd\)](https://adafru.it/aKd) if you're unsure which pin is which on the sensor. Finally, hook up the positive leg of the indicator LED through the R2 ~220 ohm resistor to 5V power.

As a quick test, with the Arduino powered on you should be able to aim a remote control at the IR sensor and press buttons to see the indicator LED flash.

The Arduino will flash the LED on one side of the bowl and then see if it can detect those flashes on the IR detector side. This is easier to do than a laser-type break beam sensor, and the light isn't visible to you or the pet so it isn't distracting.

This pet food dish is only appropriate for use with dry food. Putting water or wet food inside the bowl could leak through the hole for the LED.



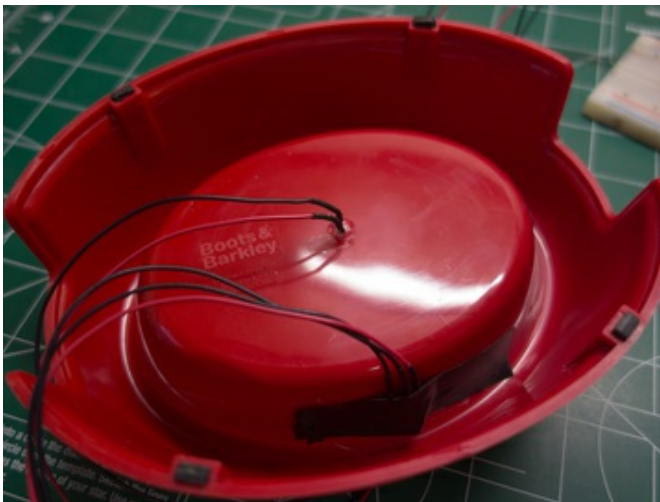
Pet Food Dish

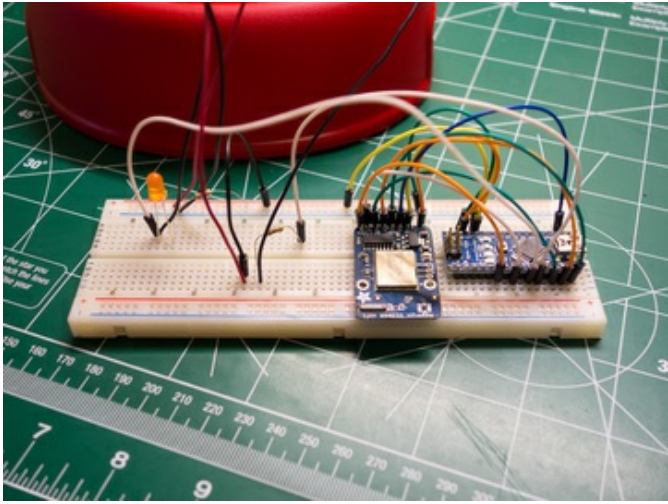
For the pet food dish you will want to drill a small hole just big enough to show the top of LED in the bottom center of the dish. Drill a hole up on the side of the dish just big enough to hold the IR sensor. Hot glue the LED and sensor to the bowl from the back.

Note: You might need to cover the back of the IR sensor with black tape to prevent stray IR light bouncing up from below the bowl and erroneously being detected.

Finally route the wires outside the bowl to the breadboard and hook them up appropriately. You will likely want to hide the electronics and wires inside of a small box to prevent your pet from getting to them.

Look at the pictures to the left to see how I constructed my pet food dish.





Be careful to ensure the LED and sensor are firmly attached to the bowl. Supervise your pet around the bowl so you can ensure they don't cause harm to themselves or the bowl.

Continue on to learn how to set up Amazon's Simple Notification Service.

Amazon SNS Setup

This project uses the [Amazon Simple Notification Service \(https://adafru.it/cXo\)](https://adafru.it/cXo) to send notifications such as SMS messages, emails, and more. The SNS service provides a simple publish web service which the Arduino can call directly using the CC3000. Once a notification is published, it will be sent to all subscribers of the notification topic. Using SNS you can easily control how and where notifications of the pet food bowl are delivered directly through Amazon's web interface.

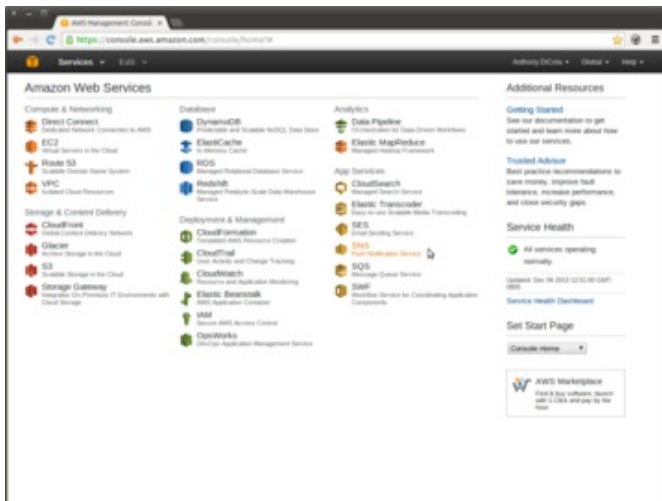
One thing to note is the [pricing of the SNS service \(https://adafru.it/cXo\)](https://adafru.it/cXo). You can currently send 100 SMS messages and 1000 emails a month for free. If you exceed these limits you will be charged a small fee, so be careful with your SNS use in projects. The code for this project implements a configurable time-based limit of how often a notification can be published. Using a rate limit like this will help keep you from exceeding the limits of the SNS service free tier.

Before you can set up the SNS service, you must first create an Amazon Web Service account if you don't have one already. To create an account you can follow the steps under 'Sign up for the Service' at [this link \(https://adafru.it/cXi\)](https://adafru.it/cXi). Make sure to save the access key and secret access keys for your AWS account in a handy place as they will be needed later in the configuration of the Arduino sketch.

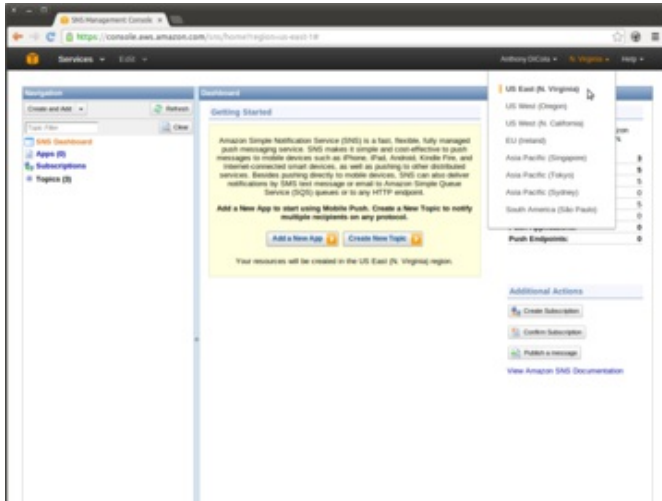
Once you have an AWS account, follow the steps below to setup SNS publishing for this project:



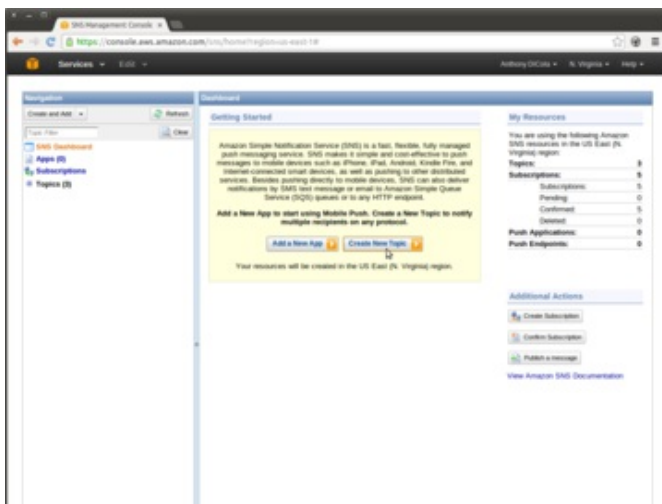
Navigate to <http://aws.amazon.com/> (<https://adafru.it/cYb>) and access the 'AWS Management Console' link in the upper right of the page.



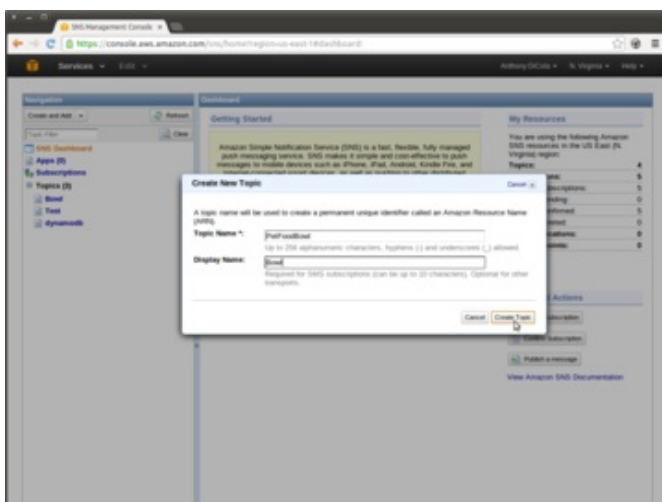
Click the SNS service link in the list of Amazon services.



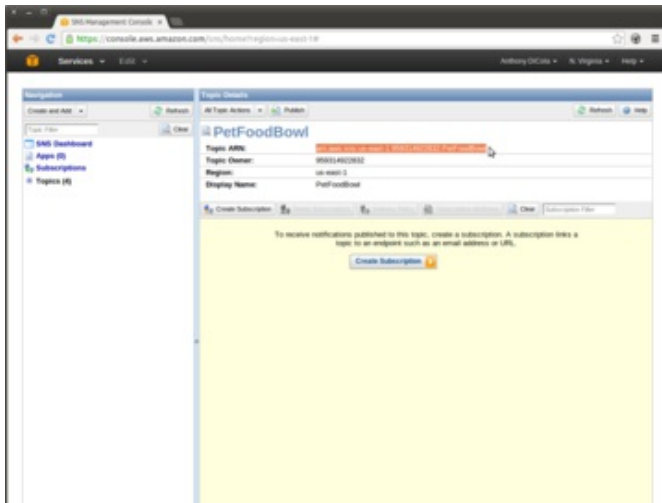
For the best performance, switch to a different region which is near your physical location using the region list in the upper right.



Click 'Create New Topic' to open the new topic creation dialog.



Enter a topic name value such as 'PetFoodBowl', and a short display name value such as 'Bowl'. The display name will be used as the prefix for SMS messages so it should be a short, descriptive identifier.



At this point your SNS topic has been created successfully. You will want to take note of these two values:

- **Topic ARN** - This ARN value uniquely identifies the SNS notification topic and will be used in the pet food bowl sketch configuration.
- **Region** - The region identifies where the SNS notification topic was created. This value will be used in the pet food bowl sketch configuration.

Next click 'Create Subscription' to create a new subscription to this SNS topic. Subscriptions can be of various types, like SMS messages, email notifications, HTTP service calls, or more. You can have multiple subscriptions to a single topic, for example both an SMS subscription and an email subscription.

Set up whatever subscription you prefer for this project. Note that each subscription has to be validated by replying to an email/SMS message.

With the steps above complete, you should have a topic created in SNS and at least one subscriber to that topic. If you wish to add, remove, or update subscriptions you can navigate to the SNS management console with the steps above and adjust the subscribers for the topic.

Continue on to learn about the software used in this project.

Software

You will need to download the Arduino sketch for this project at the following link:

<https://adafru.it/cZy>

<https://adafru.it/cZy>

Unpack this archive to a directory and load the sketch in Arduino. Don't forget you also have to have the [Adafruit_CC3000](https://adafru.it/d96) (<https://adafru.it/d96>) and [RTCLib](https://adafru.it/aX2) (<https://adafru.it/aX2>) libraries installed.

At the top of the sketch are various define values which control the configuration of the sketch. You will want to set these defines as follows:

- **ADAFRUIT_CC3000_IRQ, ADAFRUIT_CC3000_VBAT, ADAFRUIT_CC3000_CS:** Set these to the appropriate pins which are connected to the IRQ, VBEN, and CS pins respectively on the CC3000.
- **WLAN_SSID, WLAN_PASS, WLAN_SECURITY:** Set these to the appropriate values for your wireless network.
- **AWS_ACCESS_KEY, AWS_SECRET_ACCESS_KEY:** Set these to the values of your personal AWS account access keys.
- **AWS_REGION:** Set this to the region value of the SNS topic created in the previous step. You can see a list of region values [in this table](https://adafru.it/cZz) (<https://adafru.it/cZz>).
- **AWS_HOST:** Set this to the appropriate endpoint URL for the region of your SNS topic from [this table](https://adafru.it/cZz) (<https://adafru.it/cZz>).
- **SNS_TOPIC_ARN:** Set this to the *URL encoded* value of the topic ARN found in the previous step. Note that this value **must be URL encoded**--you can use a tool such as [this one](https://adafru.it/cZA) (<https://adafru.it/cZA>) to URL encode the topic ARN value.
- **IR_LED:** Set this to the Arduino pin which is connected to the infrared LED.
- **IR_SENSOR:** Set this to the Arduino pin which is connected to the infrared receiver.

The rest of the configuration values do not need to be modified to run the sketch. You can read the comments associated with each value to understand their meaning.

Compile the sketch and upload it to your hardware. Open the serial port at 115200 baud and you should see the hardware connect to your wi-fi network and wait for the bowl to become empty.

To test everything is working, block the IR LED & sensor for ~5 seconds (the default **BOWL_CHECK_SECONDS** configuration). After unblocking the LED and sensor you should see serial output as the hardware sends a notification to the SNS service. Specifically the response from the SNS web service publish call will be displayed in the serial output.

Make sure you see a 'HTTP/1.1 200 OK' response from the SNS service in the response. If you see a response such as 'HTTP/1.1 400 Bad Request' or 'HTTP/1.1 403 Forbidden' the message did *not* publish and you should check the AWS access key, region, and host values are correct.

Note that by default the sketch will only publish a notification once every 12 hours. This is to prevent too many notifications being published and potentially pushing your usage outside Amazon's free usage tier. You can adjust this rate limit by changing the **PUBLISH_LIMIT_SECONDS** configuration value to a different period of time. Also be aware that this rate limit is not stateful and will reset if the power or Arduino is reset.

Once you see a successful publish your hardware is complete!

Continue on to learn about future work you can do with this project.

Future Work

This project is a great example of sending notifications such as SMS and email with an Arduino. By using a service such as Amazon's SNS notification service, the complex work of delivering notifications is done for you automatically. Furthermore you get complete control over how and where notifications are delivered through Amazon's web interface.

Some interesting ways you might extend this project include:

- Create a do-it-yourself 'security system' to send you notifications when doors or windows are unexpectedly opened at home. Be alerted when a nosey sibling is going in your room!
- Use a load cell/scale to sense how much food is in the dish and send more accurate status messages.
- Create different SNS subscriptions such as [HTTP endpoints \(https://adafru.it/cZB\)](https://adafru.it/cZB) (good for notifying other web services you build) or [SQS queues \(https://adafru.it/cZC\)](https://adafru.it/cZC) (good for notifying other Amazon web services).
- Use the signature generation code in the sketch to call other Amazon web services, such as the [S3 storage service \(https://adafru.it/cXc\)](https://adafru.it/cXc).

What ways can you think of to extend this project?

Have fun adding notifications to your own Arduino projects!