Pop Goes the Happy Face

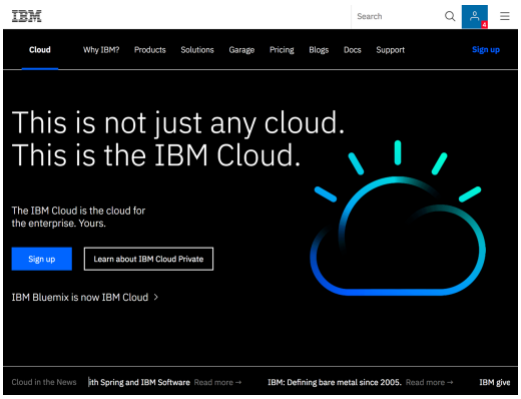
Using the power of Watson Speech to Text, IBM Cloud, and Watson Tone Analyzer, this game will pick up the tone in your statements to determine which plush toy to display!

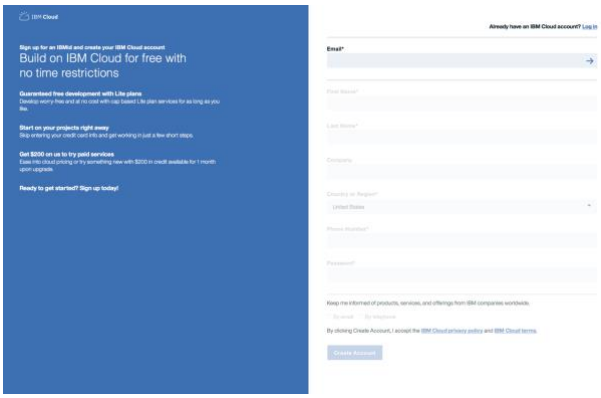


I. Required Devices
   a. Raspberry Pi 3
   b. Male-to-Female and Female-to-Female wires
   c. Microphone Headset with USB (recommended: with mute button)
   d. Servos (3; recommended: Feathermicro: hs-55)

II. Raspberry-Pi and Node-Red Set Up
   a. The Raspberry-Pi should be preconfigured with NOOBS (New Out Of Box Software). If you have never used a Pi before, I suggest reading one of this blog. If you already bought your Pi with NOOBS, some of those steps are unnecessary, but it may be helpful to understand the steps anyway. This article also covers the external device you need to work with the Pi.
      https://towardsdatascience.com/raspberry-pi-3-for-the-first-time-50634b115620
   b. Installing Node-Red on a Raspberry Pi
      Node-Red is the flow-base development tool we will use to piece together the apparatuses included in this project.
      Follow the instructions on this page to install Node-Red onto your Pi:
      https://nodered.org/docs/hardware/raspberrypi

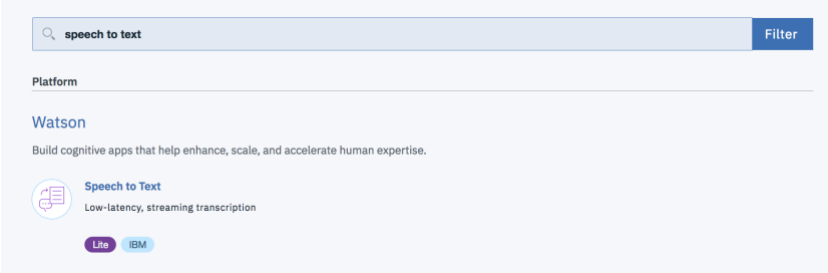
III. Provision IBM Cloud Services
   a. Speech to Text
   b. Tone Analyzer
   c. Sign up for IBM Cloud Account
      i. Go to https://www.ibm.com/cloud/
      ii. We are going to sign up for a free IBM Cloud account.
      iii. Click “Sign up”.
      iv. Fill in the required boxes.
      v. Click “Create Account”.



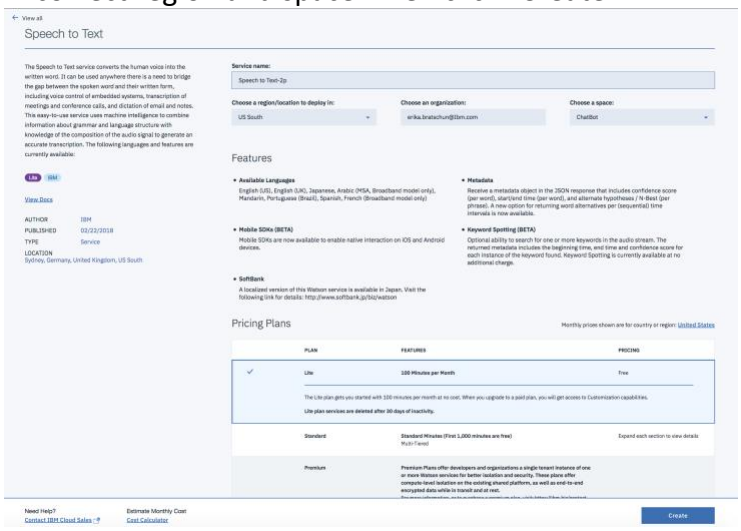


d. Provisioning a Service:
   i. Open up the Catalog by selecting “Catalog” at the top of the page
   ii. Type in “Speech to Text” into the search bar

Speech to Text is a Watson Service



   iii. Open the “Watson Speech to Text Service” by clicking on the title.
   iv. The free plan is selected by default. The service name is unique to your space by default but can be changed. Make sure you are working in the correct region and space. Then click “Create”.

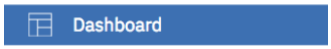


   v. Return to the Catalog by once again selective “Catalog” on the top of the page. Alternatively, you can return by navigating to the catalog from the hamburger on the top left corner.
   vi. We need to provision another service, search for Tone Analyzer.

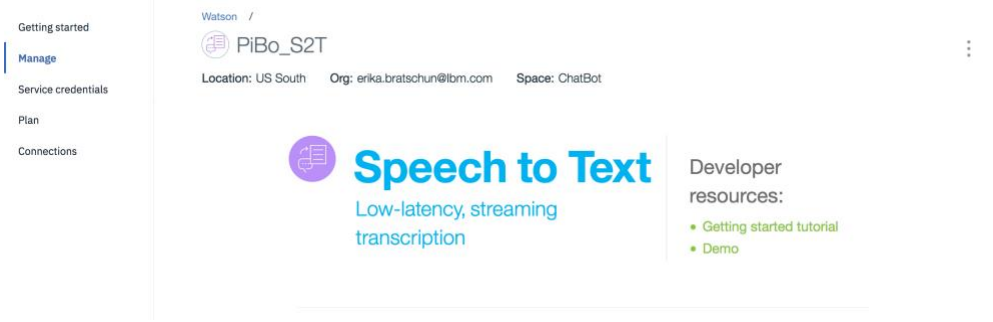
Tone Analyzer is another Watson Service.

Repeat steps iv-v.

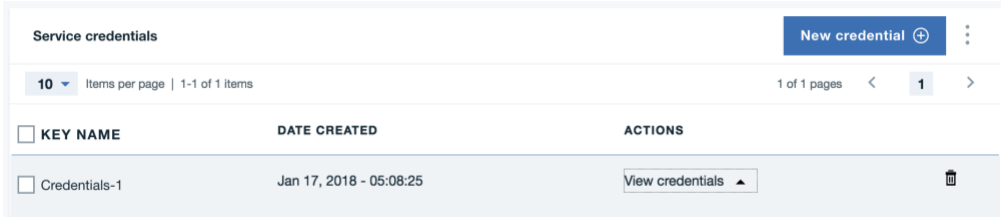
e. Service Credentials
    i. Return to your dashboard by clicking the hamburger icon in the top left corner and selecting “Dashboard”.



    ii. Both of your newly provisioned services will be listed on your dashboard.
    iii. Open the Speech to Text service by clicking on it.



    iv. On the left side of the screen select “Service Credentials”.
    v. Click “New Credentials”, “Add”, and take note of the username and passwords. These credentials will be used in the Node-Red flow to communicate with the services.



    vi. Repeat steps i-v for Tone Analyzer.

IV. Node-Red

a. Import Node-Red Flow
    i. Open Node-Red from your Raspberry Pi
        1. Use the command *node-red-start* to instantiate Node-Red
    ii. Open up the flow by pointing to a browser at http://XXXX. See below to find your unique address.

```
pi@raspberrypi:~ $ node-red-start

Start Node-RED

Once Node-RED has started, point a browser at http://10.0.0.18:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use   node-red-stop                          to stop Node-RED
Use   node-red-start                         to start Node-RED again
Use   node-red-log                           to view the recent log output
Use   sudo systemctl enable nodered.service  to autostart Node-RED at every boot
Use   sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
Started Node-RED graphical event wiring tool..
Started Node-RED graphical event wiring tool..
```

iii. Now working in the browser, click on the hamburger on the right > Import > Clipboard

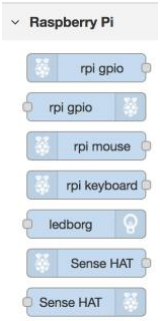
iv. Copy and paste the code found in the library titled **FILLIN** XXXXX

b. Import and Install necessary nodes

i. TJ Bot direction can be found at this link:
Credit to JeanCarl Bison for the TJ Bot Nodes. His Github page contains other recipes to play around with if you want to create a TJ Bot:
https://github.com/jeancarl/node-red-contrib-tjbot

ii. GPIO Nodes

1. May be in in the node pallet by default

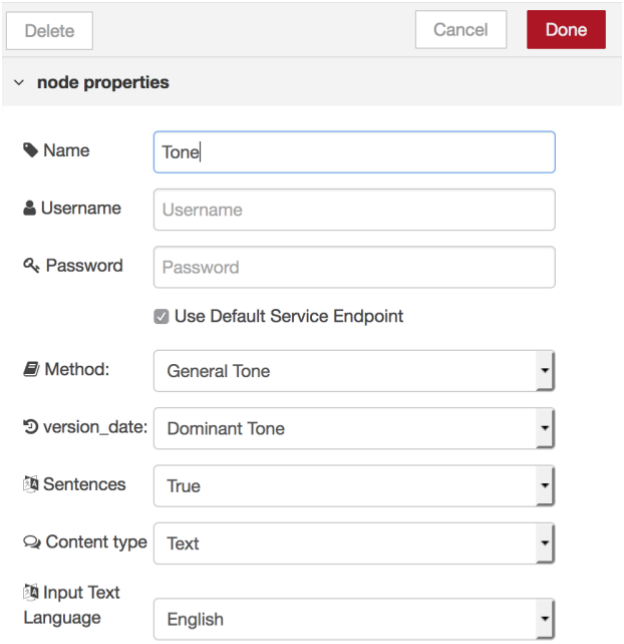


2. Type into the terminal: *npm install rpi-gpio*
3. Initializing the nodes requires some Python code. Return to the terminal window and type the following commands:
$sudo pthong
>>> import RPI.GPIO as GPIO
>>> GPIO.setmode(GPIO.BOARD)
>>> GPIO.setup(11,GPIO.OUT)
>>> GPIO.setup(13,GPIO.OUT)
>>> GPIO.setup(15,GPIO.OUT)

More can be read at http://www.toptechboy.com/raspberry-pi/raspberry-pi-lesson-28-controlling-a-servo-on-raspberry-pi-with-python/. This link provides a read on the ways to control the servos based on pwm and duty cycle. We will use the nodes to configure these in the flow.

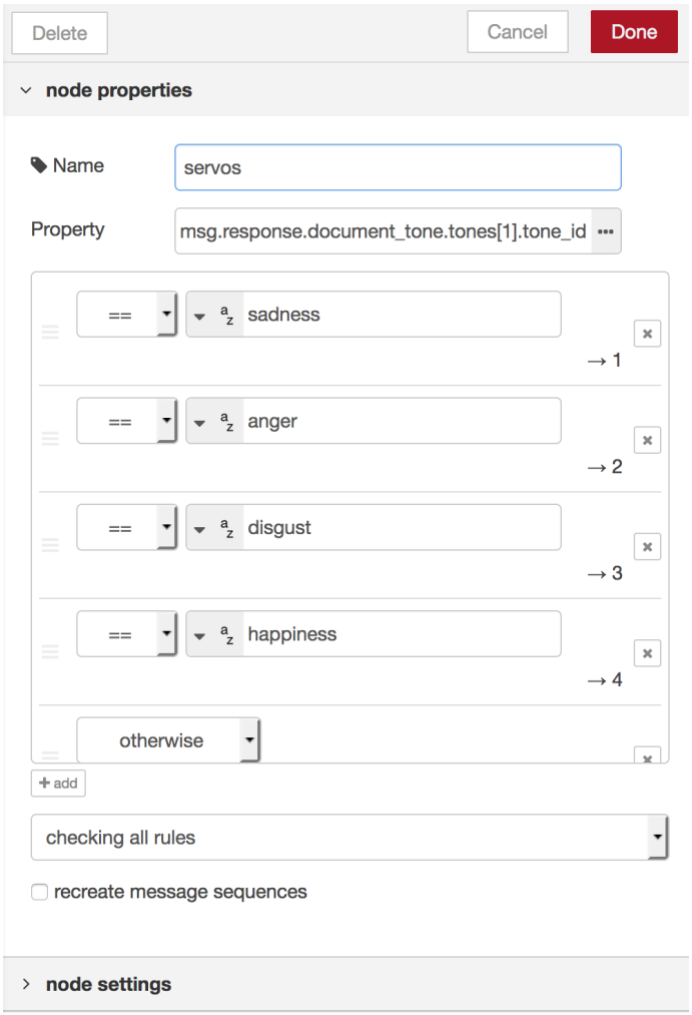
Note: Once the hardware is wired, the later steps in that article can be used to test the servos, without using the node-red flow.

c. Tone Analyzer Node
   i. Double click on the tone analyzer node.
   ii. This is where you are required to import the credentials from the tone analyzer service from IBM Cloud.
   iii. Method: General Tone; version_date: Dominant Tone



d. Switch Node
   i. The switch node will take in the dominant tone from tone analyzer and send that information to the switch node. The responsibility of the switch node is to direct the tones to the proper trigger nodes to instantiate the respective servos for that tone.

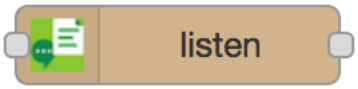
   ii. Double click on the switch node
   iii. Change the Property to msg.response.document_tone.tones[0].tone_id
   iv. Add and configure the tone inputs by click on the “+add” button.



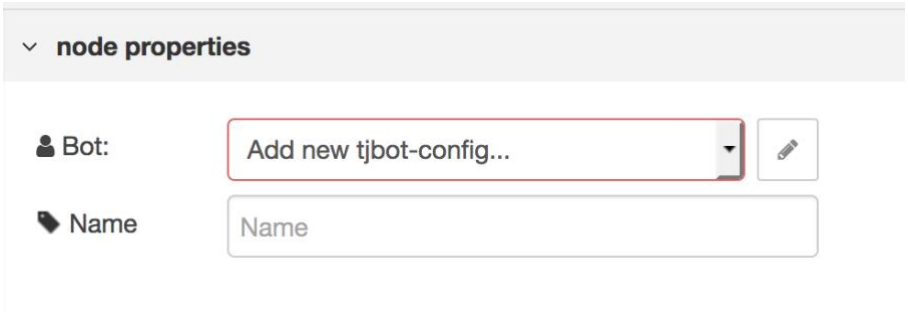
v. Since, we have three output servos so a few emotions will map to the same trigger nodes. (ie happiness and joy; anger and disgust)

e. Listen Node

i. Double click on the Listen Node



ii. “Listen” is a TJ Bot, preconfigured node. Once the node is open, click add Bot and put in the required credentials under “Speech to Text”.



f. rpi gpio Node
   i. Double click on any of the rpi gpio nodes

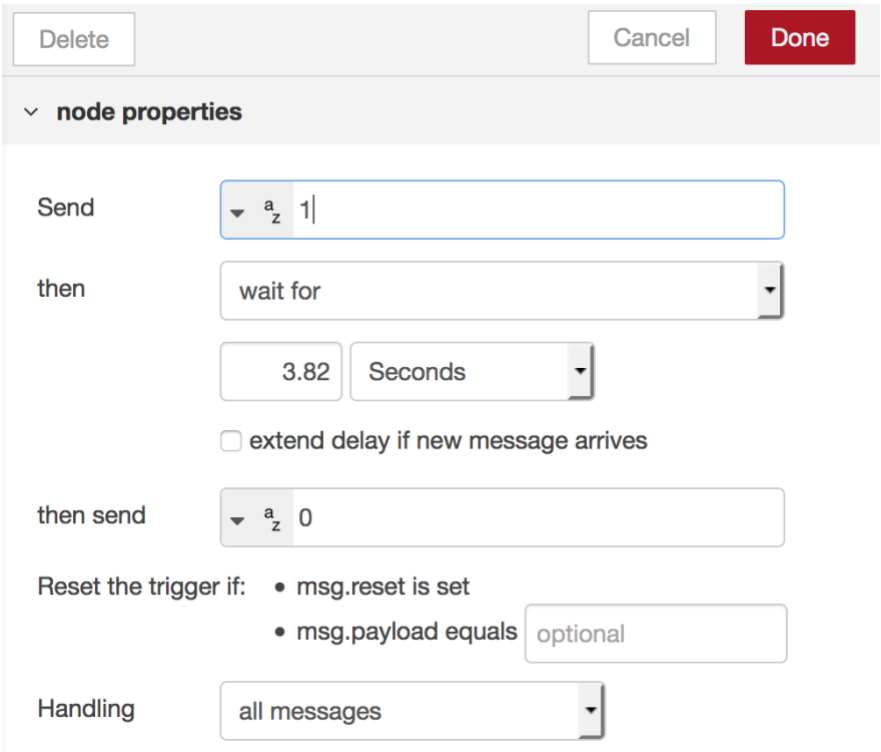
   ii. The rpi gpio node allows us to declare the output pin that we are targeting based on our connections between the data wires to the servos.

Each of the three rpio gpio pins need to be mapped to a different output pin. As mentioned above, we are using GPIO17 – 11, GPIO27 – 13, and GPIO 22 – 15.

   iii. Repeat these steps for the other two rpi gpio nodes.

g. Trigger Node
   i. Double click on any of the trigger nodes

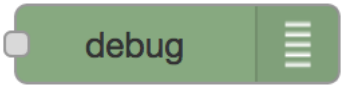
ii. Configure the first trigger node to match the configurations shown below:



Note: The specifications recommended may not work for all servos the same way. By adjusting the "wait for __XX___ seconds" in the trigger node and the frequency in the rpi gpio node you can control the speed and time the servo is rotating. The trigger node time declares the amount of time the particular servo is rotating. By adjusting this parameter and the frequency, the servo should make a 360 degree rotation and at a speed that you prefer.

h. Debug Nodes

i. The debug node is useful for debugging, hence the name ;) To view the outputs from the debug nodes, in the top right of the Node Red page is the debug tab. This is where the output will appear.



ii. Double clicking on the debug node: select msg.payload or msg.complete msg object

V. Hardware

a. Below is the schematic for wiring the servos to the Raspberry-Pi

i. Ground: BLACK. Always start the wiring with connecting the grounds. This is something to keep in mind anytime you are playing with electricity to avoid destroying your devices.

ii. Begin by wiring the ground wire from the Pi to the breadboard. This will require a male-to-female wire.

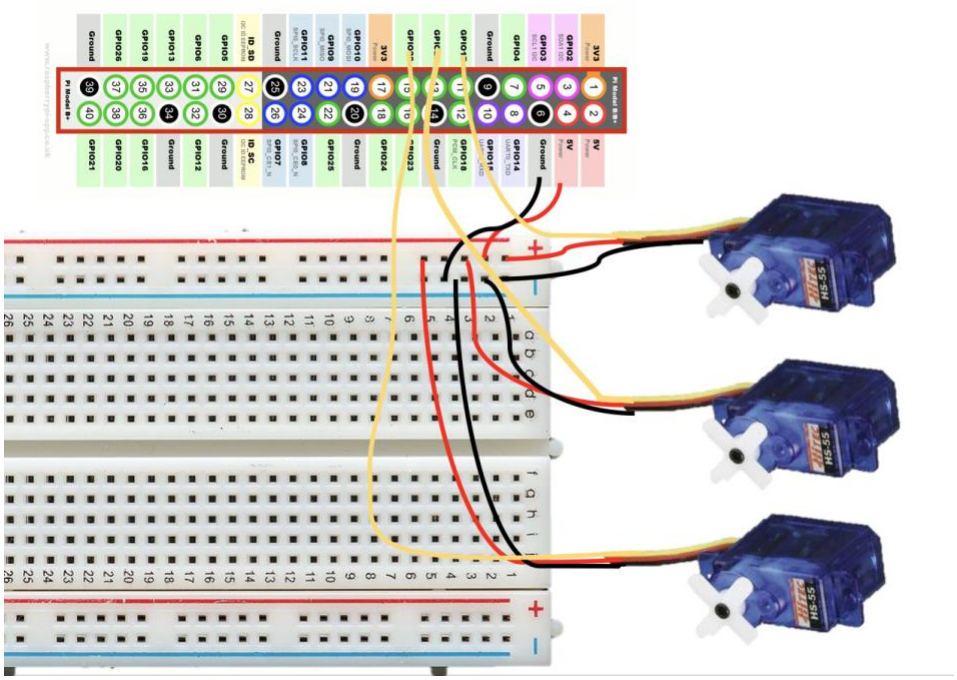
iii. Data: Yellow. Next, connect the data pins. The data pins used below are pins 11,13 and 15 which corresponds to GPIO 17, 27, and 22, respectively, and must be used based on the GPIO pin set up in earlier steps.

iv. Power: RED. The Pi has a build in 5V power source. The recommended servos have a spec of about 4-6V and function well with the 5 V power source.

b. Other hardware is connected to the Raspberry-Pi.

i. Connect a 5V power bank to the Pi.

ii. Using a USB port, connect the headset.



VI. Construction

Open for creativity!

VII. Purpose of the game

Choose your words wisely! Try to get each of the faces to pop up by talking to it. The plush toys will indicate the tone of your voice.