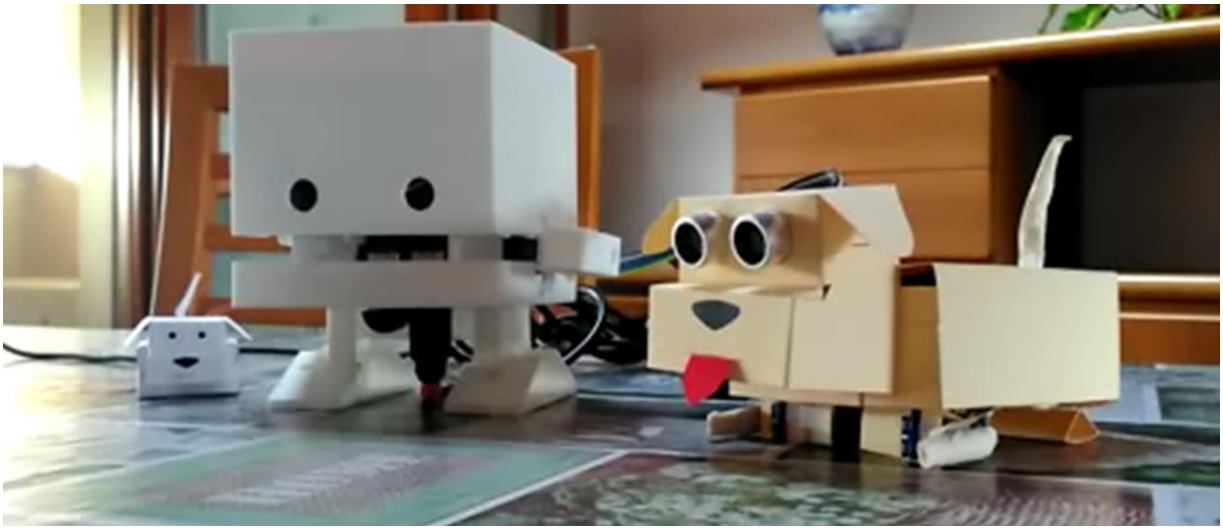
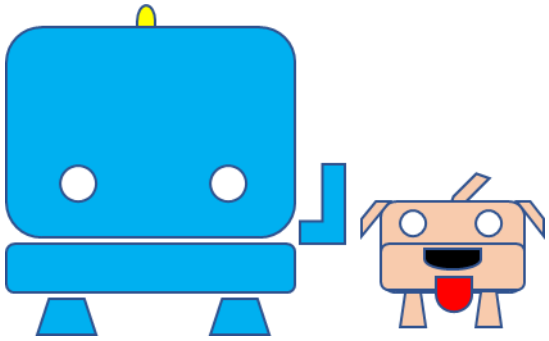
IBM Volunteers

Doggy – a pet companion for TJBot





Version: October 1st, 2018

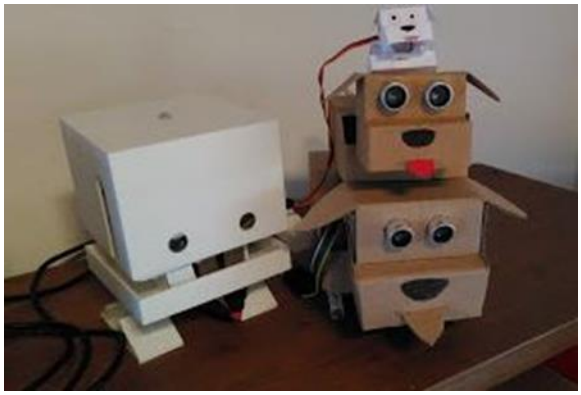
Check updates on https://github.com/fmanclossi/TJBot-playbook

Learning Objectives

This activity complement "What's Possible With AI – TJ Bot" Activity Kit expanding students' knowledge regarding robotics , cognitive technologies and AI by helping them build a simple companion for TJBot, a cute dog called Doggy.

Students will increase experience and skill using a Raspberry Pi, Servo Motors, Sonar sensor and unleash their creativity following TJBot and Doggy in funny tales powered by Watson™ services.

- Audience:
  - Ages 11-14 (Middle grade students)
  - Ages 14-19 (Secondary/high school students)
- What's the goal of this kit?
  - Introduce cognitive technologies and programming
  - Explore "embedded cognition"— everyday objects that use AI to interact with us
  - Support environmental education activity reusing cardboard and common material
- Why is this topic important?
  - The technologies introduced through this activity have a growing impact in our world.
  - A fun, creative, hands-on project is an ideal way to stimulate students' interest in STEM topics



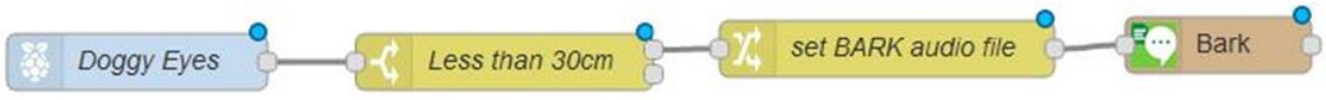
Activity Kit structure and generic requirements

This activity is composed by tales and scenes, with clear requirements and detailed steps.

Common requirements:

- TJBot already setup and tested with Node-Red nodes by JeanCarl Bisson
- Doggy already setup and tested (note: some scene requires a specific Doggy version. Build your Doggy according to the story that you want to tell). This activity also includes instructions to build Doggy.
- Skill: Node-Red (basic level)
- Age: 8+ (Doggy already setup and tested)
- Age: 12+ (Doggy not yet built and setup)

Before starting, verify if any updates are available on the Doggy Web Site, including more tales of TJBot and his Doggy.

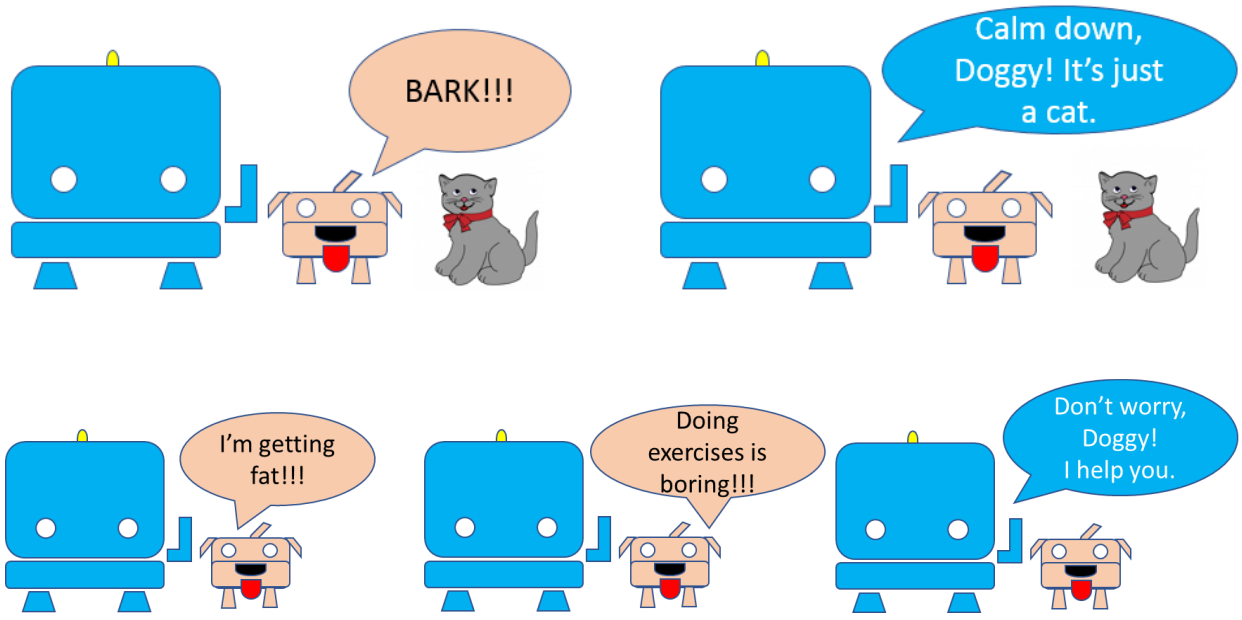


What are we going to do today?

TJBot loves pets even though they can be noisy. He/Her owns two dogs, Doggy and MiniDoggy.

Doggy is a very careful watchdog while MiniDoggy is more distracted one, it doesn't look at what happens around it but it's happy to play with TJBot.

We follow TJBot and its doggies in some funny stories learning how IBM solutions could be applied to real life examples.



Doggy is an original idea from IBM Italy Volunteers. The first draft was developed during a stage with high school students in IBM Italy Segrate, May 2018.

Table of Context

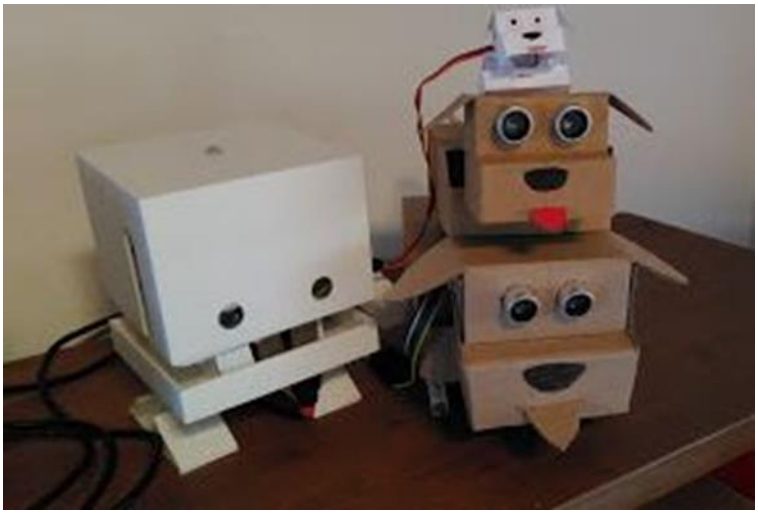
Instruction: Create and setup your own Doggy

Doggy's family:

- DOGGY BASIC - Doggy hosts an HC-SR04 sensor that provides a superdog vision.
- DOGGY STANDARD - In addition to BASIC format, Doggy has also a very cute tail, powered by one SG-90 servo motor.
- DOGGY ADVANCED (PREMIUM) - Doggy increases its capability with two motorized front legs powered by 2 x SG-90 servo motors.
- MINIDOGGY - This little Doggy has just a tail shacked by one SG-90 servo motor. Little and very friendly!

Doggy is very easy to build because it’s provided with A4 format design file (DXF/PDF). Just print/copy it and cut, fold, connect!

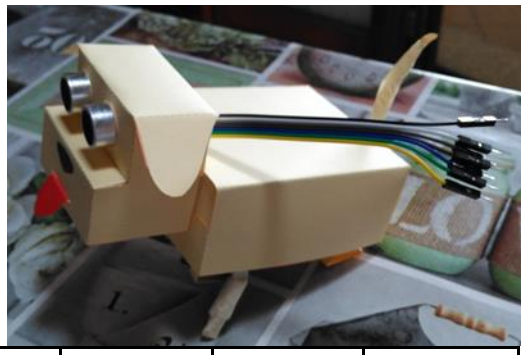
Some tasks could be tricky but could be performed under adult’s guidance.



Before let Doggy runs through digital fields, we must test its features and tune up some little things according with your Doggy version:

- Setup and Test Doggy superdog vision (HC-SR04)
- Setup and Test Doggy tail (SG-90)
- Setup and Test Doggy front legs (SG-90)

Hardware requirements



Component	Doggy Basic	Doggy Standard	Doggy Premium	MiniDoggy
TJBot setup with JCBisson Node-Red nodes	Yes	Yes	Yes	Yes
1 x HC-SR04 ultrasonic sensor for TJDog supervision	Yes	Yes	Yes	No
1 x SG90 servo motor for TJDog tail	No	Yes	Yes	Yes
2 x SG90 servo motor for TJDog front legs	No	No	Yes	No
1 x Breadboard	No	Optional	Yes	No
Cables, Cardboard, stapler, marking pens, scotch tape	Yes	Yes	Yes	Yes

Download and Print/Copy Design files

Get the last design file from Doggy Github house

- Doggy Basic - Superdog vision only (Doggy DXF format, Doggy PDF format - basic parts)
- Doggy Standard - Basic plus ultra-dynamic tail (Doggy DXF format, Doggy PDF format - basic parts
- Doggy Premium - Standard plus powerful front legs (Doggy DXF format, Doggy PDF format - basic parts, Doggy PDF format - Optional and advanced parts)
- MiniDoggy - ultra-dynamic tail only (MiniDoggy DXF format, MiniDoggy PDF format)

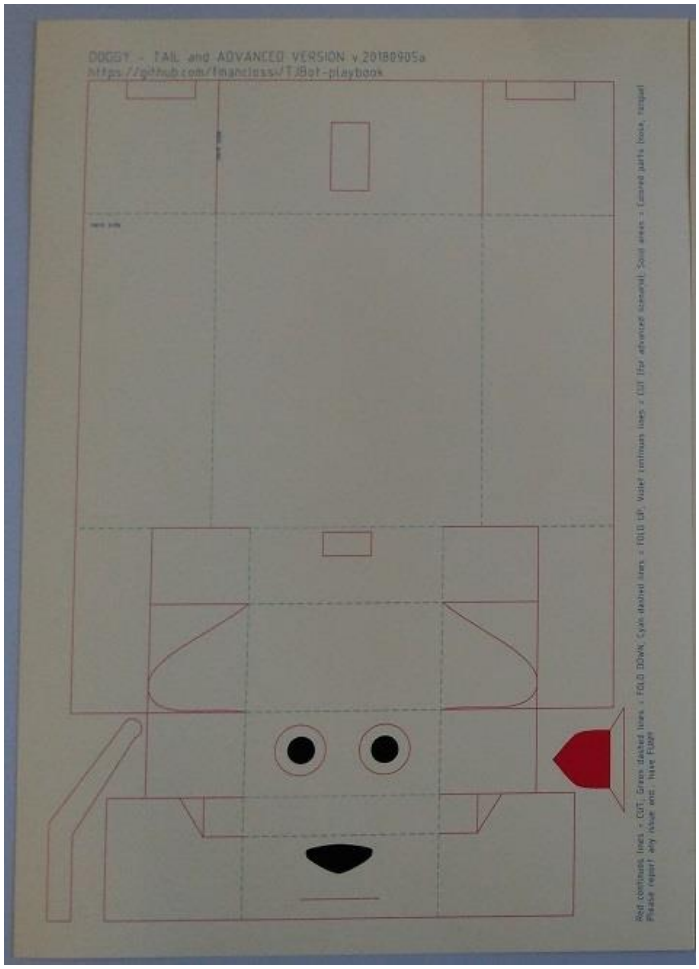
Print or Copy the design file on selected support:

- Doggy Basic - Do not use common printer paper. Doggy Basic can be easily printed on 200gr/m² A4 paper (local print shop or standard printer) or cardboard. 200gr/m² A4 paper is enough strong to support sensor weight. No need for "Belly" part (but if you like it, just fold and insert it).
- Doggy Standard – As Basic, "Belly" part is optional, depending upon wiring choices (but if you like it, just fold and insert it).
- Doggy Premium – As Basic, except that "Belly" part is mandatory. It could be printed on 200gr/m² A4 paper, but cardboard is the best choice.
- MiniDoggy - MiniDoggy can be easily printed on 200gr/m² A4 paper (local print shop or standard printer) 200gr/m² A4 paper is enough strong to support servo motor weight. No need for "Belly" part (but if you like it, just fold it and insert mounting servo on it). Belly part is mandatory if you use standard printer paper.

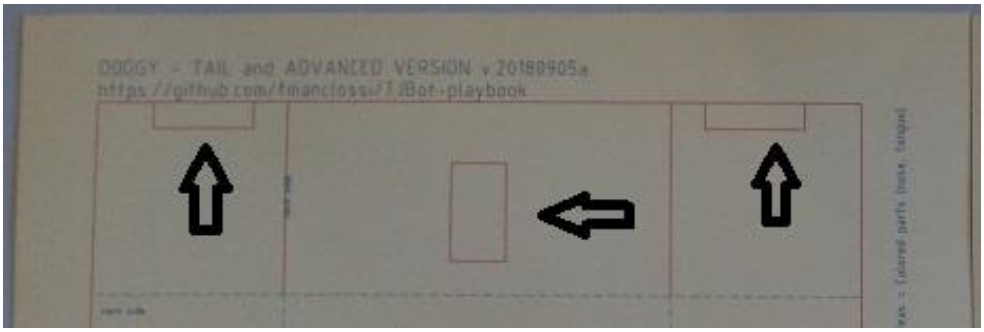
HINT: The best way to learn how to build Doggy is to print it on common A4 paper and build a prototype.

Mount Doggy Basic - a TJBot companion with superdog vision!

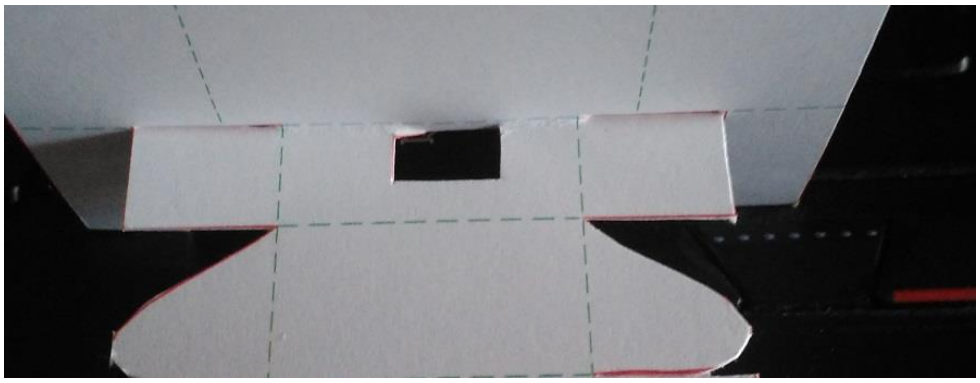
Get and Print/Copy DOGGY BASIC design file - Print/reproduce only the Tail and advanced Sheet



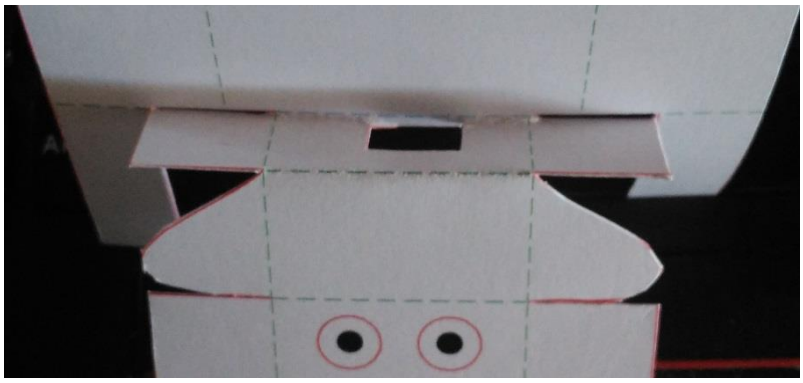
Cut all the red lines using a cutter and a scissor (do not cut parts to host servo motor for tail). If you plan to use superdog vision, remind to cut eyes too. (P.S: in the following pictures these lines are cut as used in Standard and Advanced version)



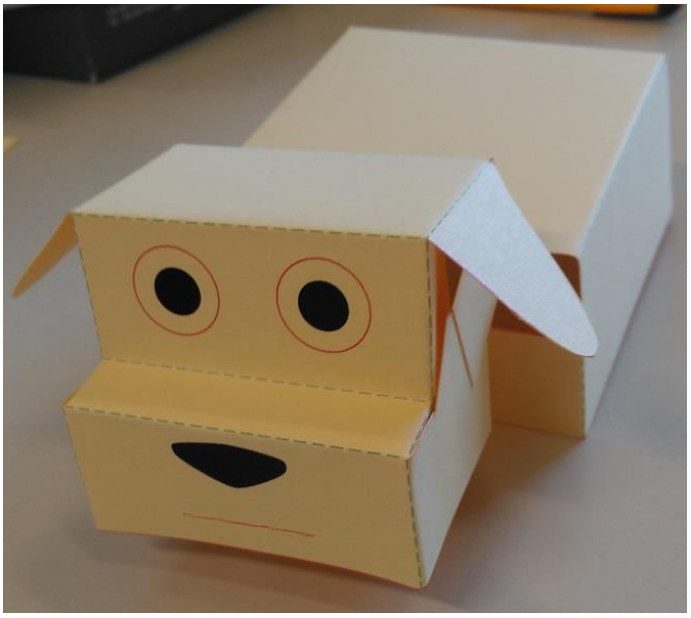
Fold up or down green and green lines



Fold up

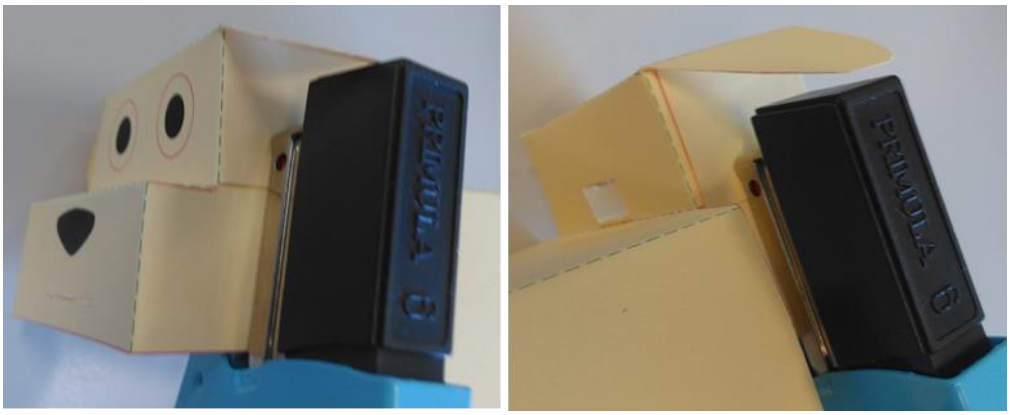


Fold down

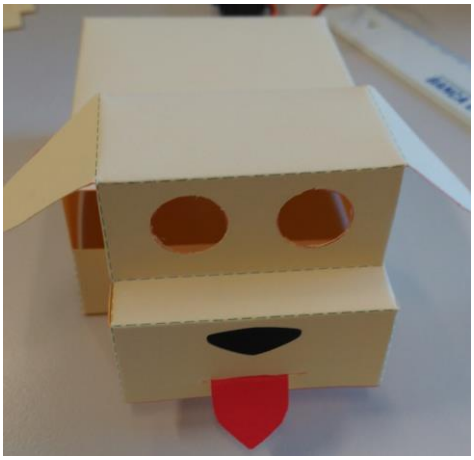


Doggy is folded up & down (and I forgot to cut eyes ☹)

Staple ears and tail, insert tongue



Staple left and right ears



insert tongue and clip the tail

Insert and wire HC-SR04 sensor to TJBot (no need for breadboard).

Build a leash keeping connectors together… or you can braid them ☺

Description	From TJBot to Doggy's eyes
Type of connectors	Female to Female
Pin 25 - Ground	Blue to Ground
Pin 23 - GPIO 11 - Echo	Green to Echo
Pin 21 - GPIO 09 - Trigger	Yellow to Trigger
Pin 4 - 5V	Orange to VCC

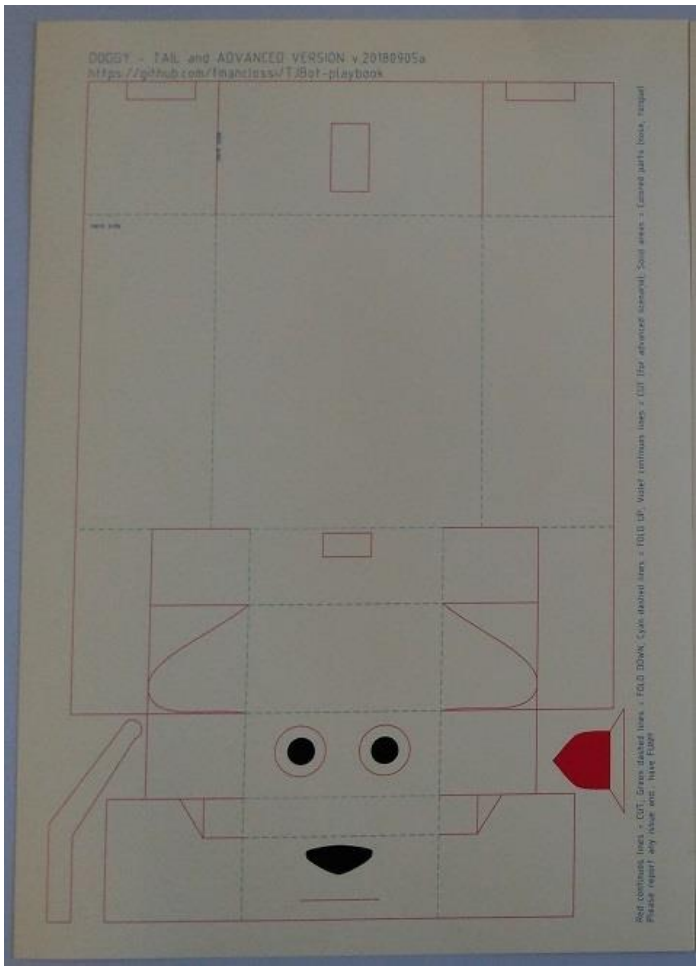
WARNING: pay attention to connect the right PINs to avoid damages to components.

Enjoy your Doggy Basic!!!

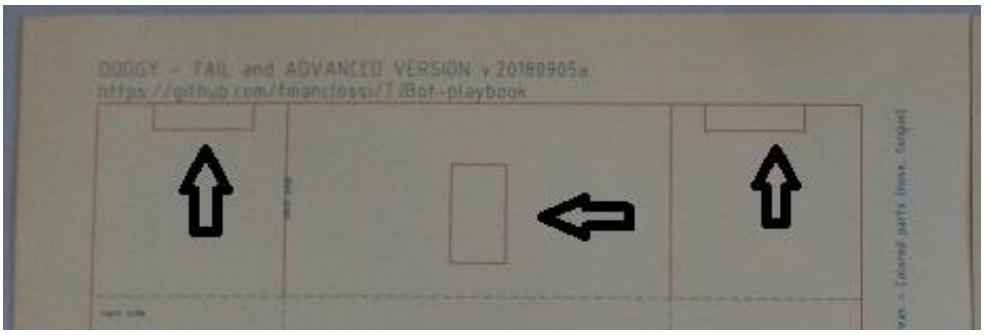
Please, share your experience on Doggy's House!!!

Mount Doggy Standard - a TJBot companion with superdog vision and a cute tail!

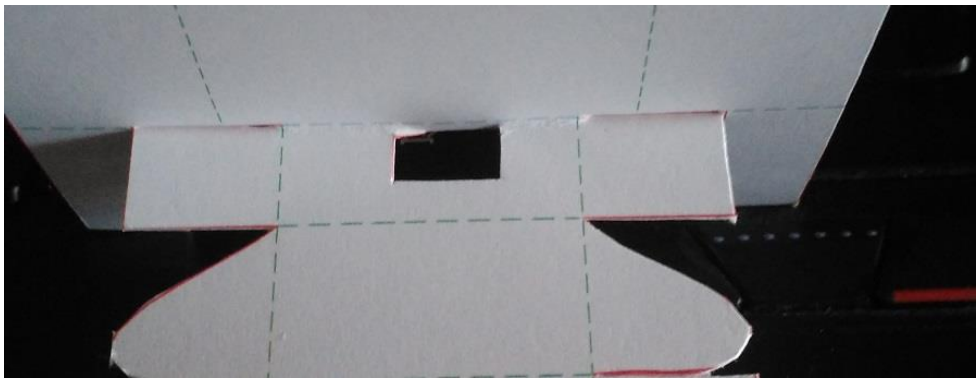
Get and Print/Copy DOGGY STANDARD design file - Print/reproduce only the Tail and advanced Sheet



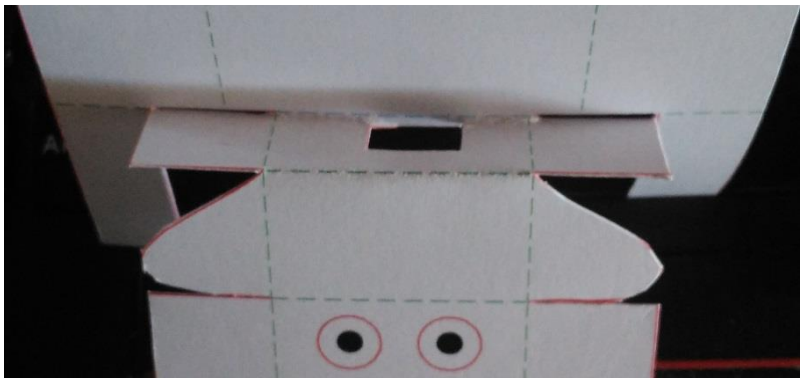
Cut all the red lines using a cutter and a scissor. If you plan to use superdog vision, remind to cut eyes too.



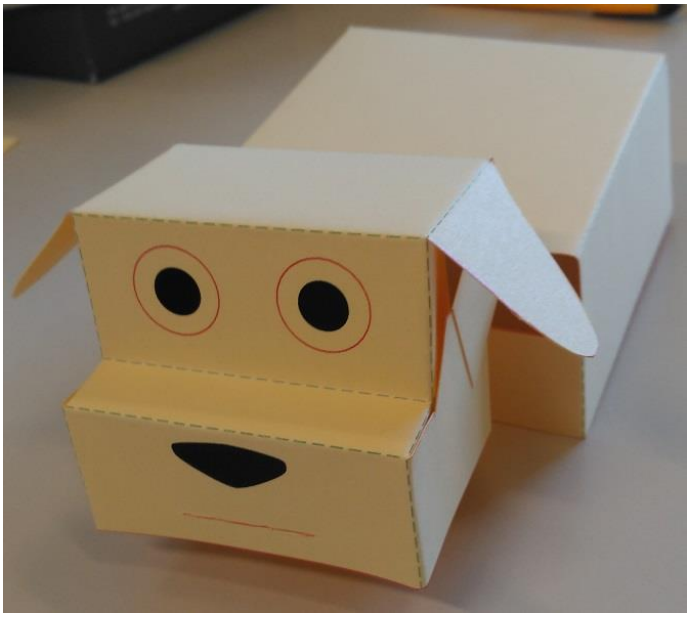
Fold up or down green and green lines



Fold up

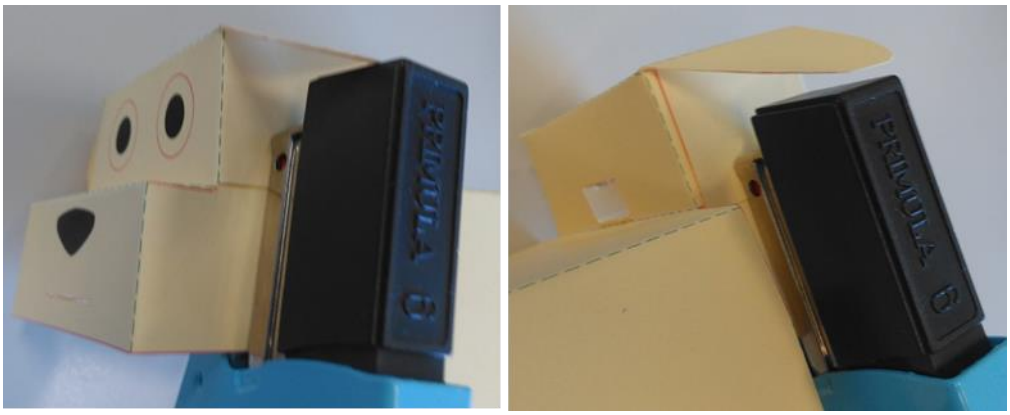


Fold down



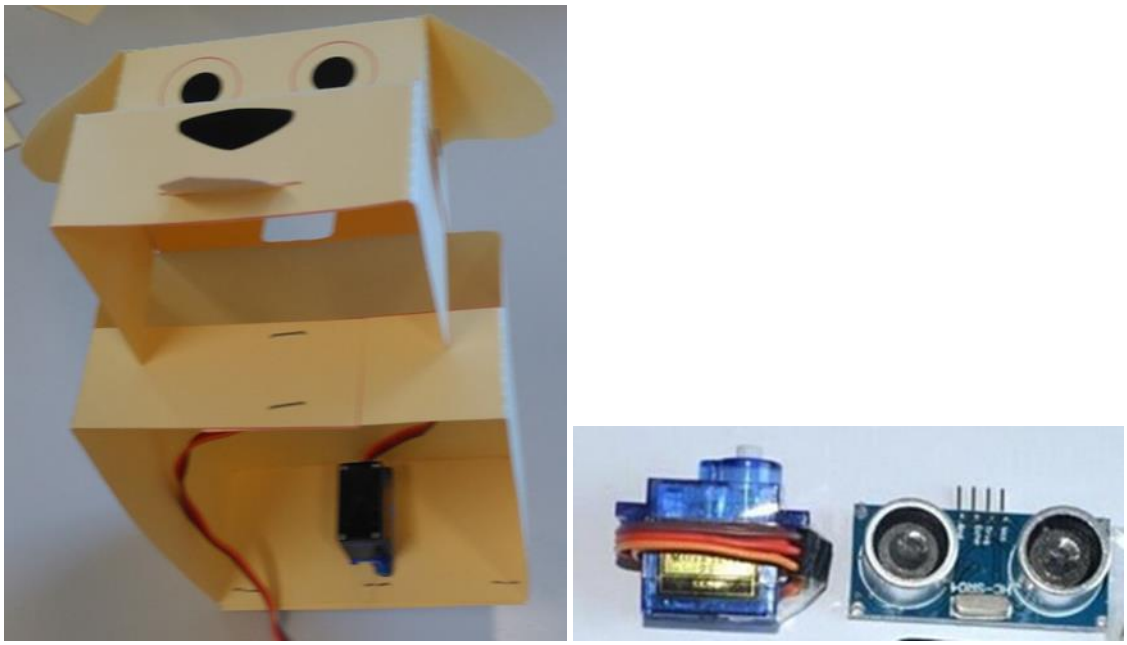
Doggy is folded up & down (and I forgot to cut eyes ☹)

Staple ears and tail.



Staple left and right ears

insert tongue and mount servo motor as shown in figure. (if you forget to cut eyes, remove them). Insert HC-SR04 sensor, too.



Wire HC-SR04 sensor and SG-90 servo motor to TJBot (look at the picture to insert servo with the right orientation). A breadboard could make this task easier, but it's not strictly required.: fix breadboard to Doggy's back (upside down).

Build a leash keeping connectors together… or you can braid them ☺

WARNING: Using Doggy, TJBot requires more power than usual. Power it with 2.4A and everything will be fine!

Description	From TJBot	From Doggy's eyes	From Doggy's tail
Type of connectors	Female to Male	Male to Female	Male to Male
Pin 25 - Ground	Blue	Blue to Ground	Brown to Brown (Ground)
Pin 23 - GPIO 11 - Echo	Violet	Green to Echo	
Pin 21 - GPIO 09 - Trigger	Gray	Yellow to Trigger	
Pin 4 - 5V White	Orange to VCC	Red to Red (5V)	
Pin 24 - GPIO 08 Data servo motor (tail)	Black	Orange to Orange (Data servo motor)	

WARNING: pay attention to connect the right PINs to avoid damages to components.

Build tail (version #1). Get Servo arm and tape tail to it.

Build tail (version #2). Tail is a personality trait. Use your imagination and try different tails. In the following photo, I wrapped tail starting from servo arm to tail's end creating a rotation effect.

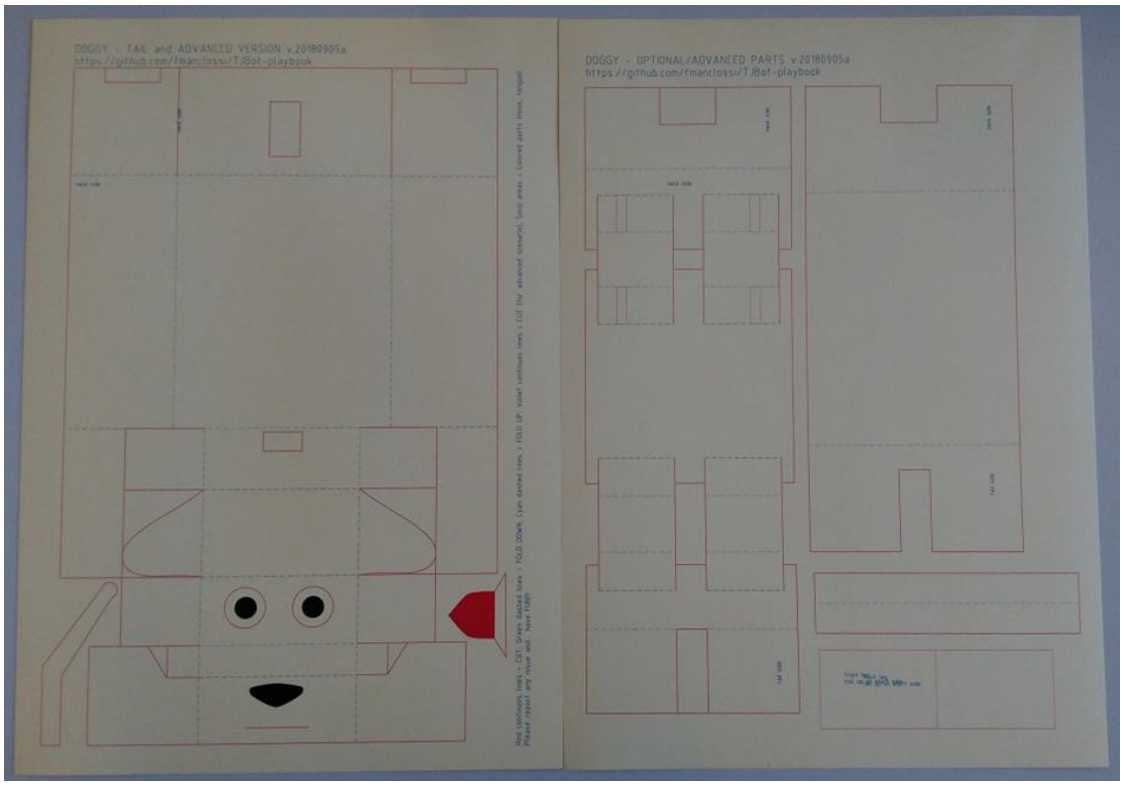
(optional) - print and cut standard belly component from Optional/Advanced Parts design sheet

Enjoy your Doggy Standard!!!

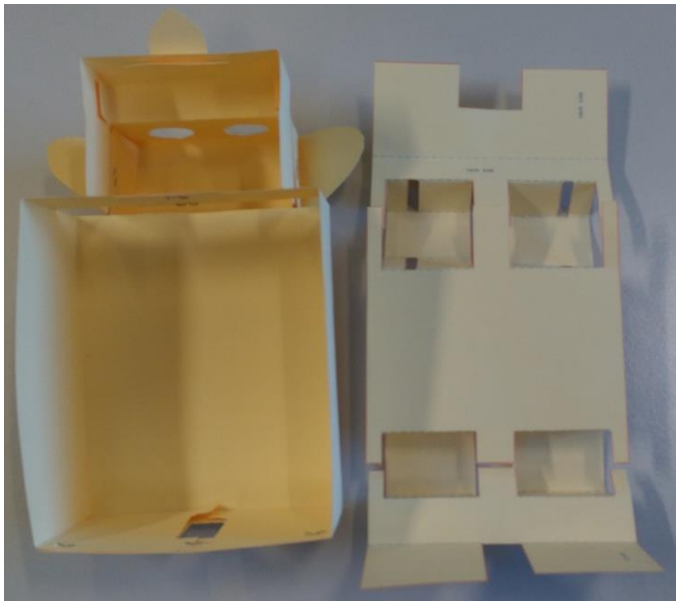
Please, share your experience on Doggy's House!!!

Mount Doggy Premium - a TJBot companion with superdog vision, a cute tail and two legs!

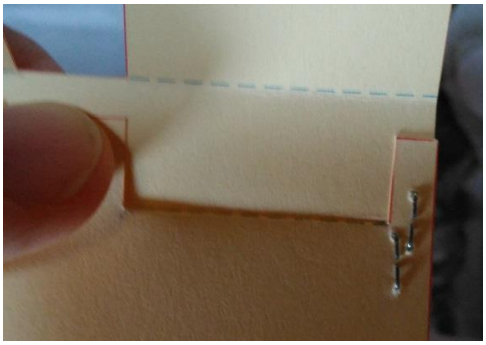
Get and Print/Copy DOGGY Premium design files

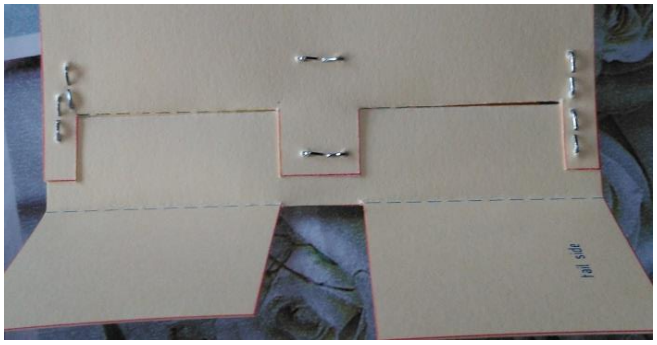


Cut and fold as described for previous Doggy's versions.

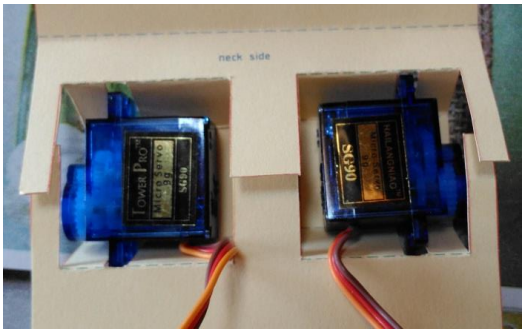


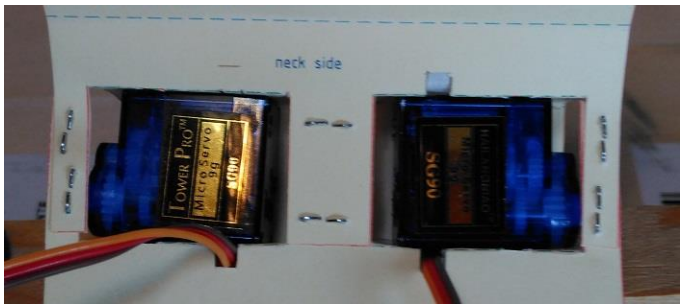
Clip hind legs



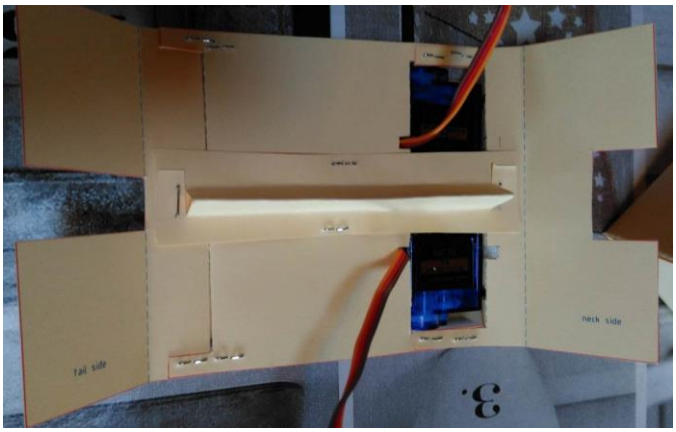


Mount servo motors for front legs and clip to fix them





(optional and in beta version) If you want to tighten up belly component (e.g. if you print it on 200gr/m² A4 paper), print, cut, fold and clip the beam component



Cut and Roll up front legs on shortest side. **TIP**: Fix leg to servo arm as shown in picture, leaving space to attach it to servo and fixing leg to the external part. As tail, legs are personality traits. Use your imagination and try different legs. In the following photo, I wrapped legs starting from servo arm to leg's end.

Wire HC-SR04 sensor and SG-90 servo motor (tail, right leg/arm, left leg/arm) to TJBot.

A breadboard is recommended. Fix breadboard to Doggy's belly.

Build a leash keeping connectors together... or you can braid them ☺

WARNING: Using Doggy, TJBot requires more power than usual. Power it with 2.4A and everything will be fine!

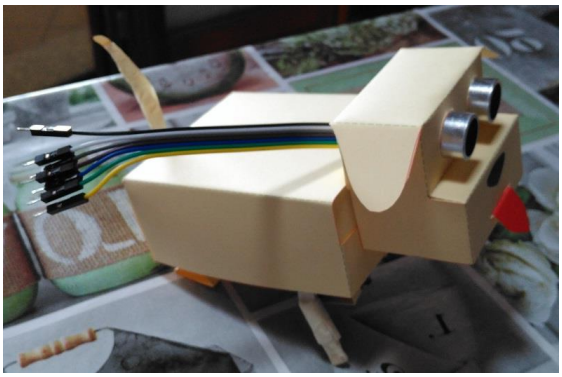
Description	From TJBot	From Doggy's eyes	From Doggy's tail
Type of connectors	Female to Male	Male to Female	Male to Male
Pin 25 - Ground	Blue	Blue to Ground	Brown to Brown (Ground)
Pin 23 - GPIO 11 - Echo	Violet	Green to Echo	
Pin 21 - GPIO 09 - Trigger	Gray	Yellow to Trigger	
Pin 4 - 5V White	Orange to VCC	Red to Red (5V)	
Pin 24 - GPIO 08 Data servo motor (tail)	Black	Orange to Orange (Data servo motor)	

Description	From TJBot	From Doggy's eyes	From Doggy's tail	From Doggy's left arm	From Doggy's Right arm
Type of connectors	Female to Male	Male to Female	Male to Male	Male to Male	Male to Male
Pin 38 - GPIO 20 - Data servo motor (right Doggy's arm)	Green				Violet to Orange (data servo motor)
Pin 40 - GPIO 21 - Data servo motor (left Doggy's arm)	Yellow			Violet to Orange (data servo motor)	
Pin 25 - Ground	Blue	Blue to Ground	Brown to Brown (Ground)	Brown to Brown (Ground)	Brown to Brown (Ground)
Pin 23 - GPIO 11 - Echo	Violet	Green to Echo			
Pin 21 - GPIO 09 - Trigger	Gray	Yellow to Trigger			
Pin 4 - 5V	White	Orange to VCC	Red to Red (5V)	Red to Red (5V)	Red to Red (5V)
Pin 24 - GPIO 08 - Data servo motor (tail)	Black		Orange to Orange (Data servo motor)		

WARNING: pay attention to connect the right PINs to avoid damages to components.

Build tail (version #1). Get Servo arm and tape tail to it.

Build tail (version #2). Tail is a personality trait. Use your imagination and try different tails. In the following photo, I wrapped the tail, starting from servo arm to tail's end creating a rotation effect.



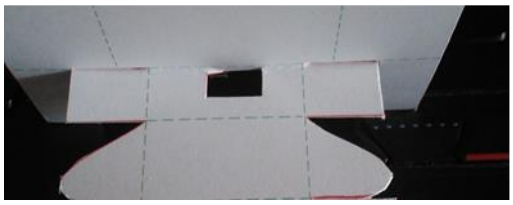
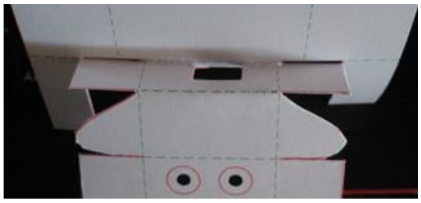
Enjoy your Doggy Premium!!!

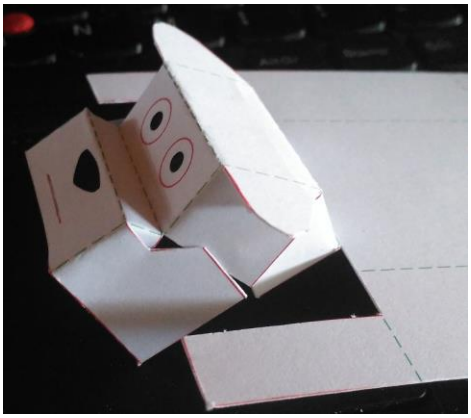
Please, share your experience on [Doggy's House](#)!!!

Mount MiniDoggy – the cutest tail in TJworld

Get and Print/Copy MiniDoggy design file and Cut all the red lines using a cutter and a scissor.

fold up or down green and green lines

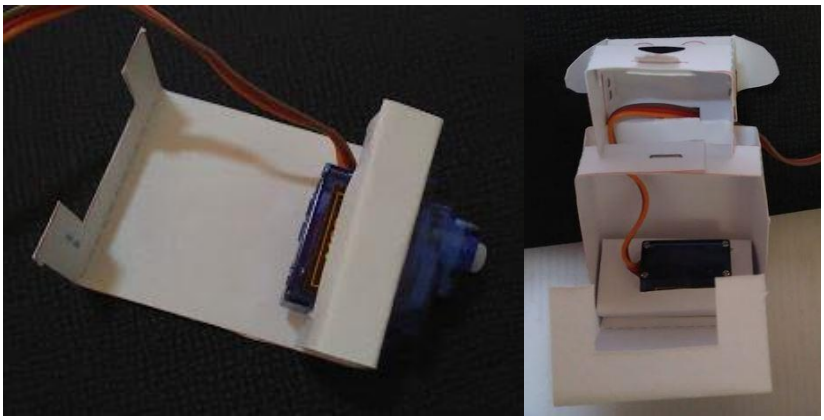


staple ears

Staple neck and back; mount servo motor (simplest version)

(optional, extended version) mount servo motor.

If you like, or if you printed MiniDoggy on standard printer paper, fold and mount MiniDoggy's belly



wire servo motor SG-90 to TJBot (no need for breadboard).

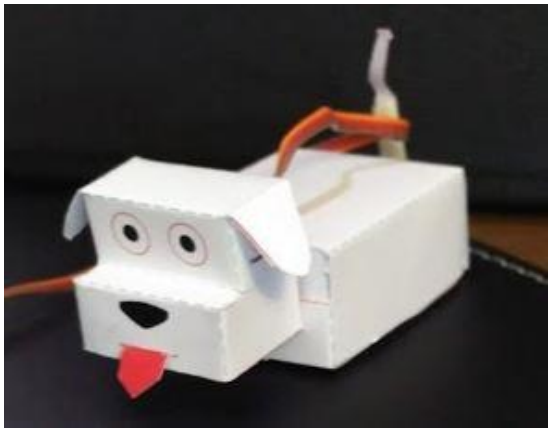
Build a leash keeping connectors together… or you can braid them ☺

Description	From TJBot	From MiniDoggy's tail
Type of connectors	Female to Male	Male to Male
Pin 25 - Ground	Blue	Violet to Brown (Ground)
Pin 4 - 5V	White	Red to Red (5V)
Pin 24 - GPIO 08 - Data servo motor (tail)	Black	Yellow to Orange (Data servo motor)

WARNING: pay attention to connect the right PINs to avoid damages to components.

Build tail (version #1). Get Servo arm and tape tail to it.

Build tail (version #2). Tail is a personality trait. Use your imagination and try different tails. In the following photo, I wrapped tail starting from servo arm to tail's end creating a rotation effect.



insert tongue and enjoy your MiniDoggy!!!

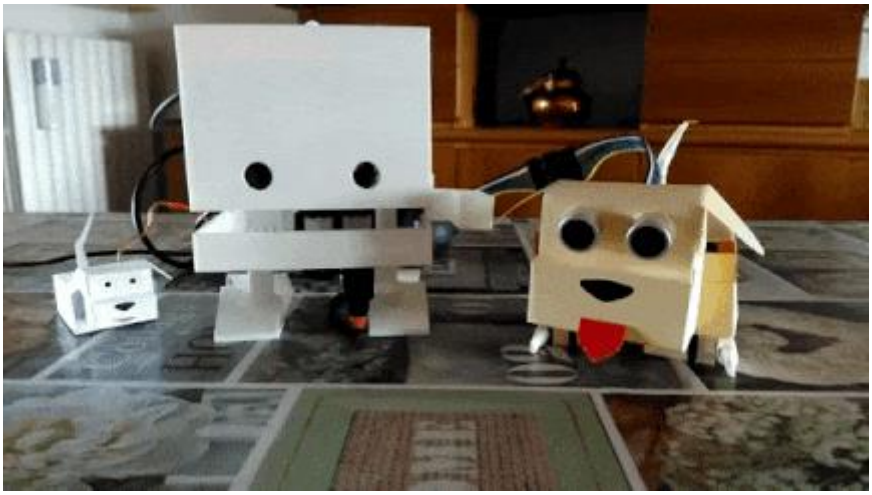
Please, share your experience on Doggy's House!!!

Instruction: Setup and test Doggy

First of all, if you never setup Node-Red environment on TJbot, follow JCBisson Node-Red nodes for TJBot - setup instructions.

Everything ok? Excellent! You can already follow other TJBot labs provided in TJBot Playbook github space!

But... **we want a Doggy**!!!



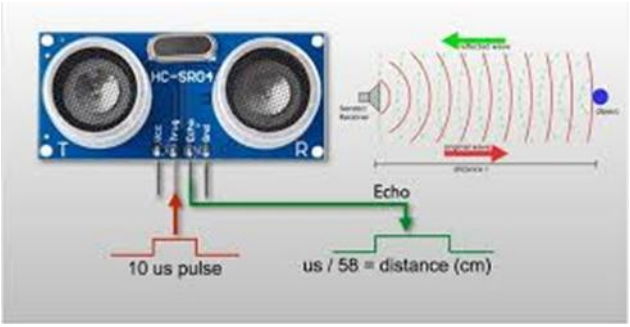
Before letting it run through digital fields, we have to test its features and tune up some little things.

Apply the following setup instructions to your version:

- Setup and Test Doggy superdog vision (HC-SR04)
- Setup and Test Doggy tail (SG-90)
- Setup and Test Doggy front legs (SG-90)

Instruction: Setup and Test Doggy superdog vision (HC-SR04)

Doggy is a superdog! It has a supervision powered by a very easy to use sensor named HC-SR04. This is a ultrasonic sonar. It emits a sound that travels through the air. If there is an object on its path, the sound will bounce back to the sensor that evaluate travel time to calculate how far is the object. HC-SR04 is able to detect object from 2cm to 400cm (1” to 13 feet).

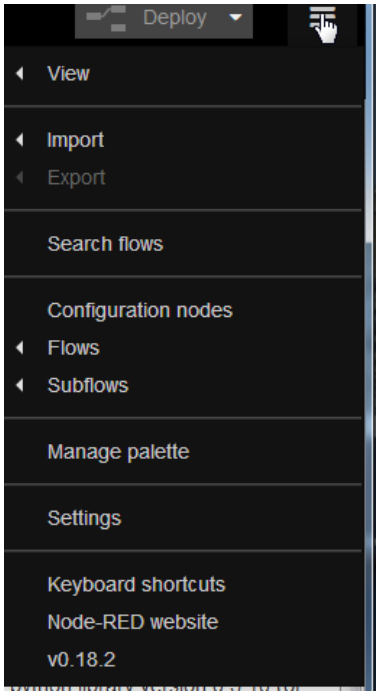


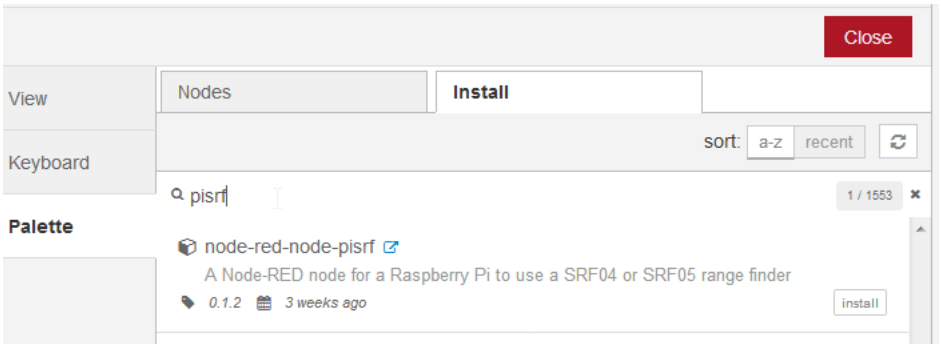
Using HC-SR04 in Node-Red is very easy! Follow wiring instructions for your Doggy version described in previous chapters.

Note: If you've changed PINs connections, modify the example according to your layout.

Setup instructions:

- point your browser to Node-Red URL - http://«TJBot_ip_address»:1880 and verify that TJBot and Node-red are fully functional
- install **node-red-node-pisrf** node to manage HC-SR04 and superdog vision with the following instructions:
- open Node-Red Dashboard.
- click on Menu, select MANAGE PALETTE
- choose INSTALL tab and search package inserting "pisrf" test in search field

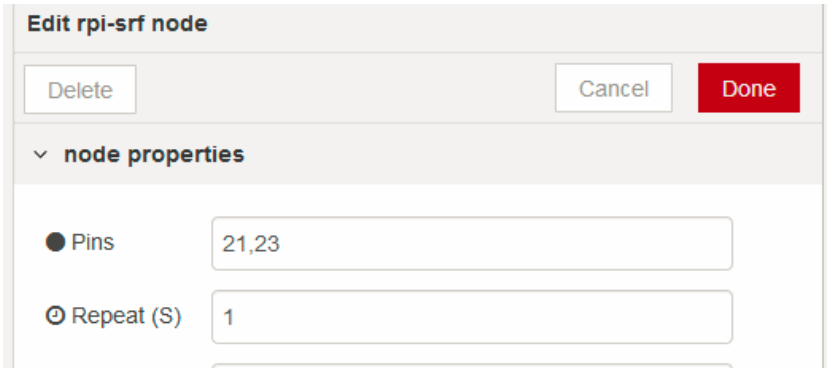




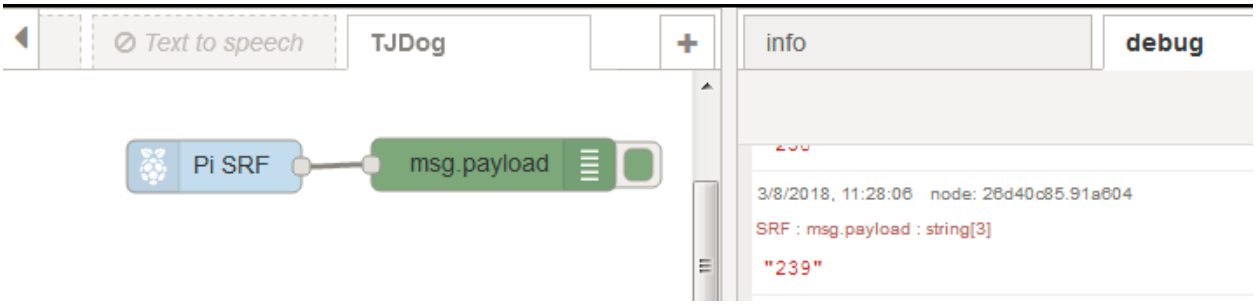
- push INSTALL button and wait successful installation.
- press CLOSE to return to Node-Red dashboard.

Test superdog vision creating the following flow using RPI-SRF and DEBUG nodes

- drag RPI-SRF node to working area; drag DEBUG node, too. Connect RPI-SRF to DEBUG.
- double-click RPI-SRF node and configure it as shown in picture.



- push DEPLOY button and verify that in the DEBUG area you'll get the distance (in centimetres) between Doggy's head and an obstacle.



That's all! We're ready to build a story with Doggy's superdog eyes!

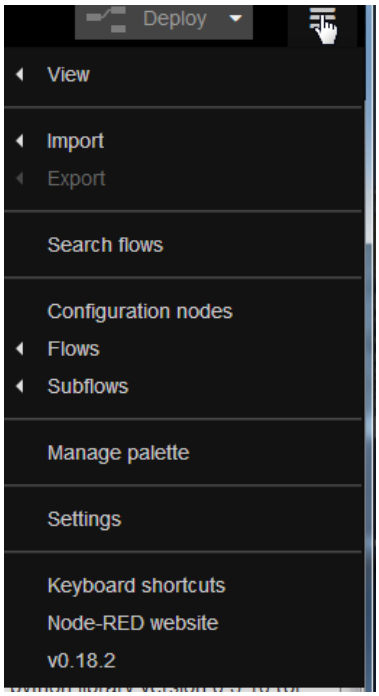
Instruction: Setup and Test Doggy tail (SG-90)

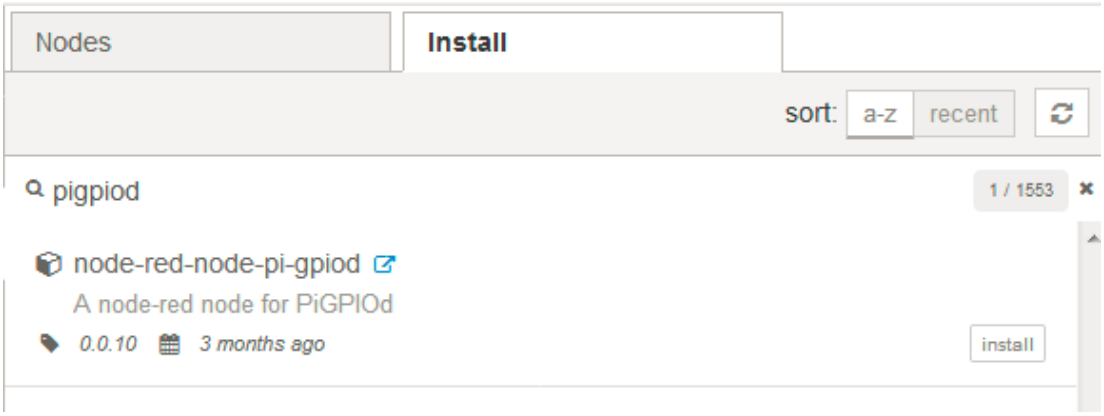
Doggy is a cute dog! It has tail that wags to show its happiness. A little servo motor (SG-90) moves the tail. Using SG-90 in Node-Red is very easy! Follow wiring instructions for your Doggy version.

Note: If you've changed PINs connections, modify the example according to your layout.

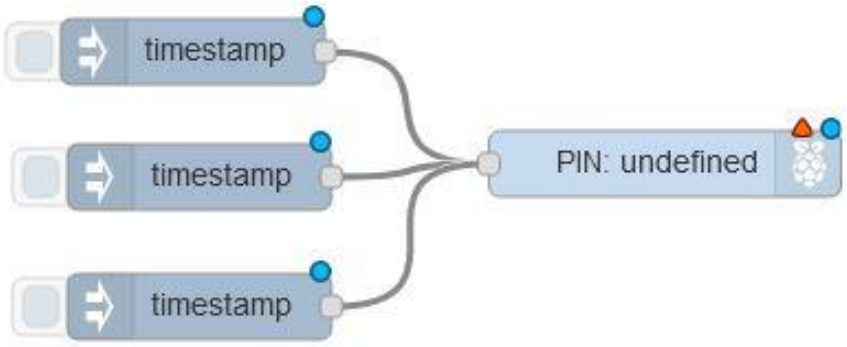
Setup instructions:

- point your browser to Node-Red URL - http://«TJBot_ip_address»:1880 and verify that TJBot and Node-red are fully functional
- install **node-red-node-pi-gpiod** node to manage HC-SR04 and superdog vision with the following instructions:
- open Node-Red Dashboard.
- click on Menu, select MANAGE PALETTE
- choose INSTALL tab and search package inserting "pigipiod" test in search field

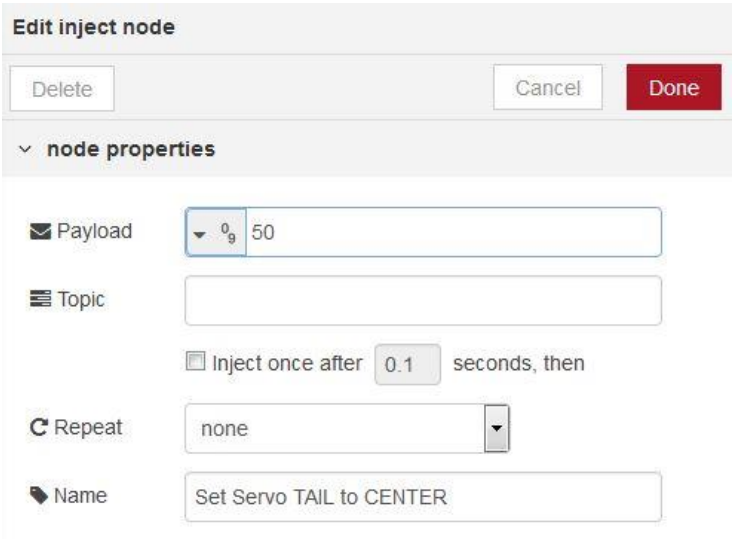




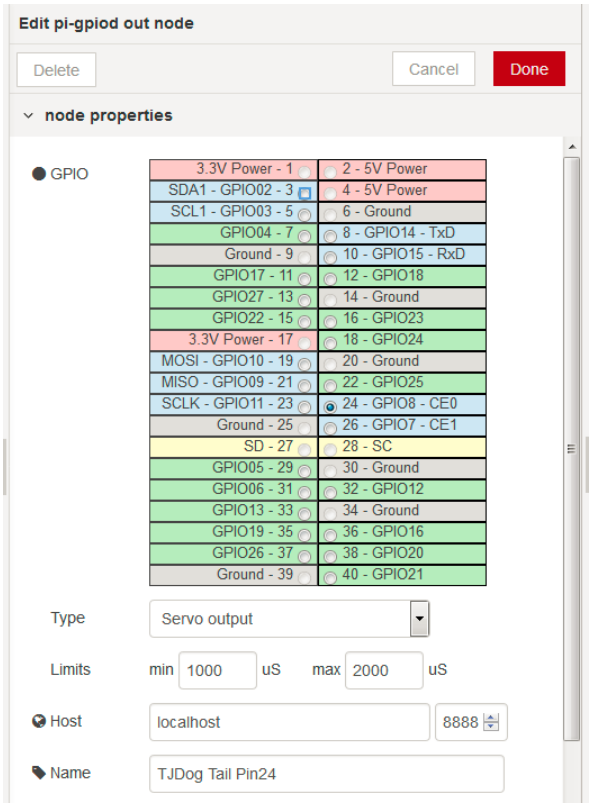
- push INSTALL button and wait successful installation.
- press CLOSE to return to Node-Red dashboard.
- take away Doggy's tail. We must set the right position to avoid damages to it.
- tune tail position creating the following flow using PI-GPIOD and INJECT nodes dragging three times INJECT node to working area (you'll get three nodes); drag PI-GPIOD node, too. Connect INJECT nodes to PI-GPIOD.



- double-click the first INJECT node and configure it as shown in the following picture. It changes name to "Set TAIL to CENTER"
- as previous step, configure other INJECT nodes with 0 and 100 integer value. New names will be "Set TAIL to RIGHT" and "Set TAIL to LEFT"
-



- double-click PI-GPIOD node and configure it as shown in following picture. Push DEPLOY button.



- Push "Set TAIL to CENTER" inject node. Servo motor should reach middle position. Attach tail to Doggy in RAISED position.
- Push "Set TAIL to LEFT" inject node. Does tail move to Doggy's left side? If not, verify that servo is mounted according with your Doggy's version.
- Push "Set TAIL to RIGHT" inject node. Does tail move to Doggy's right side? If not, verify that servo is mounted according with your Doggy's version.

That's all! We're ready to build a story with Doggy's cute tail!

Instruction: Setup and Test Doggy front legs (SG-90)

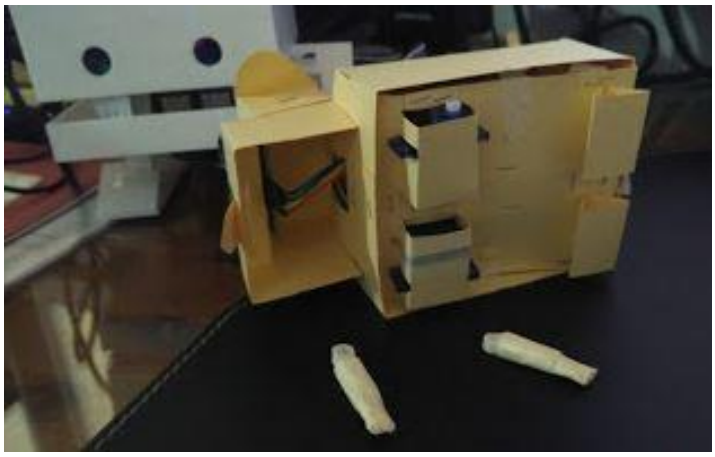
Doggy Premium is a strong dog! It can raise up on its front legs to intimidate unkown object and people. Two little servo motors (SG-90) powers front legs. Follow wiring instructions for your Doggy version.

Note: If you've changed PINs connections, modify the example according to your layout.

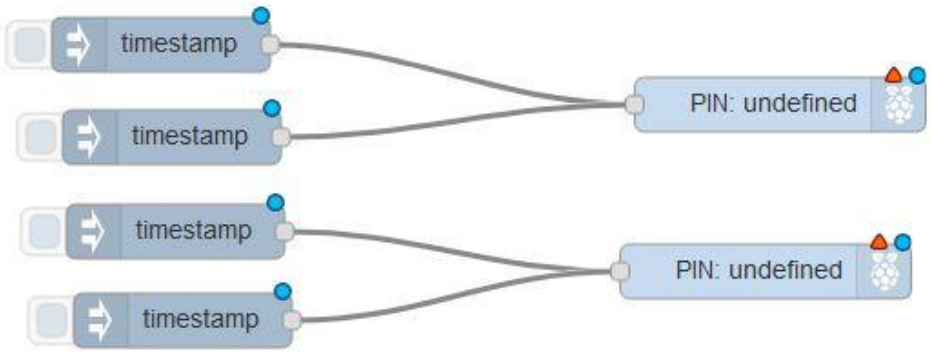
Prerequisite: install node-red-node-pi-gpiod as described in previous chapter.

Setup instructions:

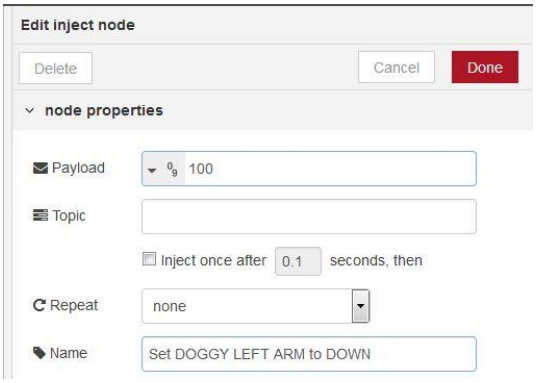
- point your browser to Node-Red URL - http://«TJBot_ip_address»:1880 and verify that TJBot and Node-red are fully functional
- **take away Doggy's legs. We must set the right position to avoid damages to them.**



- tune legs position creating the following flow using PI-GPIOD and INJECT nodes dragging four times INJECT node to working area (you'll get four nodes); drag twice PI-GPIOD node, too (You'll get two nodes). Connect two INJECT nodes per PI-GPIOD.



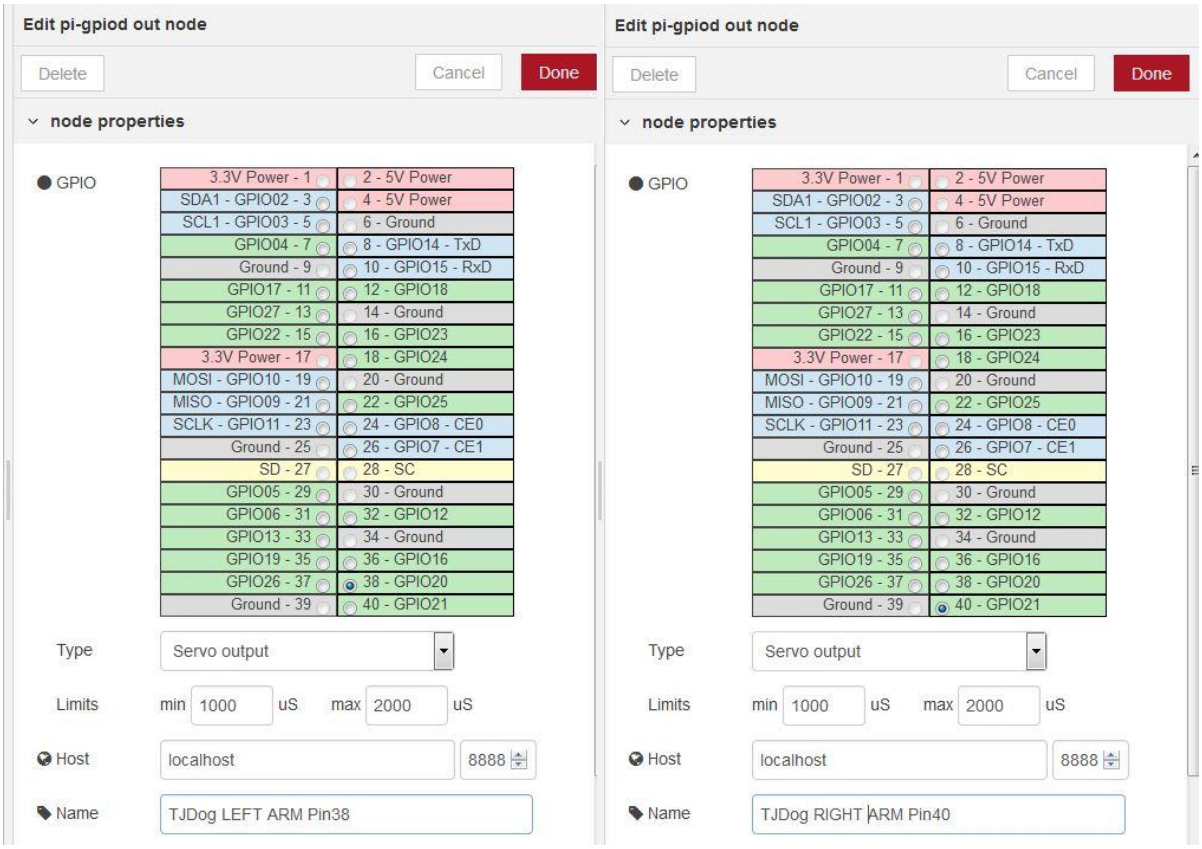
- double-click the first INJECT node and configure it as shown in picture. It changes name to "Set DOGGY LEFT ARM to DOWN“.

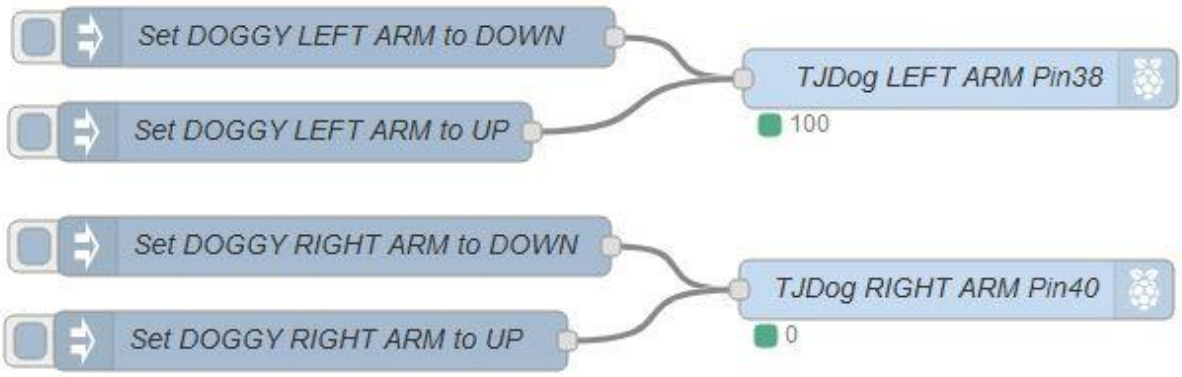


- configure other INJECT nodes as in following table

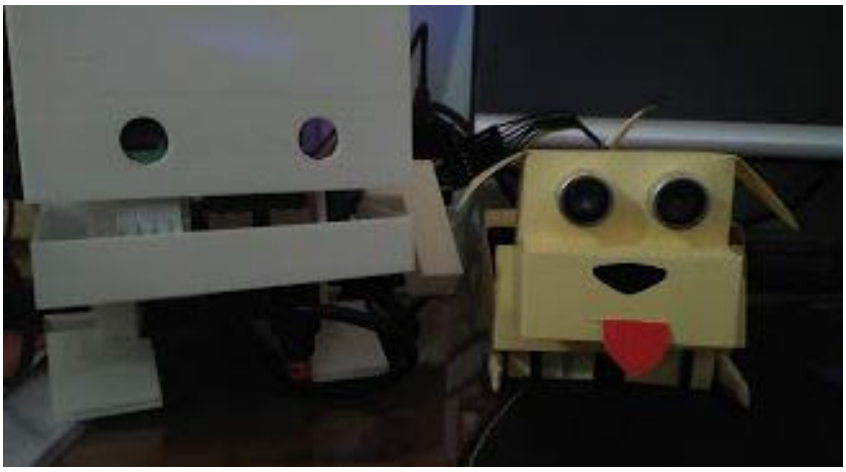
Node	Name	Value
Inject #2	Set DOGGY LEFT ARM to UP	0
Inject #3	Set DOGGY RIGHT ARM to DOWN	0
Inject #4	Set DOGGY RIGHT ARM to UP	100

- double-click PI-GPIOD nodes and configure them as shown in following pictures and push DEPLOY button.

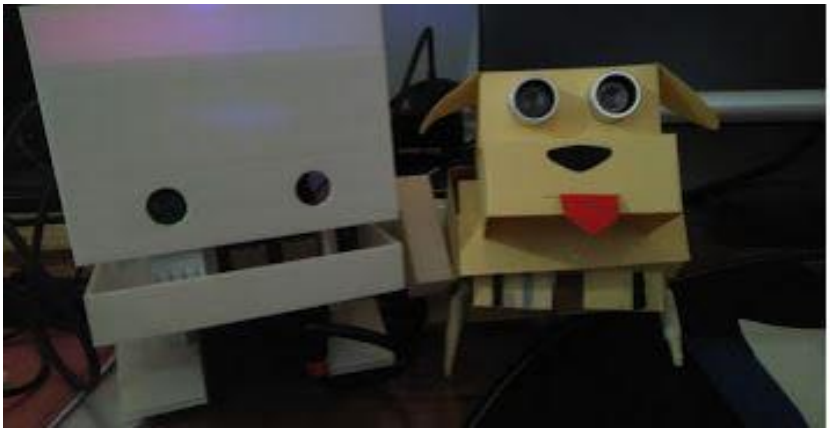




- Push "Set DOGGY LEFT ARM to UP" and "Set DOGGY RIGHT ARM to UP" inject nodes. Attach front legs to let Doggy to its belly.



- Push "Set DOGGY LEFT ARM to DOWN" and "Set DOGGY RIGHT ARM to DOWN" inject nodes. Does DOGGY raise up? If not, verify that servos are mounted as described in provided instructions.



That's all! We're ready to build a story with Doggy's front legs!

Tales of two Doggies

TJBot loves pets even they could be noisy. He/Her owns two dogs, Doggy and MiniDoggy.

Doggy is a very careful watchdog. Nothing and no one can pass in front of it without getting a strong barking.

Mini Doggy is more distracted one, it doesn't look at what happens around it but it's happy to play with TJBot.

In the following scenarios, we follow TJBot and its doggies in some funny stories learning how IBM solutions could be applied to real life examples.

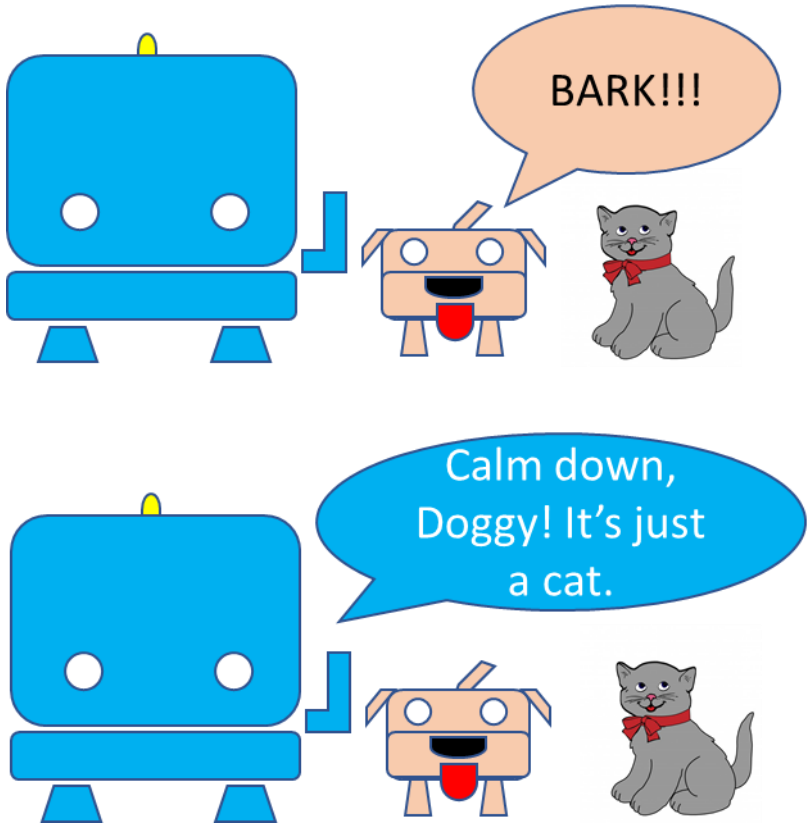
Every tale includes Doggy requirement, Skills, Age, Knowledge acquired and preparation steps.

Tales available:

- Doggy barks and wags its tail if an object is less than 30cm away
- Doggy likes his treats. It needs to eat healthier and start doing exercises.

We Need You! Submit your stories to expand Doggy's behaviour :)

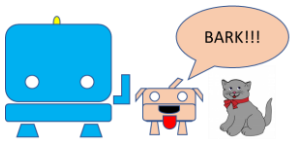
Instruction: Tale 1 - Nothing and no one can pass without getting a strong barking from Doggy. TJBot calms it, getting a cute tail wagging from Doggy.



Scene index:

- Tale1.Scene1 - Create a flow that makes Doggy say "BARK!" if an object is less than 30cm away
- Tale1.Scene2 – Make Doggy barking (a real barking!) if an object is less than 30cm away
- Tale1.Scene3 – Smart barking, just one time, please!
- Tale1.Scene4 – Doggy wags its tail
- Tale1.Scene5 - TJBot wakes up when Doggy starts barking and calms down Doggy recognizing the object

Instruction: Tale1.Scene1 - Doggy say “BARK!” if an object is less than 30cm away

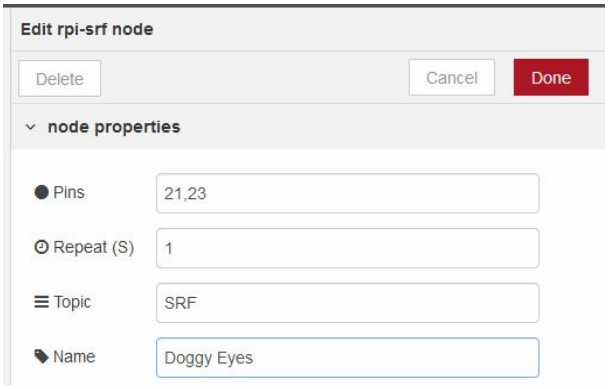


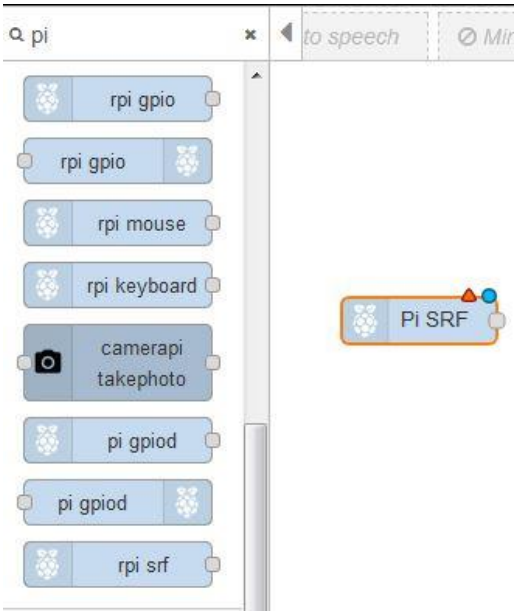
Doggy is a very careful watchdog. Nothing and no one can pass without getting a strong barking.

- **Requirement**: TJBot, Doggy (Basic, Standard or Premium) – already setup and tested
- **Skill**: Node-Red (basic level)
- **Age**: 8+
- **Knowledge acquired**: Sonar sensor usage, test condition, set value to variable, debugging
- **Preparation steps** (usually performed by IBMers, Teachers…):
  - point your browser to Node-Red Dashboard - http://«TJBot_ip_address»:1880
  - verify that TJBot is fully functional and node-red is working as expected.
  - disable every existing Flow (double-click each label, set Status to “Disabled”, press “Done” to close)

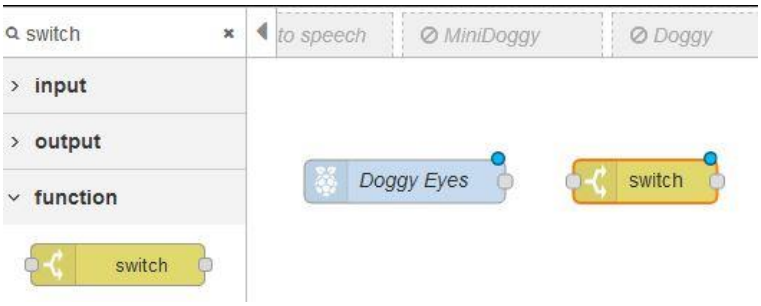
Instructions:

- point your browser to Node-Red Dashboard - http://«TJBot_ip_address»:1880
- create a new Flow 
- double-click Flow label and rename it to “Tale1.Scene1”. Press “Done” to close.
- drag PI-SRF node to the working area
- double-click PI-SRF node and configure it

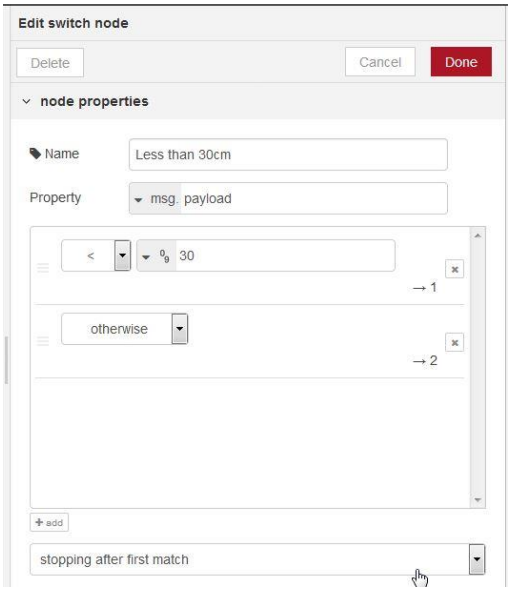


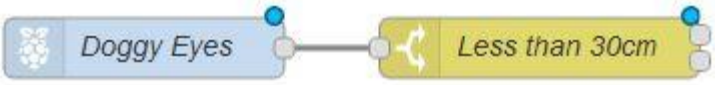


- drag a SWITCH node to the working area



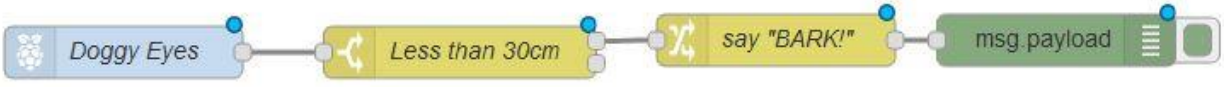
- double-click SWITCH node and configure it as shown in the following picture
- drag a line from the output of the first node to the input of the second one





- drag a DEBUG node to the working area

- connect its input to the output of CHANGE node



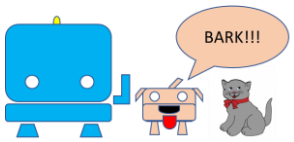
- press DEPLOY button and switch to “debug” tab (if not already highlighted)

What happen when you put an object near to Doggy’s eyes?

Excellent! But we want a strong bark not a text on dashboard.

We get it in the next scene!

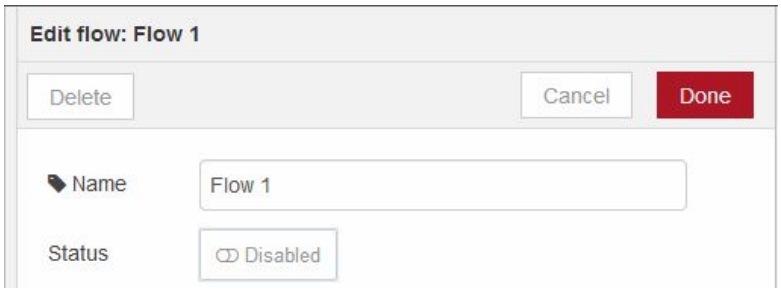
Instruction: Tale1.Scene2 –Doggy barks if an object is less than 30cm away



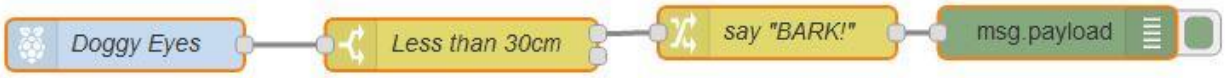
- **Requirement**: TJBot, Doggy (Basic, Standard or Premium) – already setup and tested
- **Skill**: Node-Red (basic level)
- **Age**: 8+
- **Knowledge acquired**: Sonar sensor usage, test condition, set value to variable, debugging, play audio
- **Preparation steps:**
  - point your browser to Node-Red Dashboard - http://«TJBot_ip_address»:1880
  - verify that TJBot is fully functional and node-red is working as expected (TJBot credentials, too).
  - verify that Tale1.Scene1 is enabled on dedicated Flow
  - get a good sound file (WAV format) with a dog barking (e.g. dogbark2.wav)
  - upload bark sound file to TJBot Desktop using VNC file transfer feature, an USB Key or whatever you like (e.g. /home/pi/Desktop/dogbark2.wav). P.S: You can download it directly using TJBot browser :D
  - test TJBot speaker with bark audio file

Instructions:

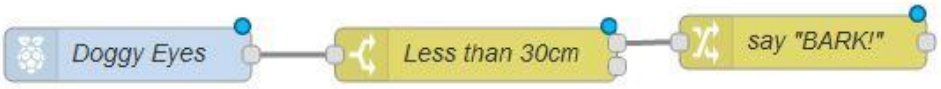
- point browser to Node-Red Dashboard http://«TJBot_ip_address»:1880
- create a new Flow 
- double-click Flow label and rename it to “Tale1.Scene2”. Press “Done” to close.



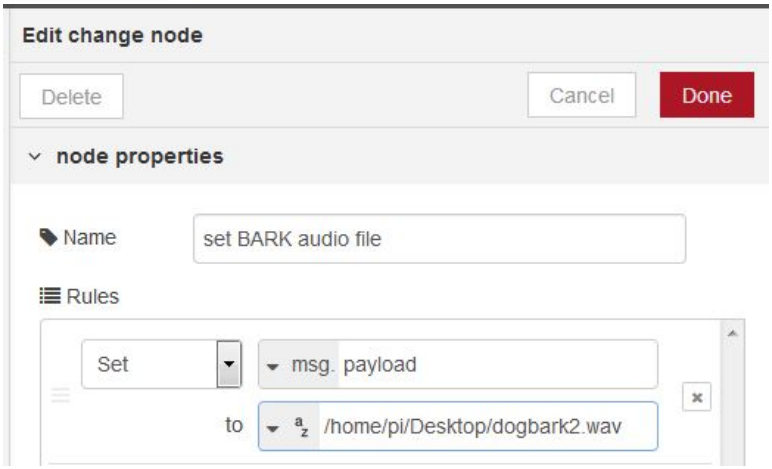
- double-click “Tale1.Scene1” Flow label and set status to “Disabled”. Press “Done” to close.
- select all nodes in “Tale1.Scene1” Flow drawing an area over them



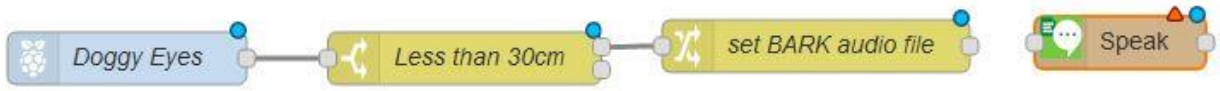
- press CTRL+C keys to copy.
- click on “Tale1.Scene2” Flow label
- press CTRL+V keys to paste nodes. When you find the best place to paste them, right click to save positions.
- delete DEBUG node (click on it and press DEL key)
-



- double-click SWITCH node and modify it as shown in the following picture

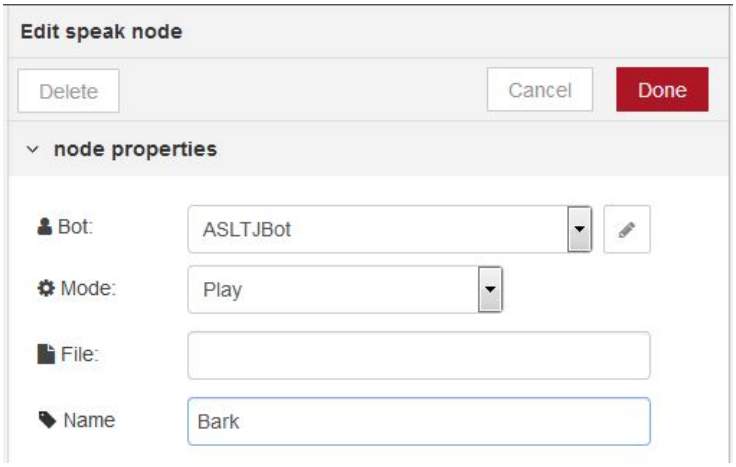


- drag SPEAK node (from TJBot palette) to the Working area



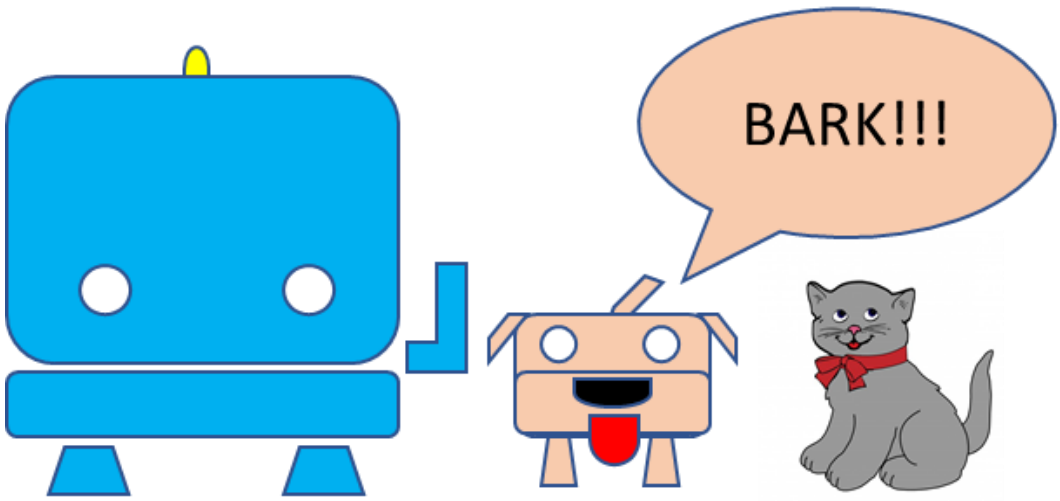
- double-click SPEAK node, select BOT according to your environment, set MODE to “Play” and set NAME to “Bark”

- connect it to CHANGE node and press DEPLOY button



Do you get a strong barking when you put an object near to Doggy’s eyes? If not, verify prerequisites (e.g. audio power turned on…)

What happen if the object stays near Doggy?



Congratulations!

You’re ready to move to the next level! Let’s make Doggy a little smarter!

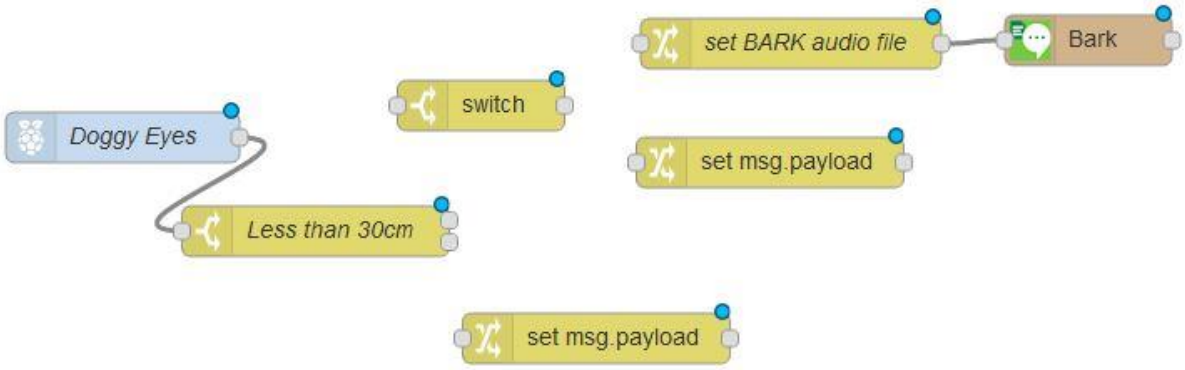
Instruction: Tale1.Scene3 – Smart barking, just one time, please!

- **Requirement**: TJBot, Doggy (Basic, Standard or Premium) – already setup and tested
- **Skill**: Node-Red (basic level)
- **Age**: 8+
- **Knowledge acquired**: Sonar sensor usage, test condition, set value to variable, debugging, play audio, use global variables
- **Preparation steps:**
  - point your browser to Node-Red Dashboard - http://«TJBot_ip_address»:1880
  - verify that TJBot is fully functional and node-red is working as expected (TJBot credentials, too).
  - verify that Tale1.Scene2 is enabled on dedicated Flow
- **Level:** some task could be tricky but can be performed under adult's guidance

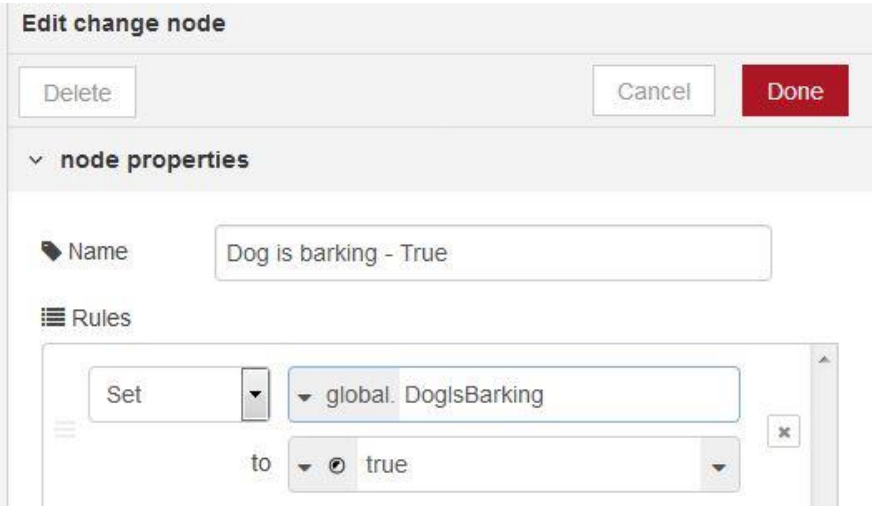
Instructions:

- point your browser to Node-Red Dashboard - http://«TJBot_ip_address»:1880
- create a new Flow and rename it to "Tale1.Scene3". Press "Done" to close.
- double-click "Tale1.Scene2" Flow label and set status to "Disabled". Press "Done" to close.
- select all nodes in "Tale1.Scene2" Flow drawing an area over them
- press CTRL+C keys to copy.
- click on "Tale1.Scene3" Flow label
- press CTRL+V keys to paste nodes. When you find the best place to paste them, right click to save positions.
- delete connection between SWITCH and CHANGE nodes, and make space in the flow

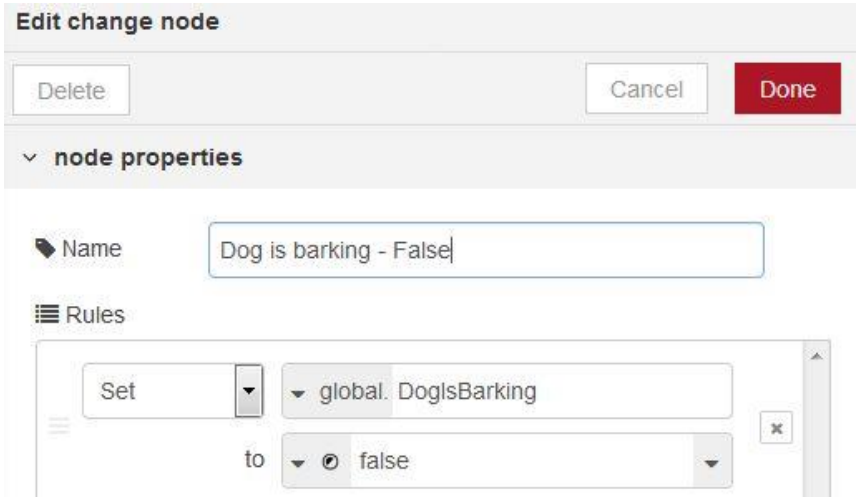
- drag twice CHANGE node to Working area. Drag SWITCH node, too.



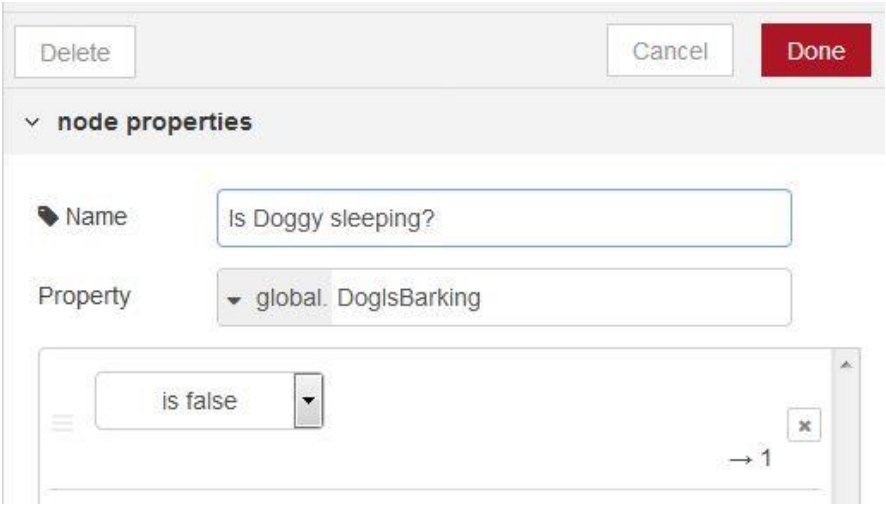
- double-click upper new CHANGE node and configure it. Press “Done” to close.



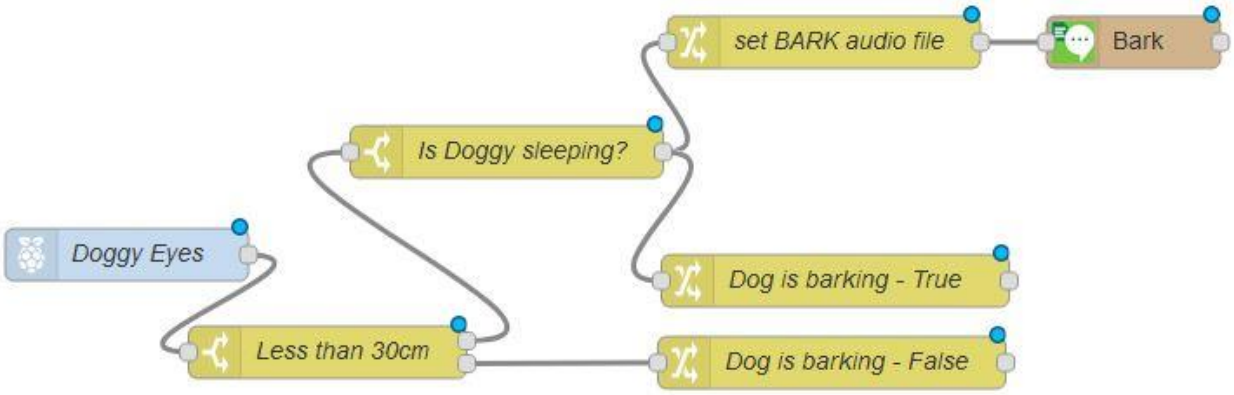
- double-click other new CHANGE node and configure it:



- double-click new SWITCH node and configure it as follow. Press “Done” to close.



- *(if needed, rearrange nodes to make space and) connect as show in picture and press DEPLOY button



What happen when an obstacle reaches Doggy? As previously, Doggy start barking.
What happen if the object stays near Doggy? What needs to happen to let Doggy barking again?

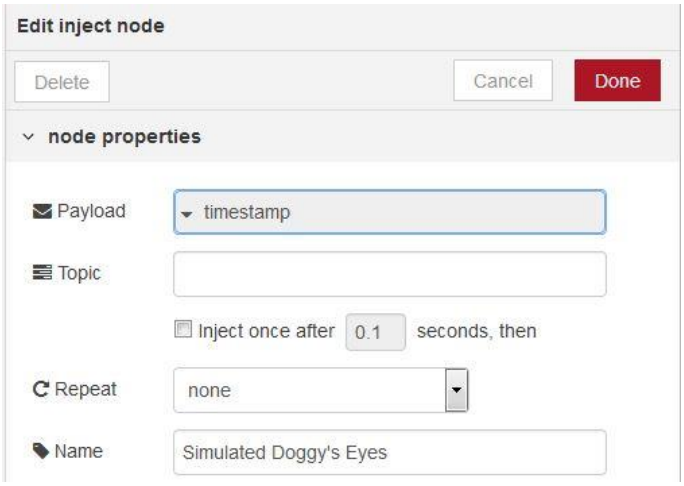
Awesome! Doggy is growing up smarter, step by step! It's time to wag its tail!

Instruction: Tale1.Scene4 – Doggy wags its tail!

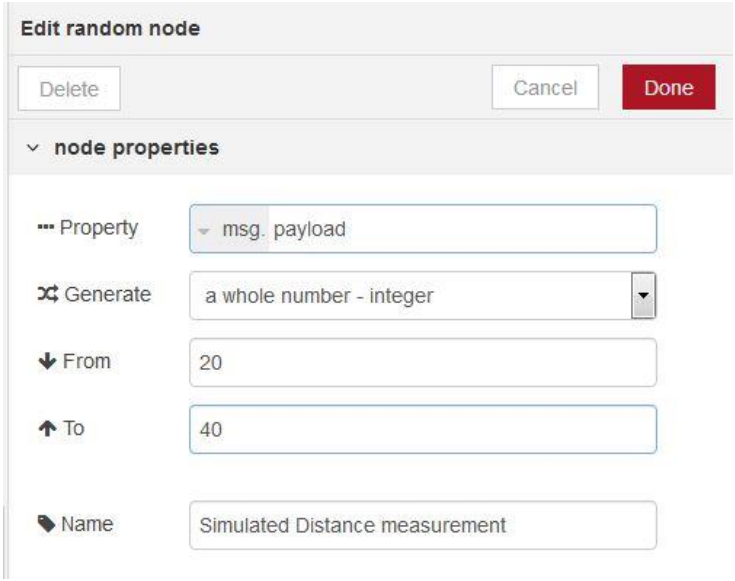
Instructions:

- **Requirement**: TJBot, MiniDoggy or Doggy (Basic, Standard or Premium) – already setup and tested
- **Skill**: Node-Red (basic level)
- **Age**: 8+
- **Knowledge acquired**: Servo motor and Sonar sensor usage, test condition, set value to variable, use random values, create loop
- **Preparation steps:**
  - point your browser to Node-Red Dashboard - http://«TJBot_ip_address»:1880
  - verify that TJBot is fully functional and node-red is working as expected (TJBot credentials, too).
  - verify if LOOPTIMER is installed. If not, install node-red-contrib-looptimer via “Manage Palette”
  - (Doggy version only) verify that Tale1.Scene3 is enabled and working on dedicated Flow
- **Level:** some task could be tricky but can be performed under adult’s guidance

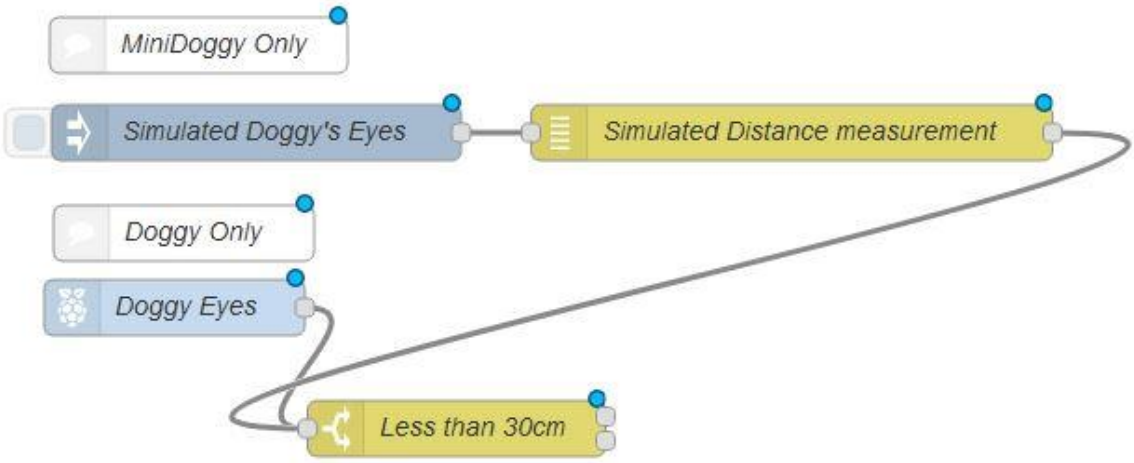
- point your browser to Node-Red Dashboard and create a new Flow. Rename it to “Tale1.Scene4”. Press “Done” to close.
- (Doggy version only) double-click “Tale1.Scene3” Flow label and set status to “Disabled”. Press “Done” to close.
- (Doggy version only) select “Doggy Eyes” and “Less than 30cm” nodes in “Tale1.Scene3” Flow. Press CTRL+C keys to copy them. Click on “Tale1.Scene4” Flow label. Press CTRL+V keys to paste them.
- (MiniDoggy only) drag INJECT node to Working area. Double-click it and configure as in picture



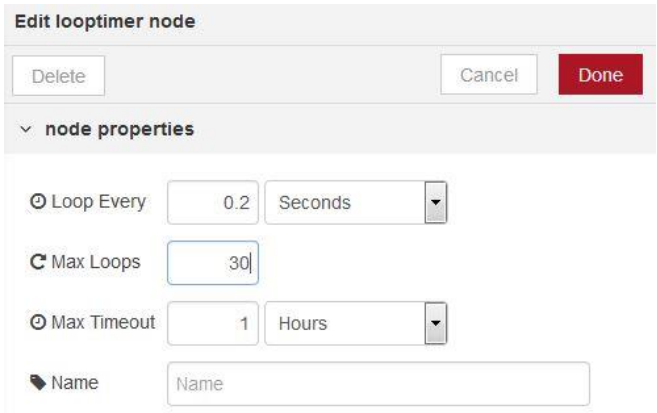
- (MiniDoggy only) drag RANDOM node to Working area. Double-click it and configure as in following picture.



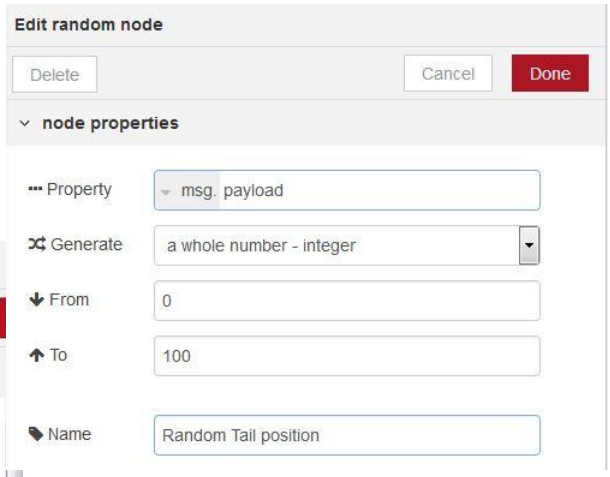
- Rearrange and connect nodes



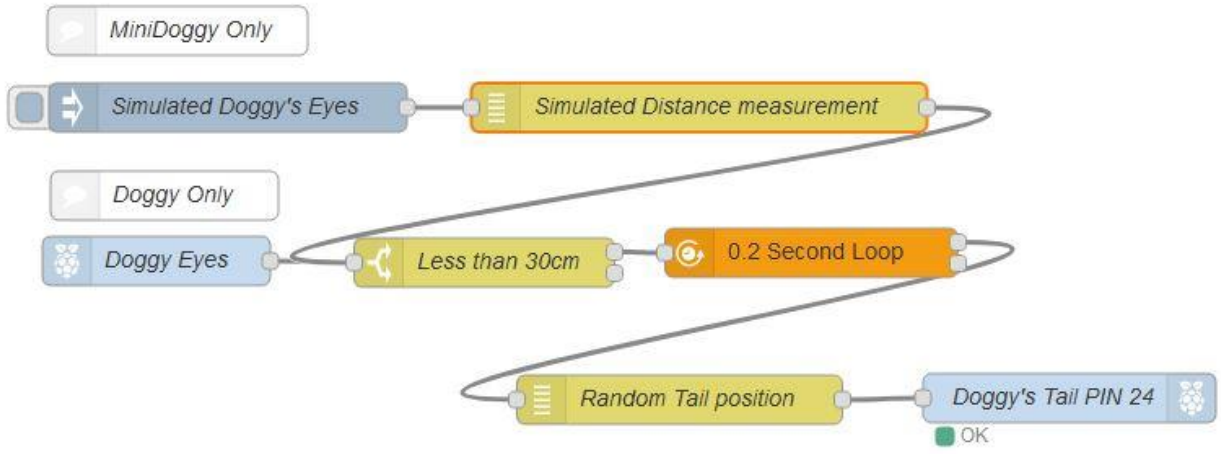
- drag LOOPTIMER node to Working area. Configure it to fire 30 times, every 0.2 seconds.



- drag RANDOM node to Working area. Configure it to send an integer between 0 and 100.
- drag PI-GPIOD node to Working Area and configure it to PIN 24 (GPIO 8)



- connect and rearrange, if needed, as show in the following picture. Press DEPLOY button



(Doggy version only)

What happen when an obstacle reaches Doggy?

What happen if the object stays near Doggy?

(MiniDoggy version only)

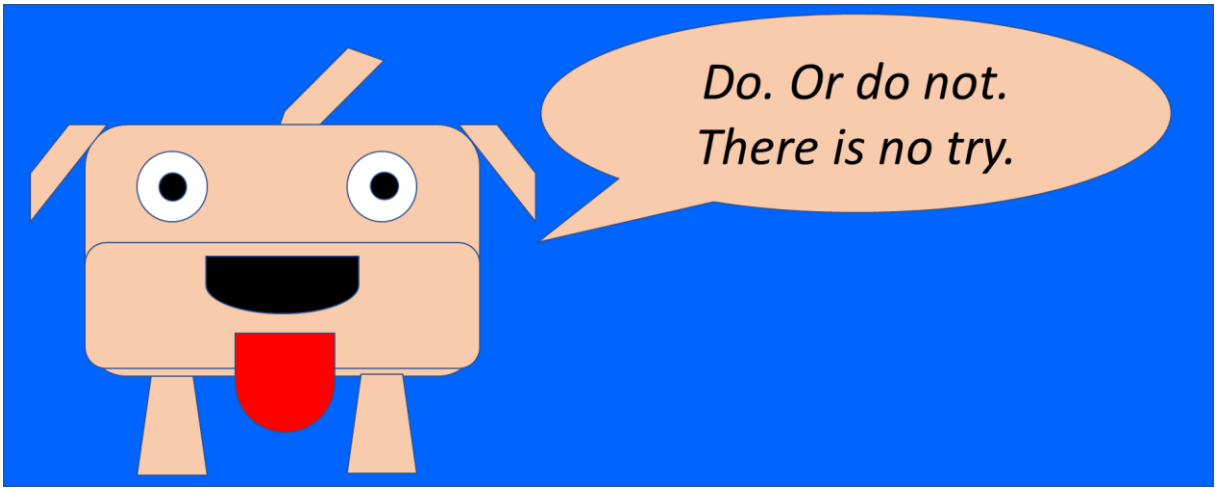
What happens when you press “Simulated Doggy’s Eyes” INJECT node?

Why the tail wags only some time and not always?

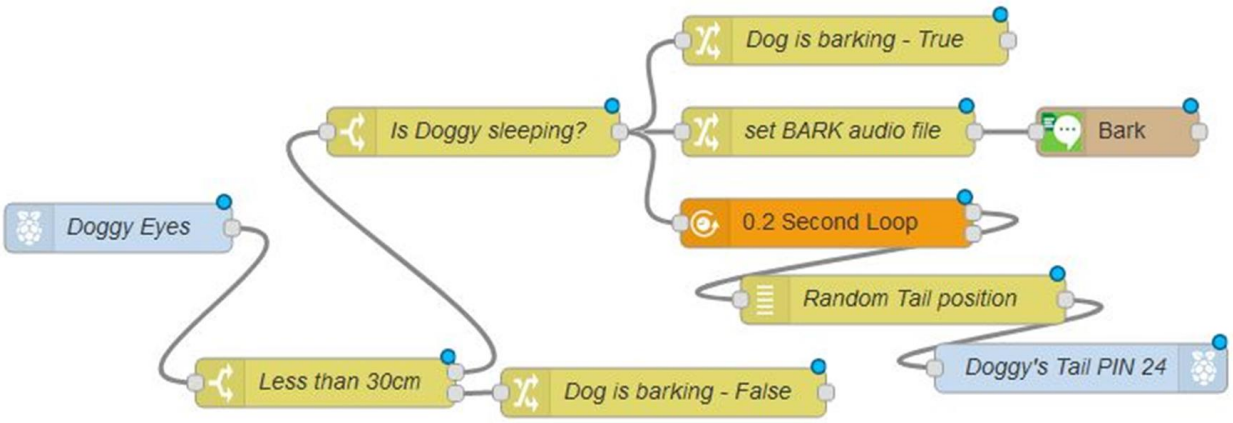
HINT: add a debug node to show “Simulated Distance Measurement” output. Call it “Distance” and press DEPLOY. Press again “Simulated Doggy’s Eyes” INJECT node

Fantastic! Doggy is growing up smarter, step by step!

Challenge T1C1: Add scene3 elements to make Doggy barking and shaking tail when an object is closer than 30cm.



ONE OF MANY solutions is on the next page

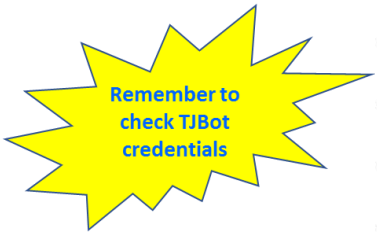


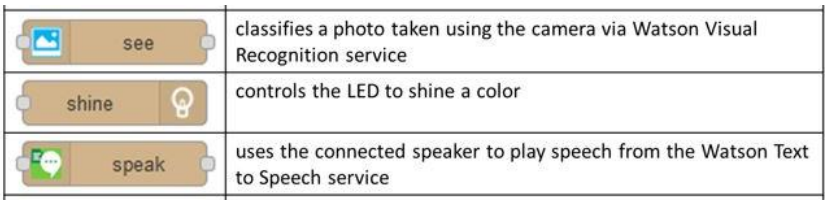
ONE OF MANY solutions

Instruction: Tale1.Scene5 – TJBot wakes up when Doggy starts barking and calms it down

- **Requirement**: TJBot, MiniDoggy or Doggy (Basic, Standard or Premium) – already setup and tested
- **Skill**: Node-Red (basic level)
- **Age**: 10+
- **Knowledge acquired**: Servo motor and Sonar sensor usage, test condition, set value to variable, TJBot capabilities (Text to Speech, Visual Recognition, Wave)
- **Preparation steps:**
  - point your browser to Node-Red Dashboard - http://«TJBot_ip_address»:1880
  - verify that TJBot is fully functional and node-red is working as expected (TJBot credentials, too).
  - verify if LOOPTIMER is installed from previous Scenes. If not, install node-red-contrib-looptimer via "Manage Palette"
  - (Doggy version only) verify that Tale1.Scene3 is enabled and working on dedicated Flow
- **Level:** some task could be tricky but can be performed under adult's guidance

There are several steps blocks in this scene according to Doggy version used

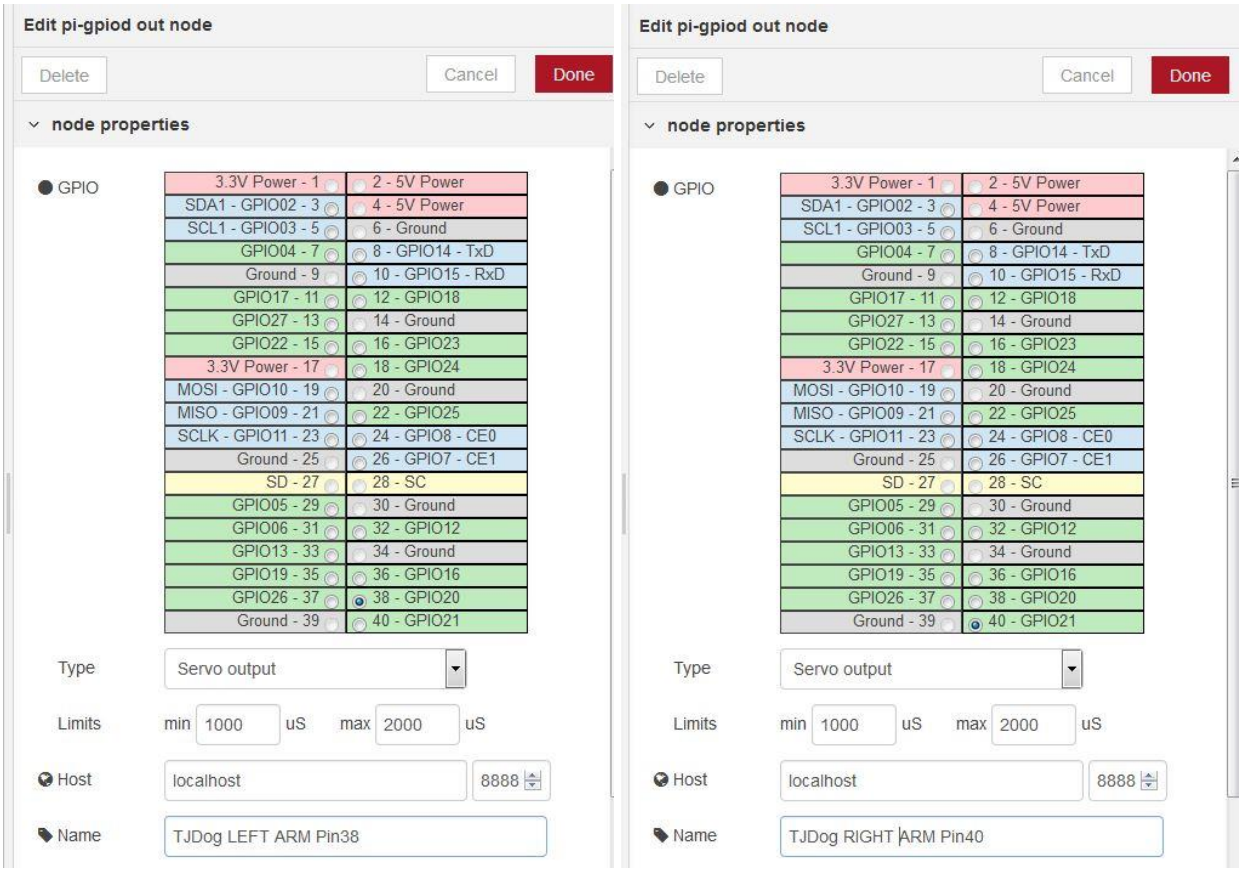




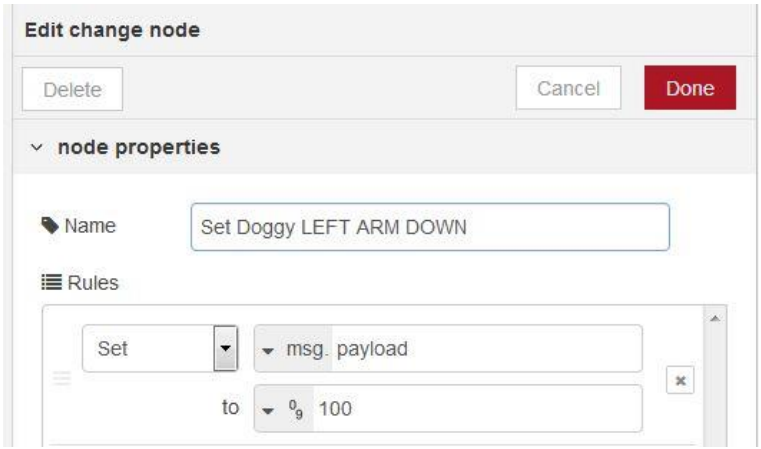
Steps (part 1):

- point your browser to Node-Red Dashboard - http://«TJBot_ip_address»:1880
- create a new Flow
- double-click Flow label and rename it to "Tale1.Scene5".
- Press "Done" to close.
- disable any other active flow.

- select all nodes in “Tale1.Scene3” Flow. Press CTRL+C keys to copy them.
- Click on “Tale1.Scene5” Flow label. Press CTRL+V keys to paste them.
- drag twice PI-GPIOD node to working area (you'll get two nodes)
- double-click PI-GPIOD nodes and configure them as shown in pictures on the right.

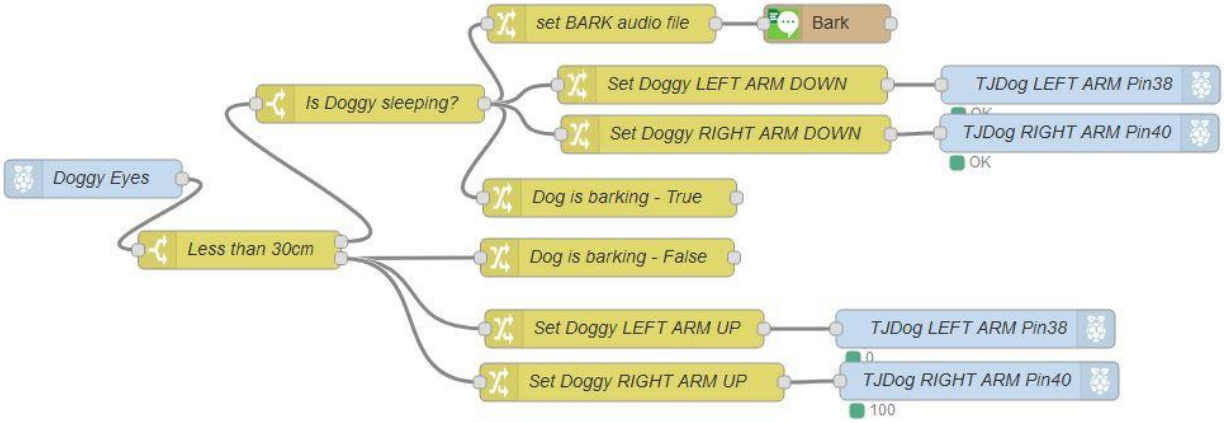


- duplicate PI_GPIOD nodes (copy and paste)
- drag four times CHANGE node to working area and configure them as shown in picture following the table provided



Node	Name	Value
Change #1	Set DOGGY LEFT ARM to DOWN	100
Change #2	Set DOGGY LEFT ARM to UP	0
Change #3	Set DOGGY RIGHT ARM to DOWN	0
Change #4	Set DOGGY RIGHT ARM to UP	100

- connect and rearrange, if needed, as show in the following picture.
- press DEPLOY button.

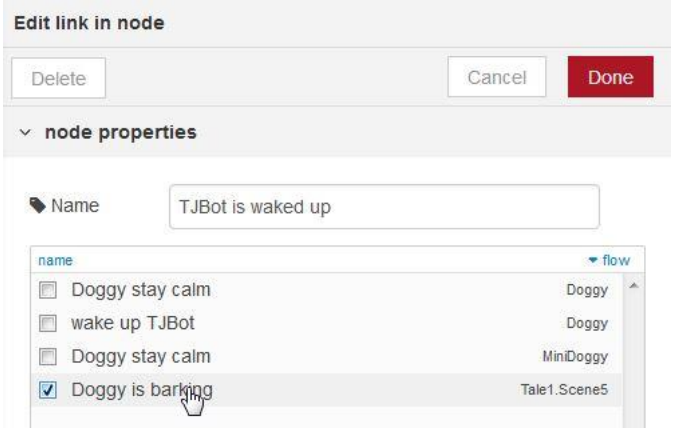


(HINT: use comment nodes - Things can grow up very fast and one comment can save your life)

What happens when a moving object is in front of Doggy?

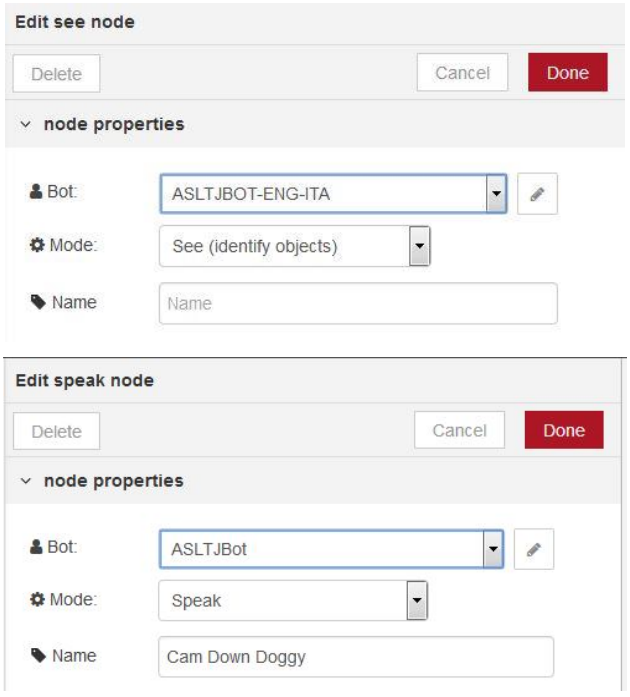
Steps (part 3): Let's TJBot take on the stage!

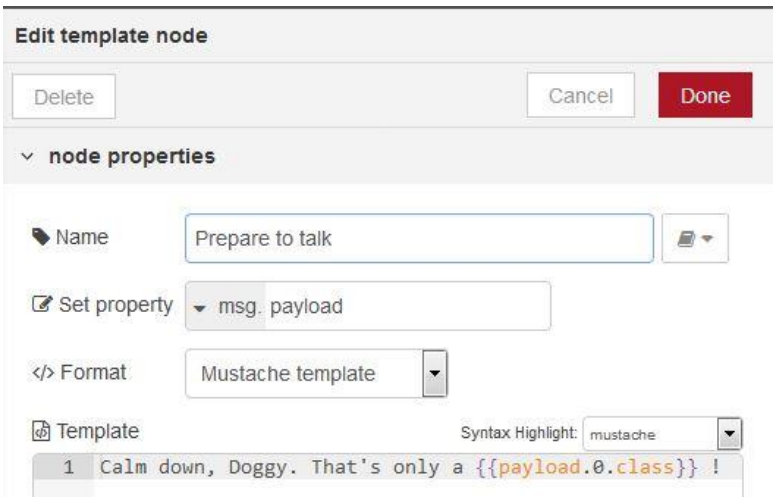
- drag once OUTPUT LINK node to working area beside SPEAK node
- drag one INPUT LINK node to working area below the flow. Double-click it and name it "Doggy is barking"
- double-click OUTPUT NODE and link it to "Doggy is barking"



- drag TJBot's SEE and SPEAK, TEMPLATE (and DEBUG) nodes from Palette and connect them as in picture

- configure SEE, SPEAK and TEMPLATE nodes as following





This is the code inserted in FUNCTION to copy and paste:

```
Calm down, Doggy.
That's only a
{{payload.0.class}}!
```

- drag twice TJBot's SHINE node to Working area. Connect and configure them as shown in following picture.
- press DEPLOY button.

What happens moving object in front of Doggy?

What happens if an object goes away and then back while TJBot is talking?

Probably you (or your students) noticed that Doggy and TJBot try to identify multiple objects mixing barking and speaking.

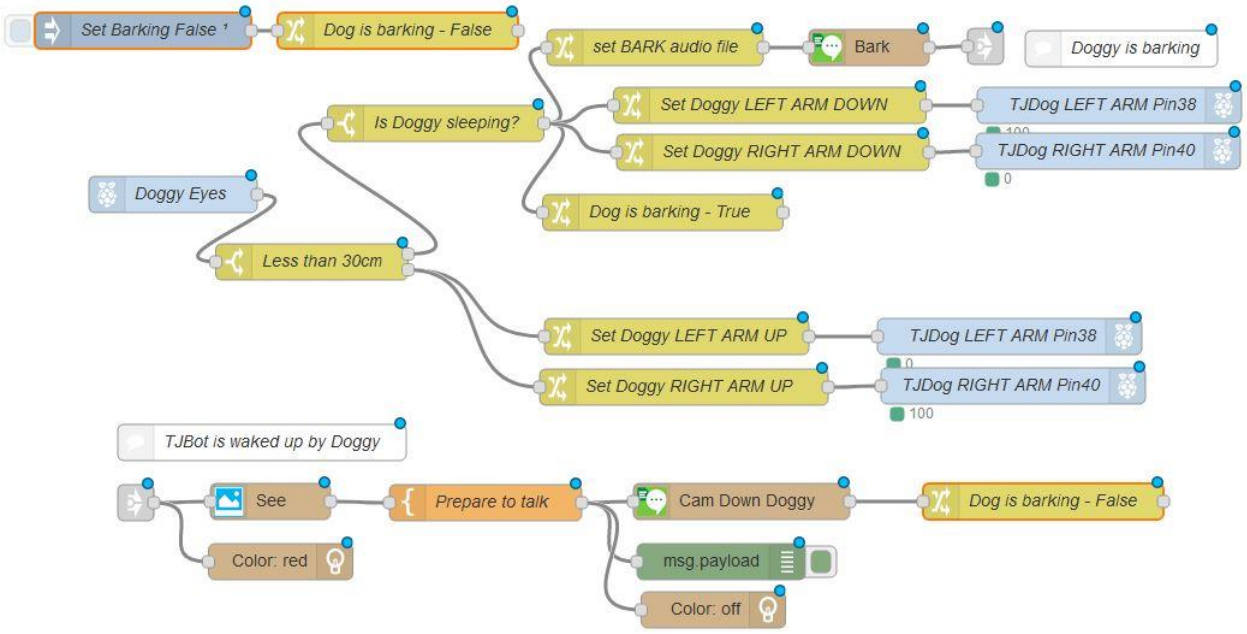
This is bad ☹

How can you fix this behaviour?

(*IBMers/Teachers hint*: *this is the same problem addressed in Tale2.Scene2*

Look at it to get the details)

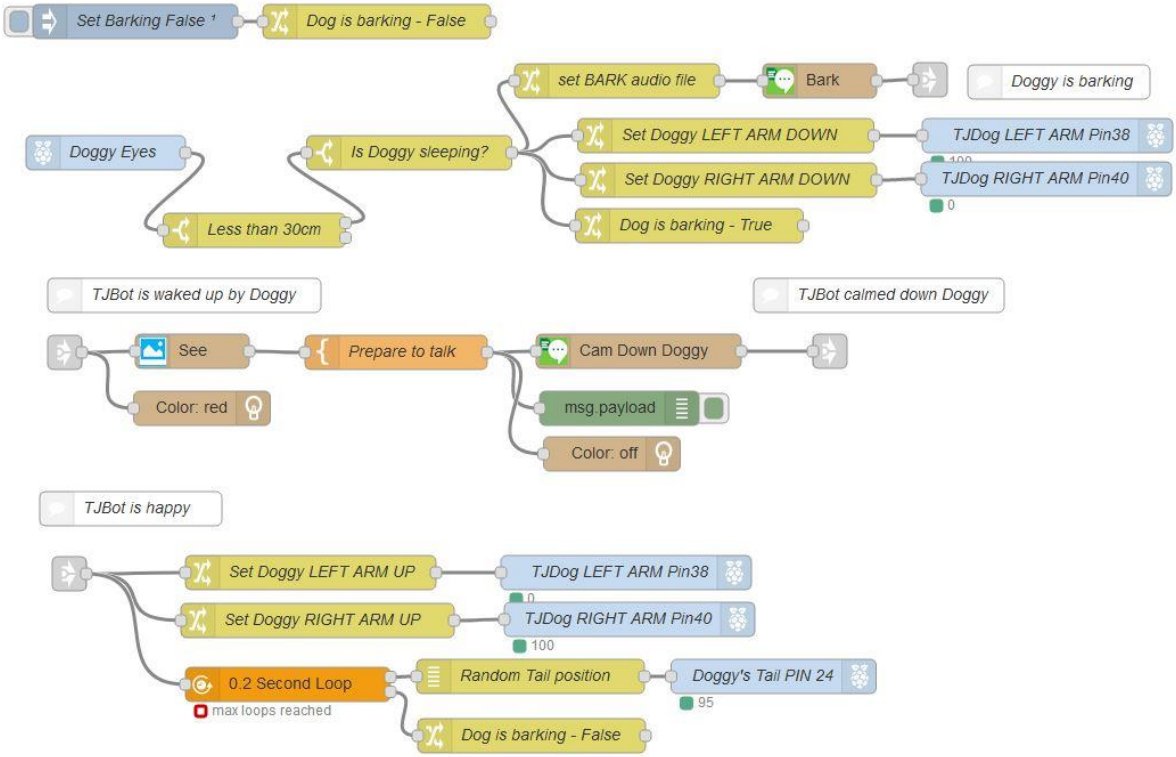
- change flow as in following picture and retry (changed nodes are highlighted).



Are TJBot and Doggy more realistic?

Steps (part 4): Doggy is calmed and happy When Doggy is calm, it wags its tail and go down on its belly.

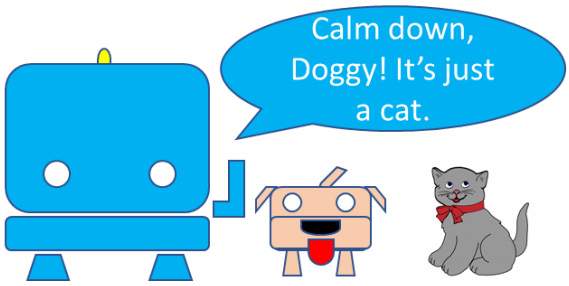
- modify flow as shown in the following figure (**hint**: copy nodes from previous flow)



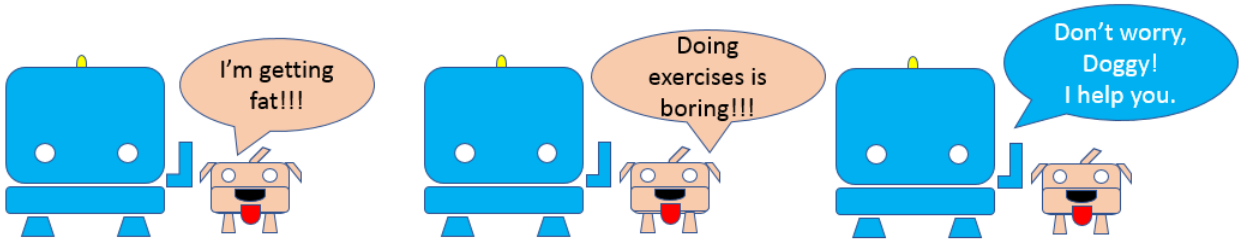
What happen when an obstacle reaches Doggy?

What happen if the object stays away from Doggy?

What happen if the object stays in front of Doggy's eyes?



Instruction: Tale 2 – Doggy likes his treats. Doggy needs to eat healthier and start doing exercises

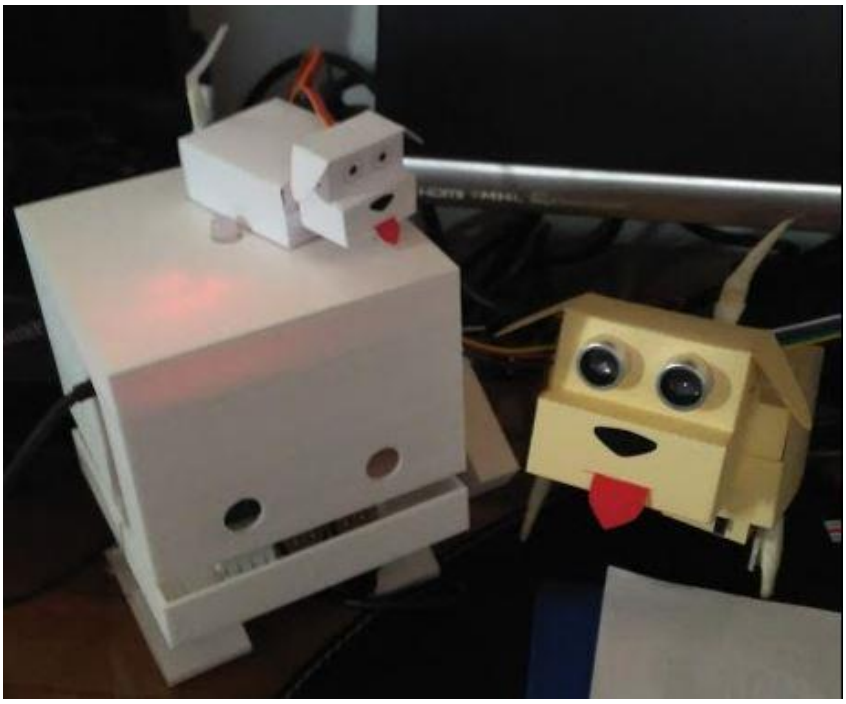


Doggy is a sweet tooth. It loves treats, cakes and ice cream. Also likes hamburgers, chips, pizza and pasta! It's getting fat, so it needs to eat healthier and start doing exercises! But it doesn't like gymnastic!!!

Challenges included

Scene Index:

- Tale2.Scene1 - Stand Up, Doggy - Doggy stands up on its front legs when anything is less that 30cm away
- Tale2.Scene2 – Doggy, trying to do push-ups - Doggy does push-ups when anything is less that 30cm away
- Tale2.Scene3 – TJBot helps Doggy doing its exercises - When anything is less that 30cm away, TJBot starts counting push-ups for Doggy

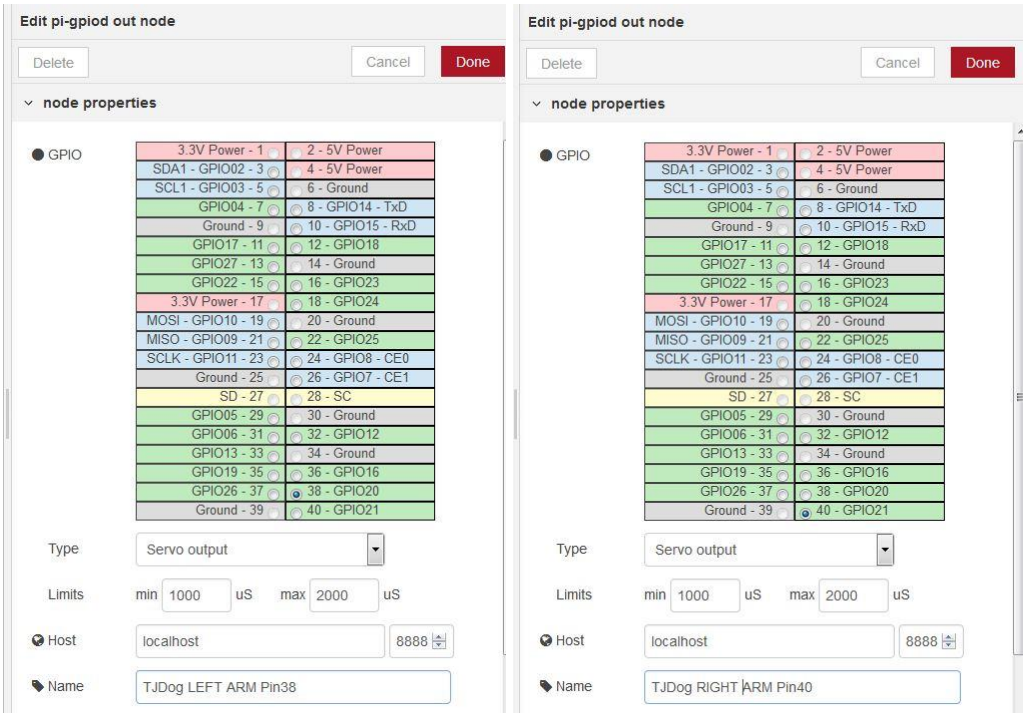


Instruction: Tale2.Scene1 –Doggy stands up on its front legs when anything is less that 30cm away

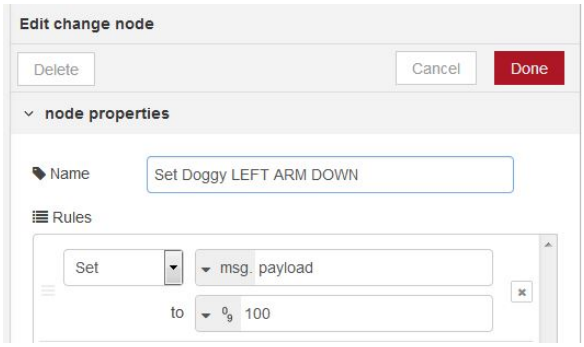
- **Requirement**: TJBot, Doggy Premium – already setup and tested
- **Skill**: Node-Red (basic level)
- **Age**: 8+
- **Knowledge acquired**: Servo motor and Sonar sensor usage, test condition, set value to variable
- **Preparation steps:**
  - Go to Node-Red Dashboard - http://«TJBot_ip_address»:1880
  - verify that TJBot is fully functional and node-red is working as expected (TJBot credentials, too).
  - Create/verify that Tale1.Scene4 is enabled and working
- **Level:** some task could be tricky but can be performed under adult's guidance

Instructions:

- go to Node-Red Dashboard http://«TJBot_ip_address»:1880. Create a new Flow
- double-click Flow label and rename it to "Tale1.Scene5". Press "Done" to close.
- disable "Tale1.Scene4" Flow. Press "Done" to close.
- copy "Doggy Eyes" and "Less than 30cm" nodes in "Tale1.Scene4" Flow.
- Click on "Tale1.Scene5" Flow label. Press CTRL+V keys to paste them.
- drag twice PI-GPIOD node to working area (you'll get two nodes)
- double-click PI-GPIOD nodes and configure them as shown in following pictures.

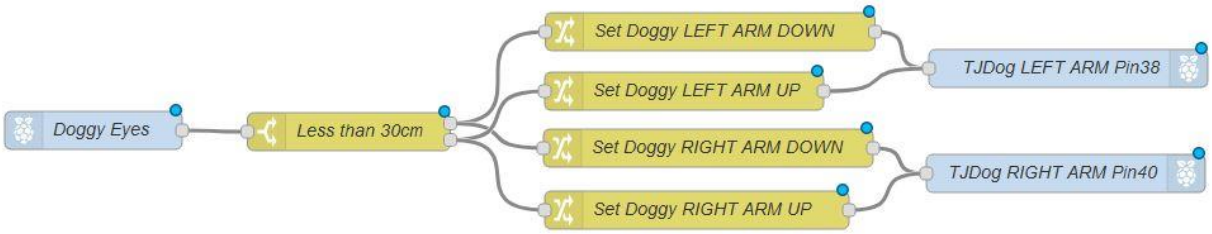


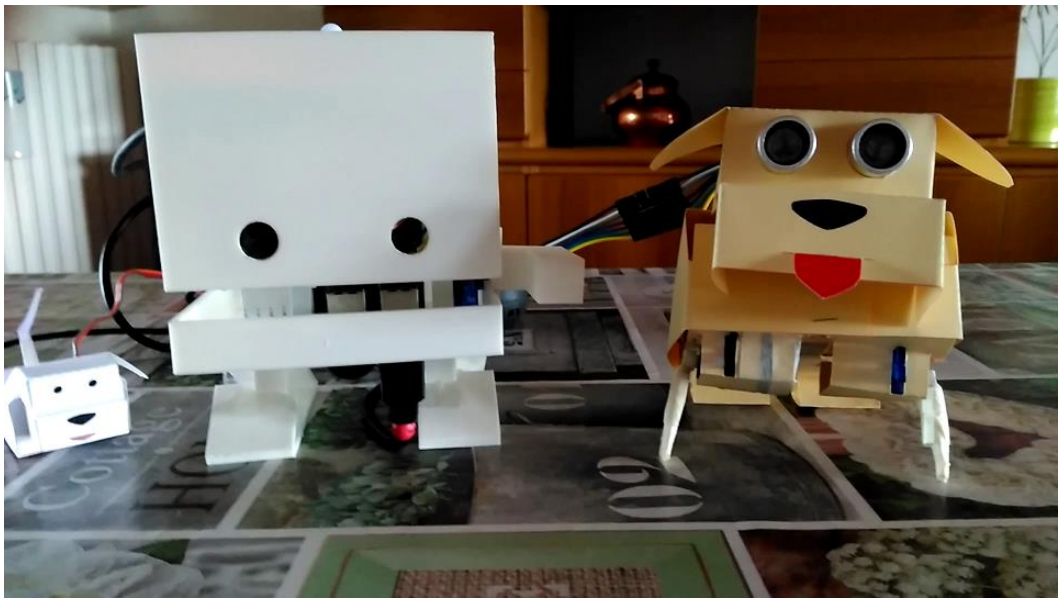
- drag four times CHANGE node to working area and configure them as shown in picture following the table provided



Node	Name	Value
Change #1	Set DOGGY LEFT ARM to DOWN	100
Change #2	Set DOGGY LEFT ARM to UP	0
Change #3	Set DOGGY RIGHT ARM to DOWN	0
Change #4	Set DOGGY RIGHT ARM to UP	100

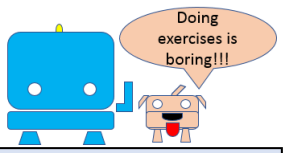
- connect and rearrange, if needed, as show in the following picture and press DEPLOY button





Anyone scared by Doggy? Don't panic! It's a cute dog.

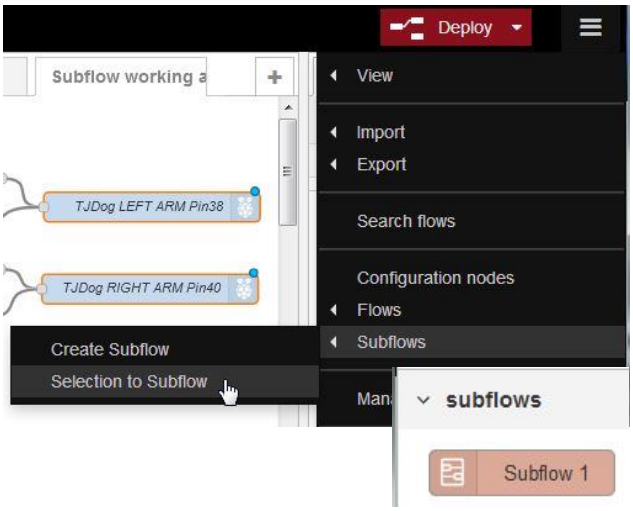
Instruction: Tale2.Scene2 –Doggy does push-ups when anything is less that 30cm away



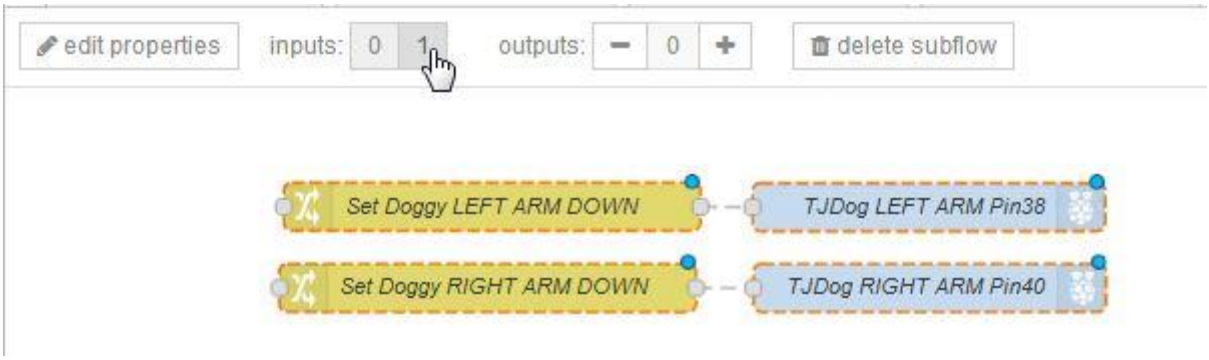
- **Requirement**: TJBot, Doggy Premium – already setup and tested
- **Skill**: Node-Red (basic level)
- **Age**: 10+
- **Knowledge acquired**: Servo motor and Sonar sensor usage, test condition, set value to variable, Node-red subflow, Problem decomposition
- **Preparation steps:**
  - point your browser to Node-Red Dashboard - http://«TJBot_ip_address»:1880
  - verify that TJBot is fully functional and node-red is working as expected (TJBot credentials, too).
  - verify if LOOPTIMER is installed. If not, install node-red-contrib-looptimer via “Manage Palette”
  - verify that Tale2.Scene1 is enabled and working on dedicated Flow
- **Level:** some task could be tricky but can be performed under adult’s guidance

Steps (part 1 - just one push-up!):

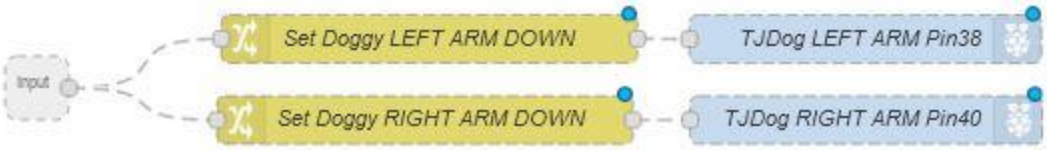
- create a new Flow. Rename it to “Subflow working area”. Press “Done” to close.
- disable “Tale2.Scene1” Flow.
- copy all nodes in “Tale2.Scene1” Flow to “Subflow working area”.
- select "Set Doggy LEFT ARM DOWN", "Set Doggy RIGHT ARM DOWN", "TJDog LEFT ARM Pin38" and "TJDog RIGHT ARM Pin40" nodes keeping CTRL key pressed. Copy them on the same flow and select them again.
- Click on Node-Red Menu, click on SUBFLOWS, and select "Selection to SubFlow". Selection will be substituted by a new node, that appears even in palette area.



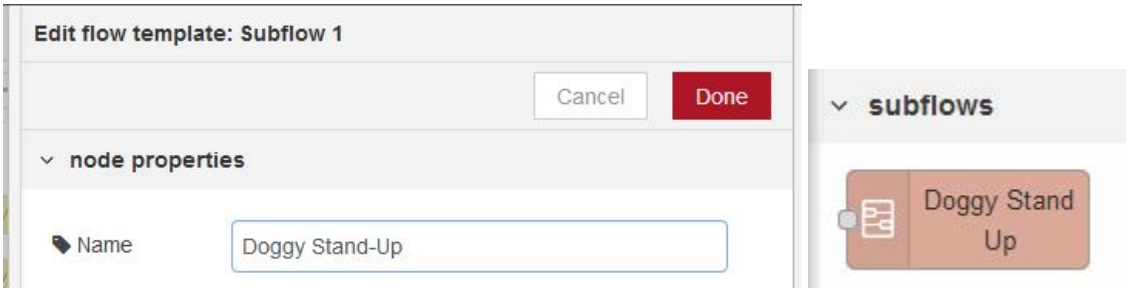
- double-click on created subflow in the Palette area. It opens in Working area and could be edited. Click on value "1" in Inputs selector.



- connect new input node to CHANGE nodes



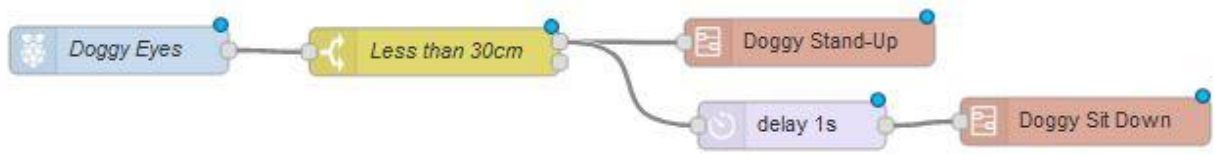
- click on "Edit Properties" and change name to "Doggy Stand-Up". Press "Done" to close and close subflow editor.



- do the same again with "Set Doggy LEFT ARM UP", "Set Doggy RIGHT ARM UP", "TJDog LEFT ARM Pin38" and "TJDog RIGHT ARM Pin40", creating a new subflow called "Doggy Sit Down". In the same way, you can create basic Doggy actions (wags_tail, bark...).

- delete “Subflow working area” flow

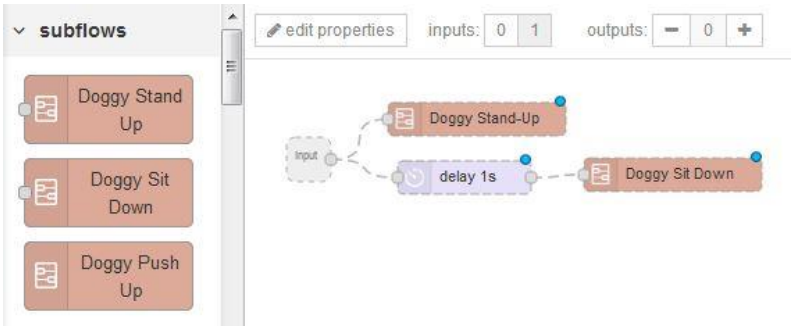
- create a new Flow
- double-click Flow label and rename it to “Tale2.Scene2”. Press “Done” to close.
- double-click “Tale2.Scene1” Flow label.
- select “Doggy Eyes” and “Less than 30cm” nodes in “Tale2.Scene1” Flow. Press CTRL+C keys to copy them. Click on “Tale2.Scene2” Flow label. Press CTRL+V keys to paste them.
- OK, now start little, just one push-up! From palette, drag "Doggy Stand Up" and "Doggy Sit Down" subflows to Working area. Drag DELAY node, too. Double-click it and set a 1 second delay. Connect nodes as show in figure and DEPLOY.



What happens when an obstacle reaches Doggy? Doggy must do more exercise :)

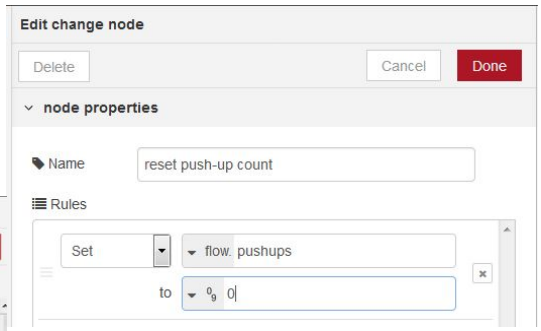
Steps (part 2 - 10 push-ups):

- create a subflow called "Doggy Push-Up" using "Doggy Stand Up", "Doggy Sit Down" and "delay 1s" nodes.

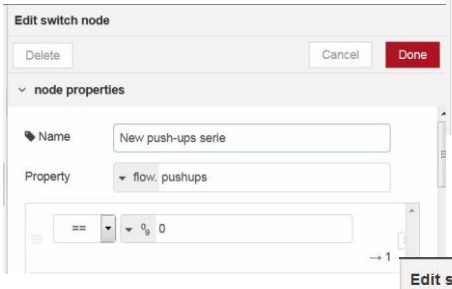


- modify “Tale2.Scene2” flow using new created subflow.

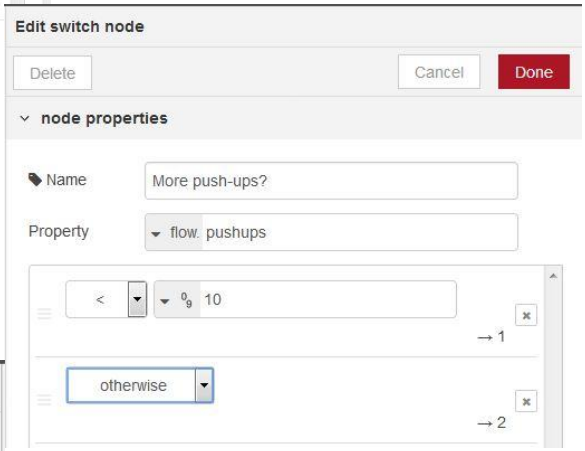
- delete connection between SWITCH and "Doggy Push-Up"
- drag once CHANGE node from Palette to Working area. Modify it as shown in figure



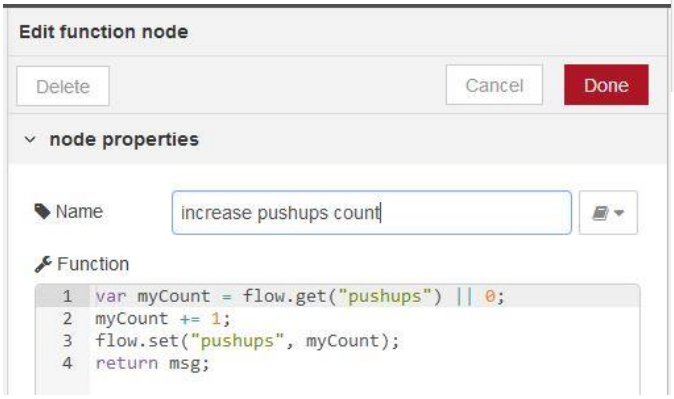
- drag once SWITCH node from Palette to Working area. Modify it as shown in figure



- drag another SWITCH node from Palette to Working area. Modify it as shown in figure

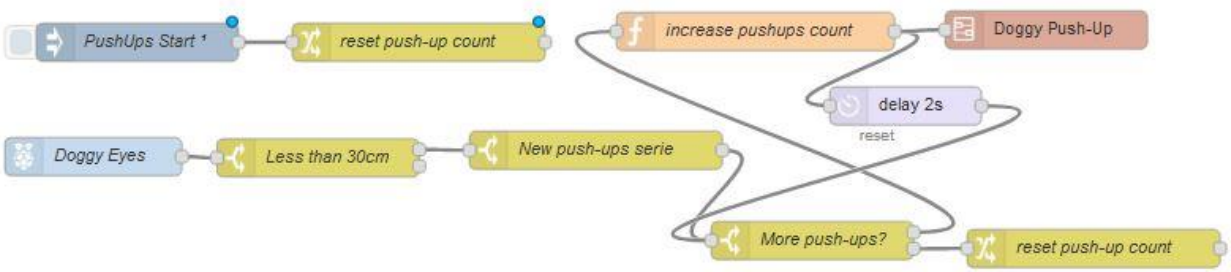


- drag once FUNCTION node from Palette to Working area. Modify it as shown in figure, inserting the following code that set/increase pushups count:

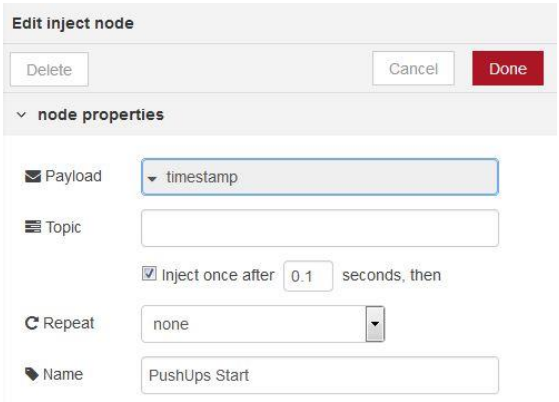



```
`var myCount = flow.get("pushups")
|| 0;
myCount += 1;
flow.set("pushups", myCount);
return msg;`
```

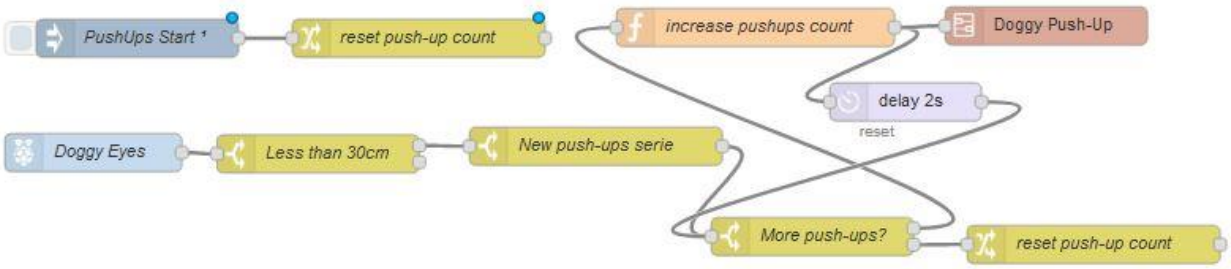
- drag once DELAY node from Palette to Working area. Set delay to 2 seconds.
- Connect nodes as show in figure. Pay attention to create the loop!



- we need to instantiate count variable (teacher/IBMers, introduce this concept to classroom).
- drag once INJECT node to Working area.
- configure INJECT node as shown



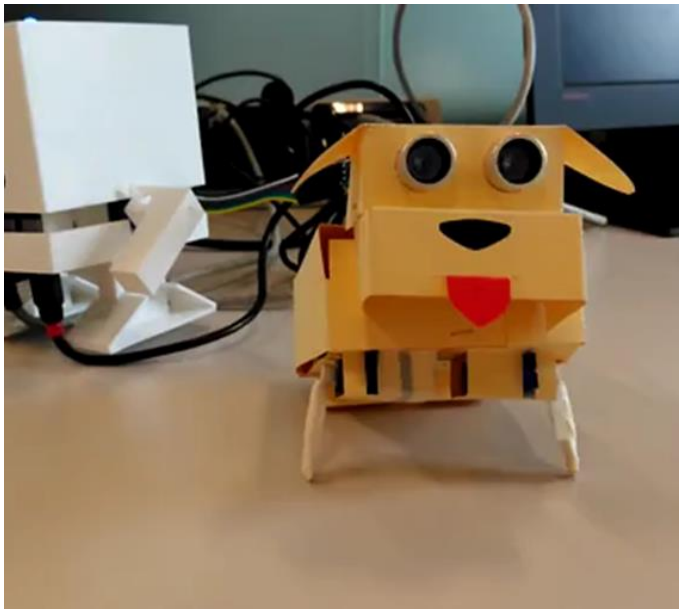
- copy "reset push-up count" node and connect as show in figure.
- Press DEPLOY button



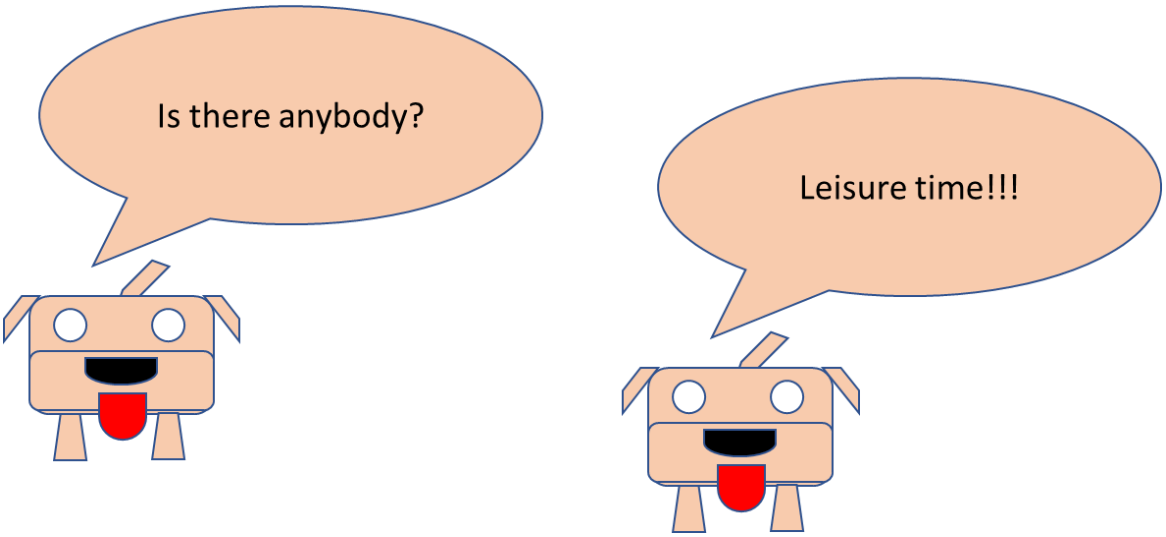
What happen when an obstacle reaches Doggy?

What happen if the object stays away from Doggy?

What happen if the object stays in front of Doggy's eyes?



Challenge T2C1: Doggy doesn't like to do exercises. When it starts a series, if nothing is near to it, it moves its tail instead of doing push-ups.



In the next page

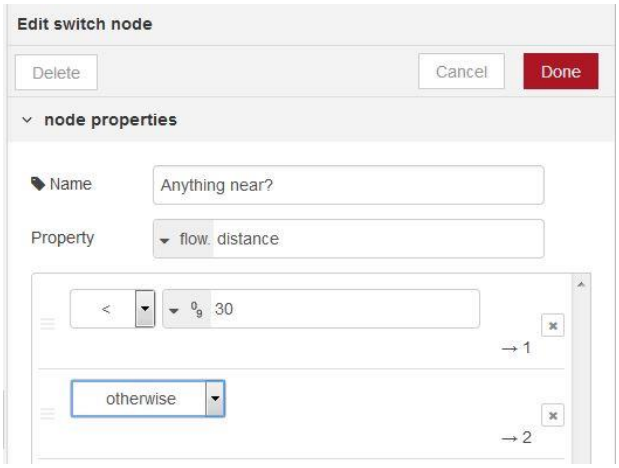
you can find

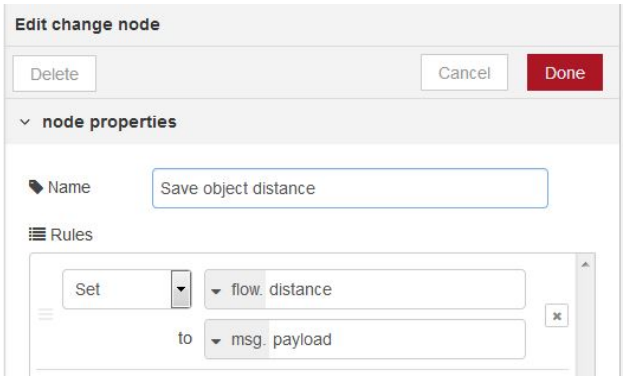
one of many solutions

Challenge T2C1: Doggy doesn't like to do exercises. When it starts a series, if nothing is near to it, it moves its tail instead of doing push-ups.

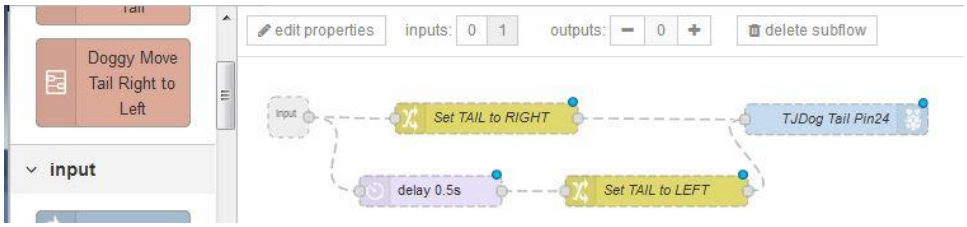
Macro Steps:

- save distance value
- test if anything is near

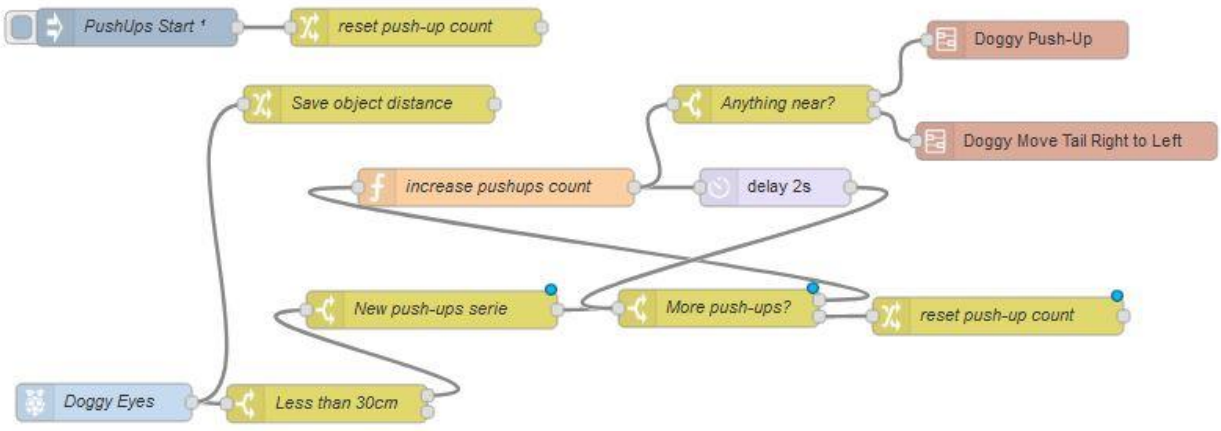




- create a new Subflow called "Doggy Move Tail Right to Left" with DELAY, two change nodes and "TJdog Tail Pin24" node from Doggy setup. Right tail position is at "0". Left is on "100" value.

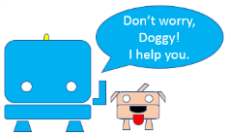


- connect nodes as in following picture and deploy it.



What happen when you move an object in front of Doggy?

Instruction: Tale2.Scene3 – TJBot helps Doggy doing exercises by counting push-ups

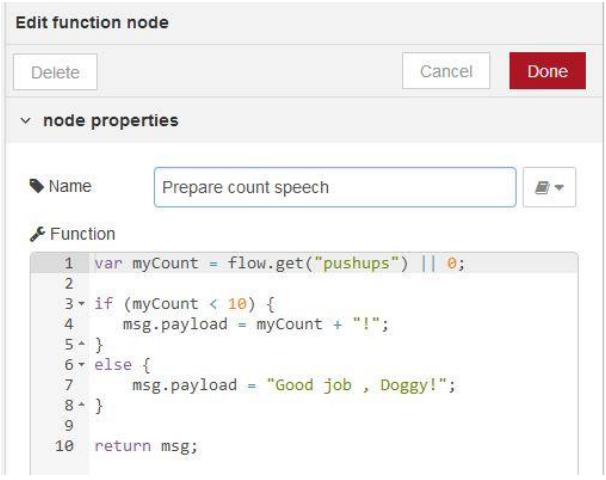


- **Requirement**: TJBot, Doggy Premium – already setup and tested
- **Skill**: Node-Red (basic level)
- **Age**: 10+
- **Knowledge acquired**: Text to Speech, TJBot capabilities (wave, speak)
- **Preparation steps:**
  - point your browser to Node-Red Dashboard - http://«TJBot_ip_address»:1880
  - verify that TJBot is fully functional and node-red is working as expected (TJBot credentials, too).
  - verify that Tale2.Scene2 is enabled and working on dedicated Flow
- **Level:** some task could be tricky but can be performed under adult's guidance

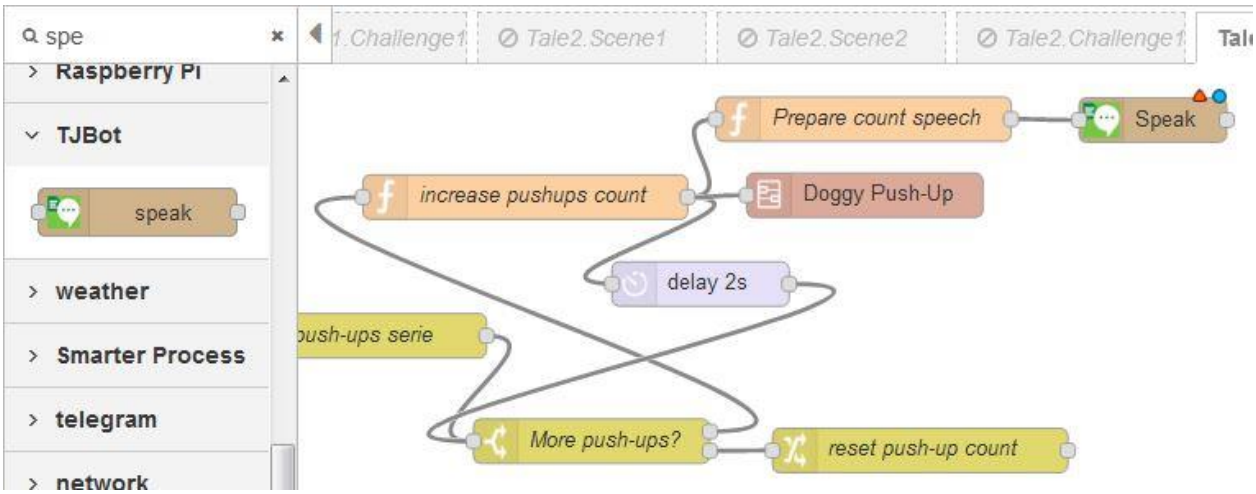
Steps (part 1 - TJBot speaks):

- point your browser to Node-Red Dashboard - http://«TJBot_ip_address»:1880 and create a new Flow
- double-click Flow label and rename it to "Tale2.Scene3". Press "Done" to close.
- double-click "Tale2.Scene2" Flow label and set status to "Disabled". Press "Done" to close.
- select all nodes in "Tale2.Scene2" Flow. Press CTRL+C keys to copy them. Click on "Tale2.Scene3" Flow label. Press CTRL+V keys to paste them.
- drag FUNCTION node to Working area and modify it to create a message that include the current number of push-ups performed by Doggy. Code is provided for cut&paste ☺
-

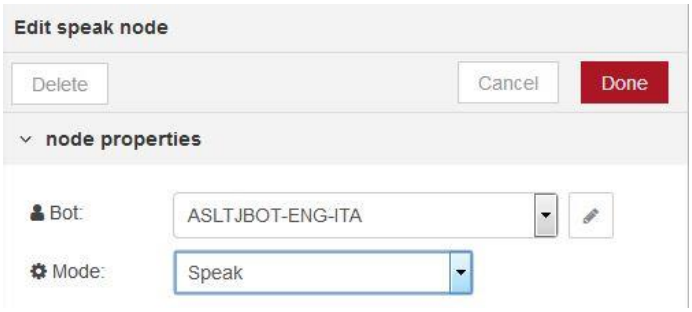
```
var myCount = flow.get("pushups") || 0;
if (myCount < 10) {
   msg.payload = myCount + "!";
}
else {
    msg.payload = "Good job , Doggy!";
}
return msg;
```



- drag TJBot's SPEAK node to Working area connecting it



- configure TJBot's SPEAK node according with your TJBot configuration (TTS credential, HW setup...) and press DEPLOY



(Note for Teacher/IBMers: this is a good time to explain local vs central processing)

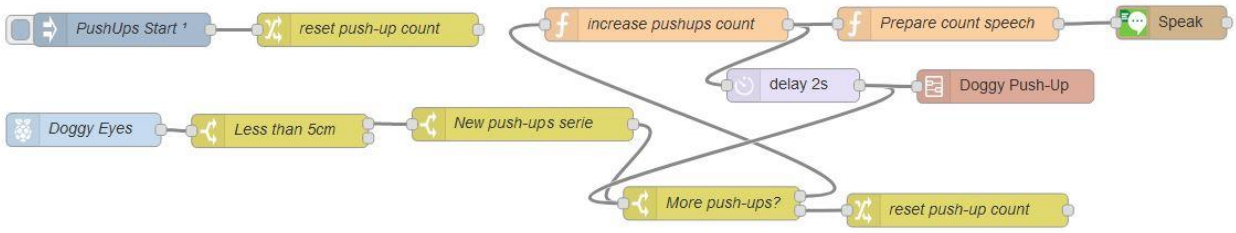
What happen when an obstacle reaches Doggy?

Is TJBot in synch with Doggy exercises?

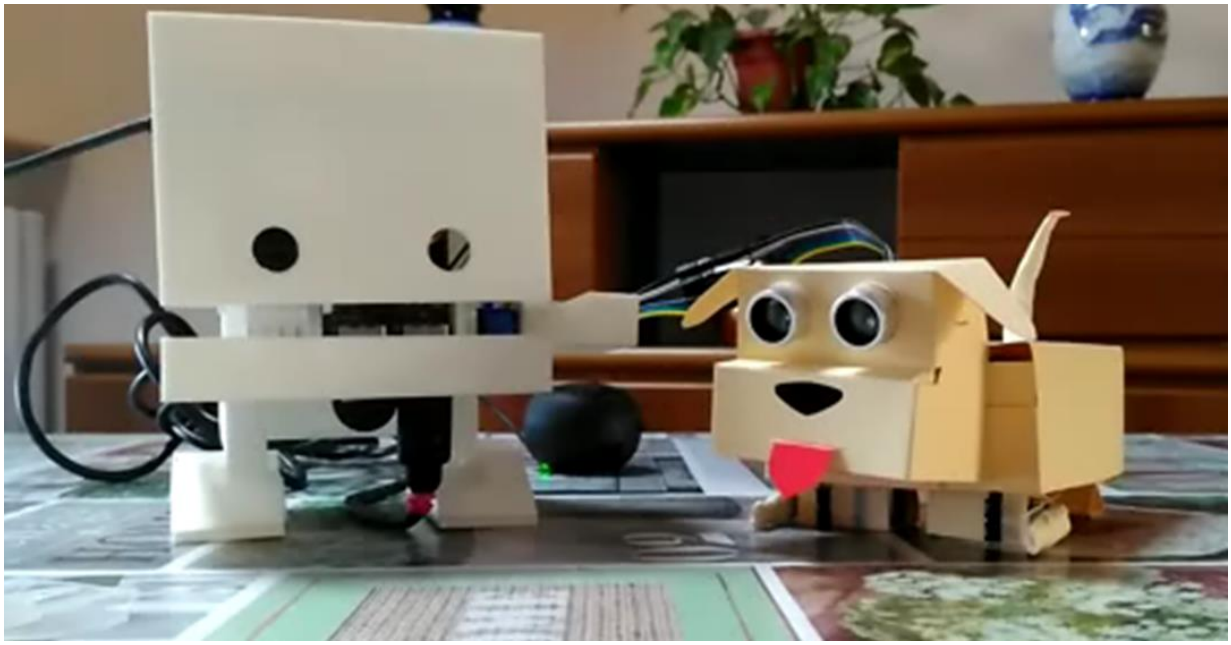
How can we get TJBot and Doggy more synched?

(ONE OF MANY solutions on the next page)

Solution: connect Doggy subflow to a different node)



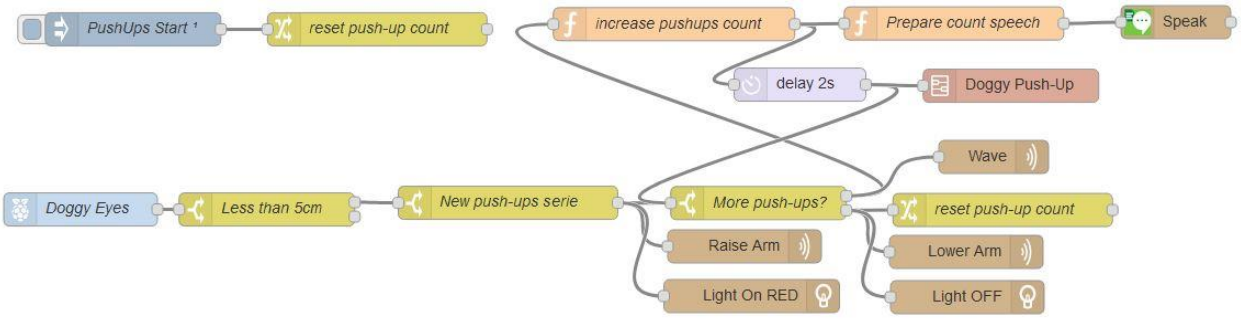
Challenge time: Now, leverage on TJBot capabilities. TJBot thinks that could be useful to move his/her arm while Doggy do push-ups and to light up led to advice that a new series is started.



Ready... Steady... Go!!!

(a candidate solution on next slide)

Challenge T2C2: TJBot thinks that could be useful to move his/her arm while Doggy do push-ups and to light up led to advice that a new series is started.



We Need You!

Submit your scenes to expand Doggy's behaviour :)

End notes and Credits

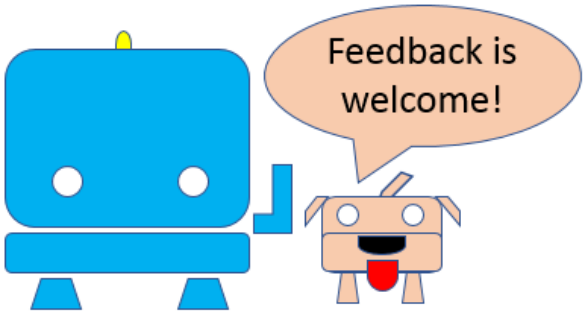
TJBot is a brilliant way to introduce and play with IBM Watson solution and robotics.

Node-Red makes programming a very easy task, from children to grandparents.

Doggy demonstrates how to expand TJBot using common and low cost IOT devices.

Mixing all together is an excellent way to have fun and effective activities.

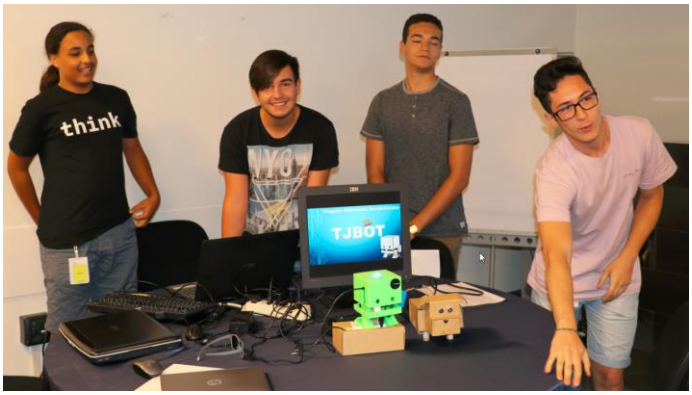
Ferruccio Manclossi,
IBM Italy, September 2018
https://github.com/fmanclossi/TJBot-playbook



Doggy is an original idea from a team of IBM Italy Volunteers.

The first draft was developed during a stage with high school students in IBM Segrate, July 2018.

A special ***GRAZIE*** to Fabrizio Repossi, Hossam Lemtaffah, Matteo Vergottini, Riccardo Manfredonia, Thierno Filippo D'Amelio and Usama Muhammad



The goal of education is not to increase the amount of knowledge but to create the possibilities for a child to invent and discover, to create men who are capable of doing new things. Jean Piaget

About this volunteer activity

IBM is sharing activities our employees have performed successfully worldwide, and we invite you to use them in your own community. Many are drawn from our community service initiative, IBM Volunteers, which supports our colleagues' volunteer efforts. Find more resources at ibm.com/ibm/responsibility/initiatives/volunteers.shtml.