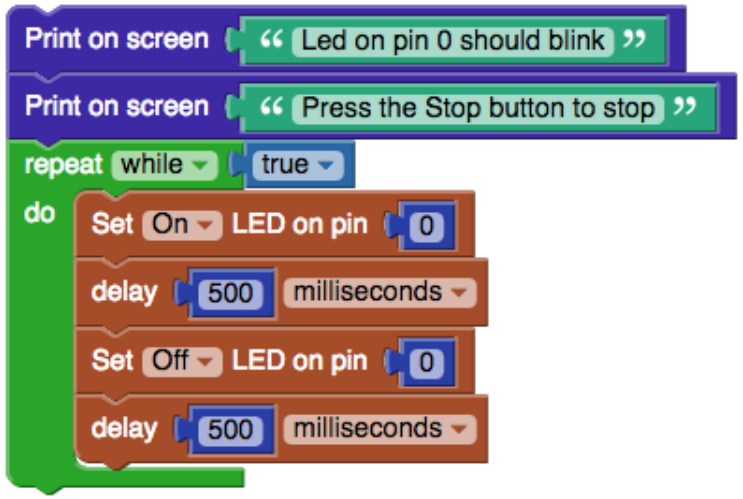
# Programming using Wyliodrin

SLIDES TAKEN FROM IOT SUMMER CAMP 2015, INNOVATION LABS

# What Language Should I use?

# Visual Programming



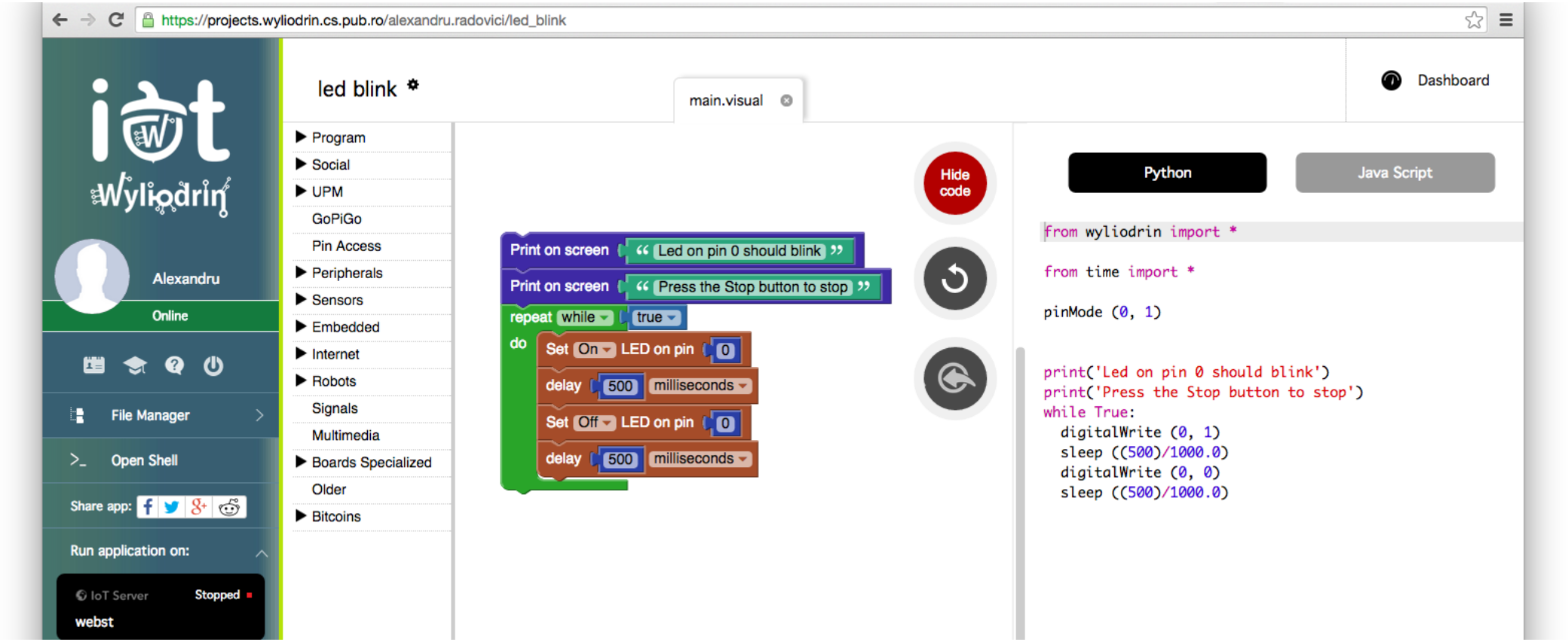
# Prototyping

Drag and drop blocks

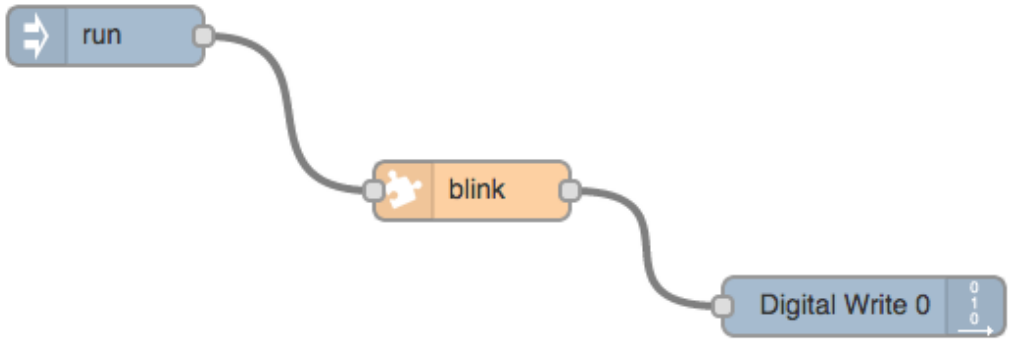
Writes code in Python or Javascript

Good for documentation

# Python


https://projects.wyliodrin.cs.pub.ro/alexandru.radovici/led_blink
Wyliodrin
Alexandru
Online
File Manager
Open Shell
Share app:
Run application on:
IoT Server
Stopped
webst
led blink
main.visual
Dashboard
Program
Social
UPM
GoPiGo
Pin Access
Peripherals
Sensors
Embedded
Internet
Robots
Signals
Multimedia
Boards Specialized
Older
Bitcoins
Hide code
Python
Java Script
Print on screen "Led on pin 0 should blink"
Print on screen "Press the Stop button to stop"
repeat while true do
Set On LED on pin 0
delay 500 milliseconds
Set Off LED on pin 0
delay 500 milliseconds
from wyliodrin import *
from time import *
pinMode (0, 1)
print('Led on pin 0 should blink')
print('Press the Stop button to stop')
while True:
digitalWrite (0, 1)
sleep ((500)/1000.0)
digitalWrite (0, 0)
sleep ((500)/1000.0)

# Streams Programming



# Data Driven Programming

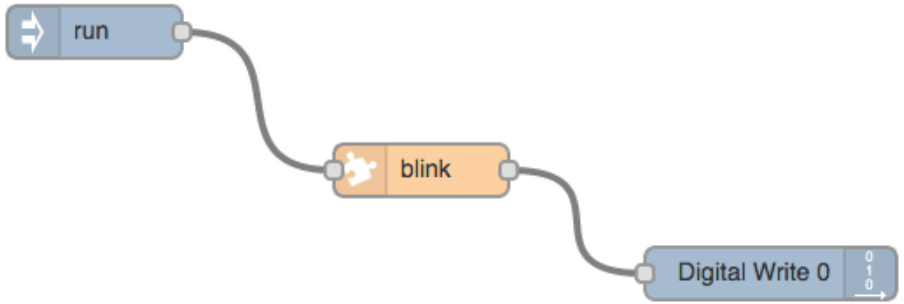
Event based

Data determines actions

The program is a graph

Elements

- Nodes
- Data routes



# Messages

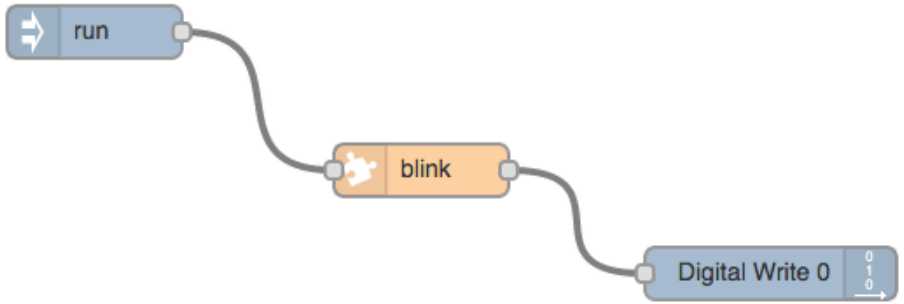
Nodes send messages to each other

- An input
- Multiple outputs

When a message arrives in the input

- Node activates
- Might send out some messages

Example



# Messages

Javascript objects

Nodes expect to receive payload

{

*topic*:

**payload**:

**…**

}

# Nodes

Run

Digital and analog pins access

Trigger

Delay

Value

Buffer

Switch

Change

Range

Function

Visual

Subflows

Signals

Messages

Print

Web request

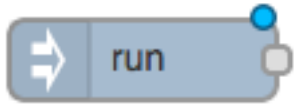
Web server

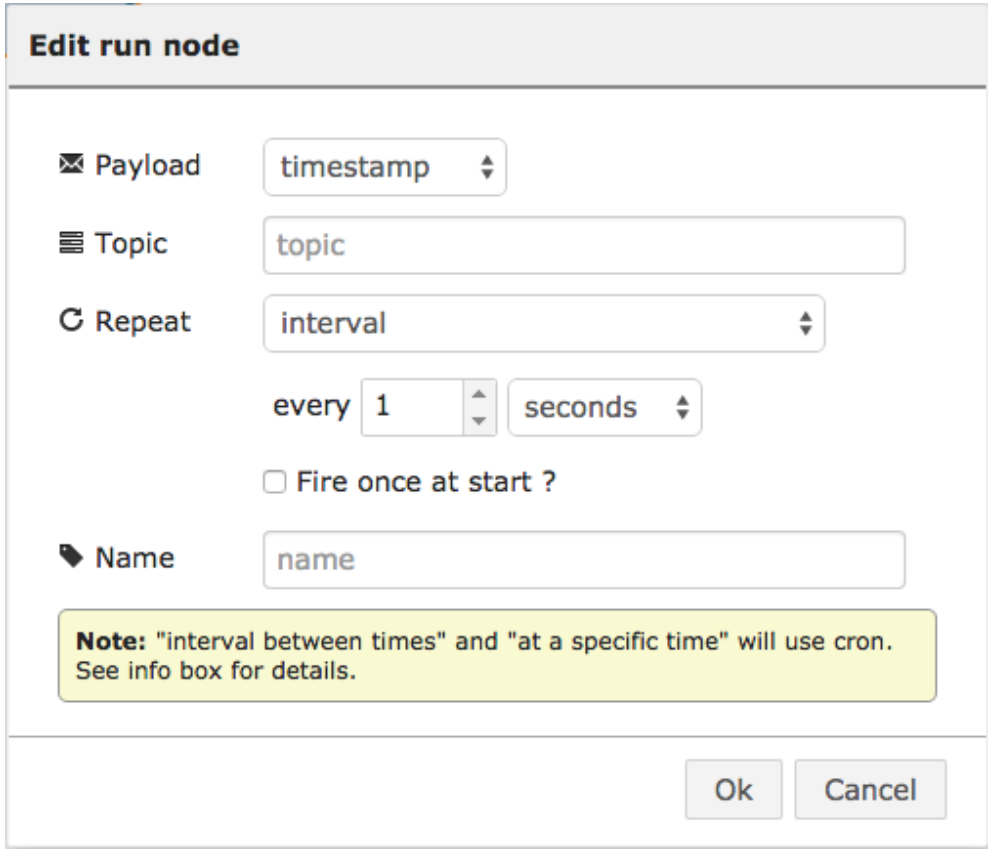
# Run

Sends a message at a certain interval

Payload:

- Timestamp
- Blank
- String
- # number of message



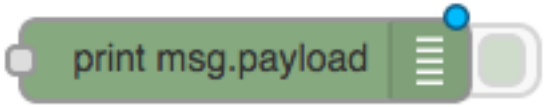



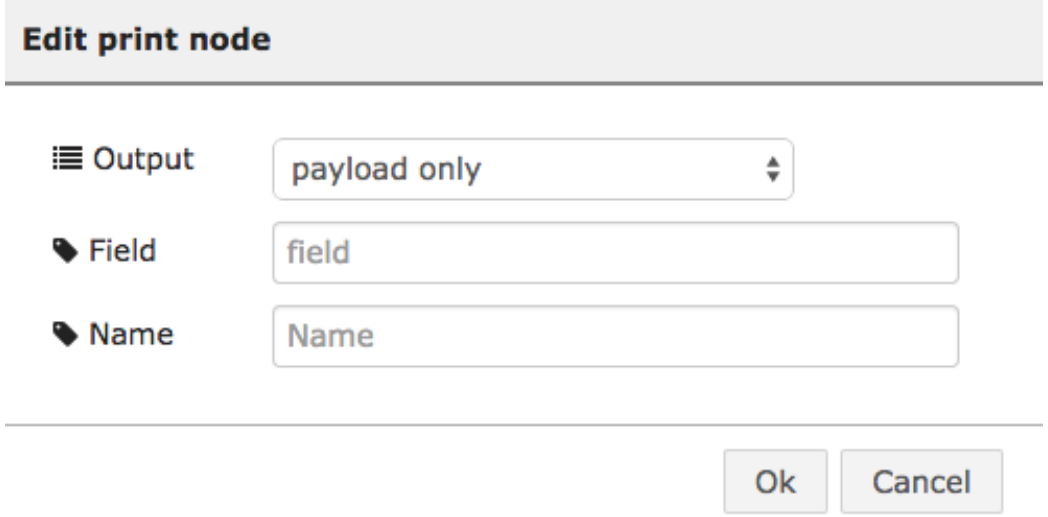
# Print

Prints to the screen

- Only payload
- Whole messages
- Some field

May be stopped with the button

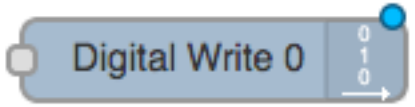


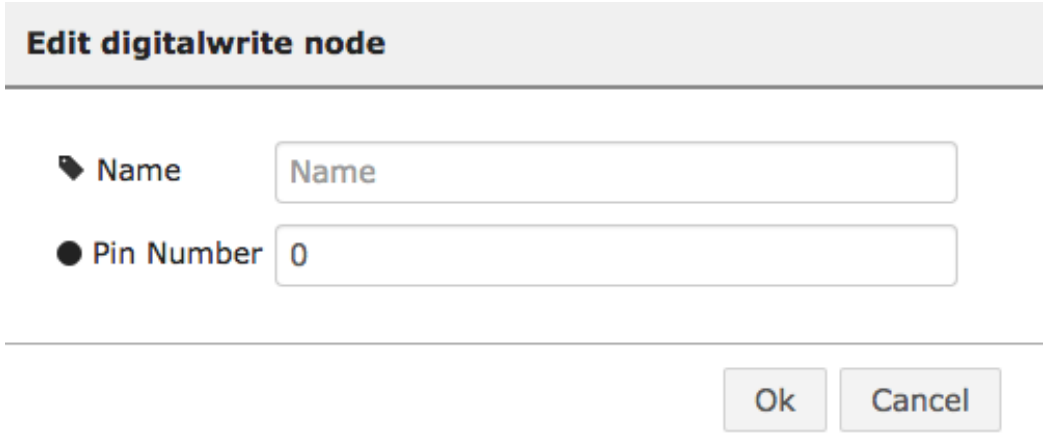



# Digital and analog pins access

Digital write

- Writes HIGH(1) or LOW (0) when it receives a message
- The value written is the payload

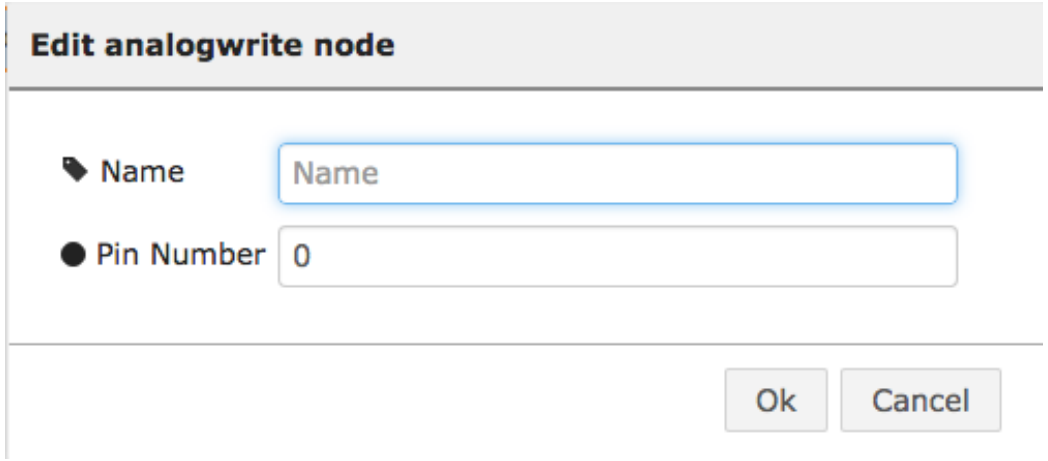


# Digital and analog pins access

Analog write

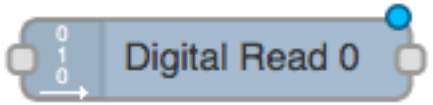
- Writes PWM when it receives a message
- The value written is the payload

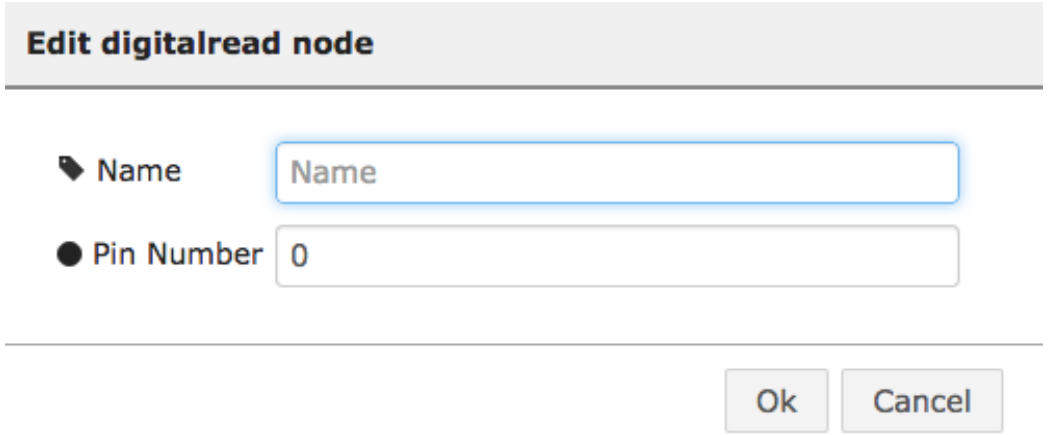



# Digital and analog pins access

Digital read

- Reads HIGH(1) or LOW (0) when it receives a message
- The value read is sent in the payload

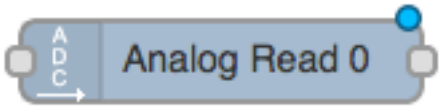


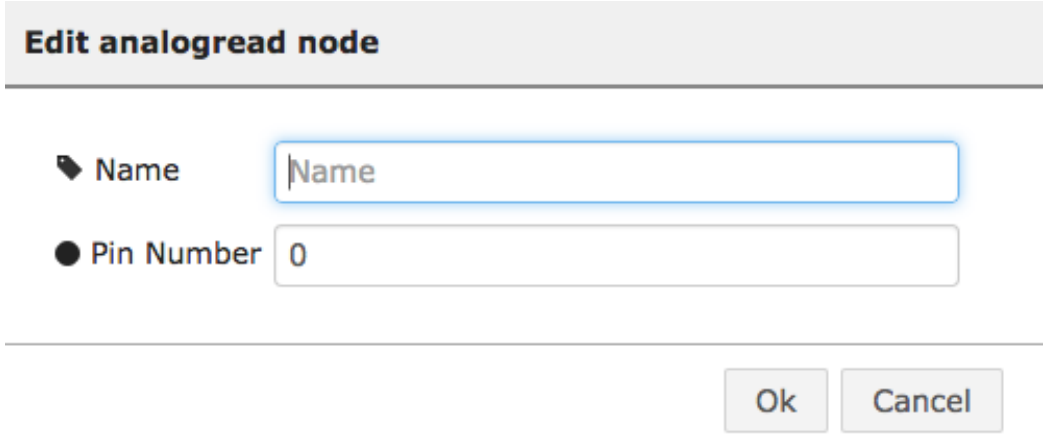



# Digital and analog pins access

Analog read

- Reads an analog value when it receives a message
- The value read is sent in  the payload

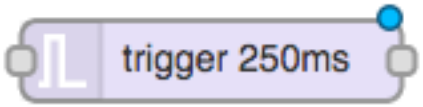


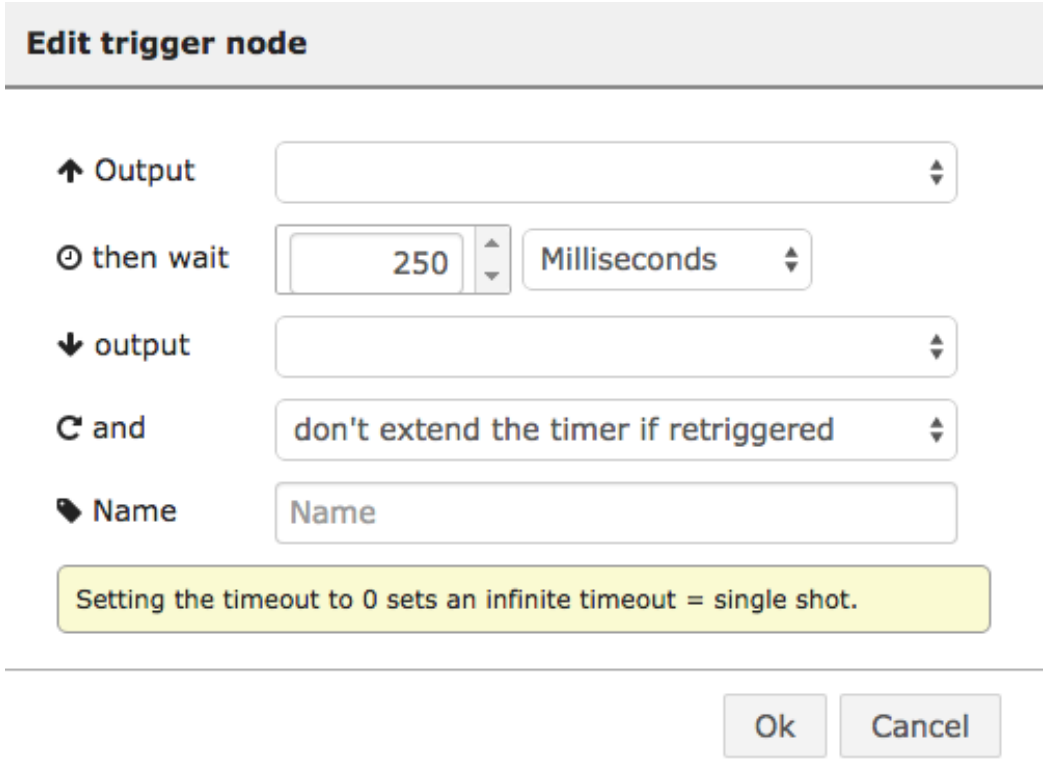



# Trigger

When it receives a message

- Sends a message with Output as payload
- Waits for some time
- Sends the second Output as a payload



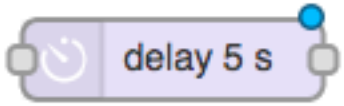
# Delay

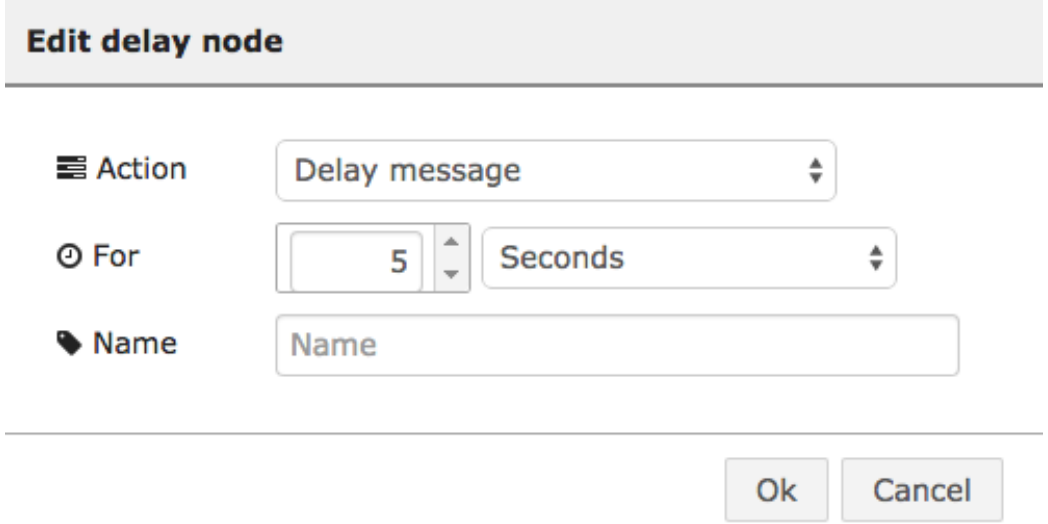
Delay

- Sends the received message with a delay

Limit

- Limits the number of messages that it sends
  - Drops or storas messages

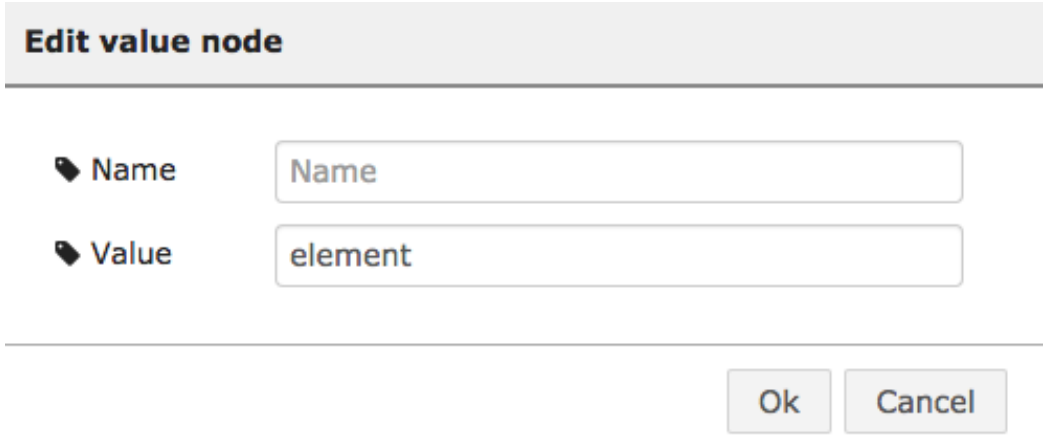


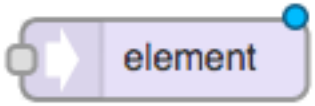



# Value

Value

- Stores the payload
- context.global._value_name_

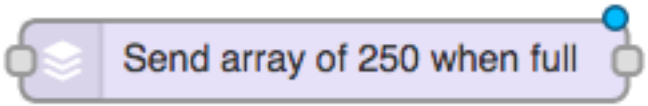


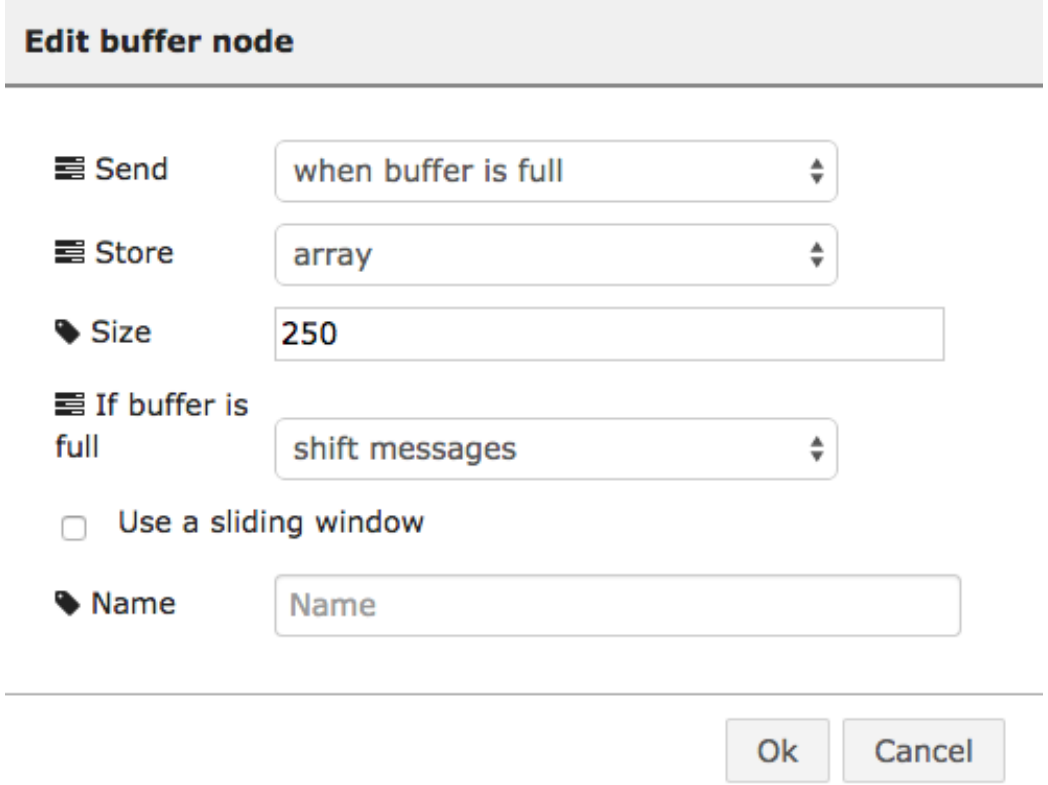



# Buffer

When it receives a messages

- Adds the payload and other properties to an array
- It sends the array when
  - The buffer is full
  - Receives an event

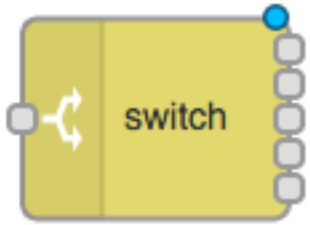


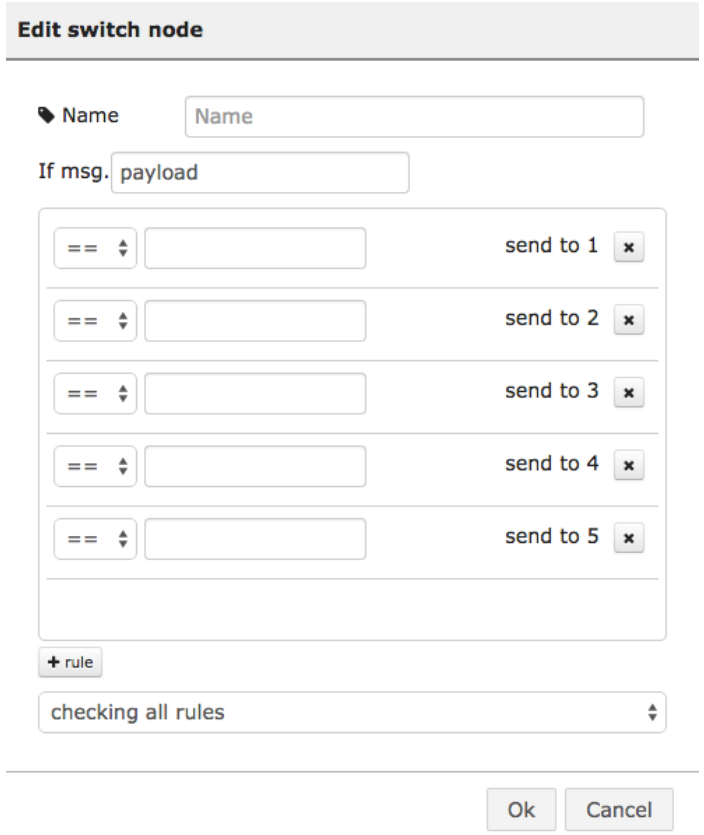



# Switch

When it receives a messages

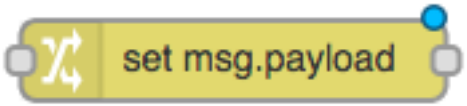
- Checks some rules
- Send the message using the rule's output
- Values may be used
  - {{value}}

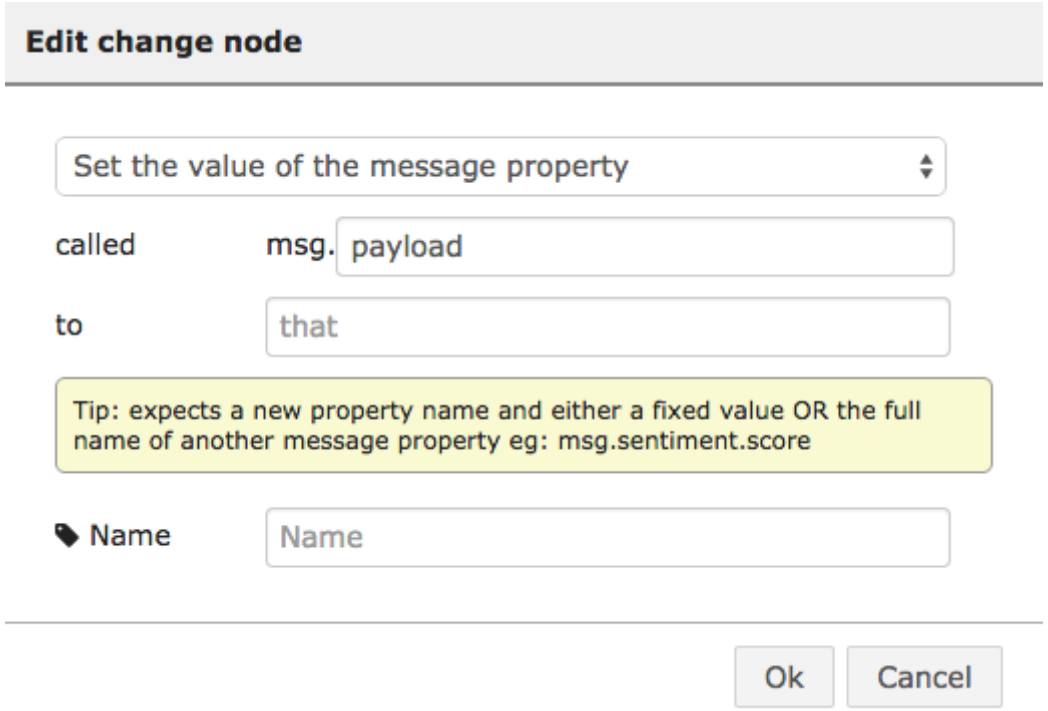


# Change

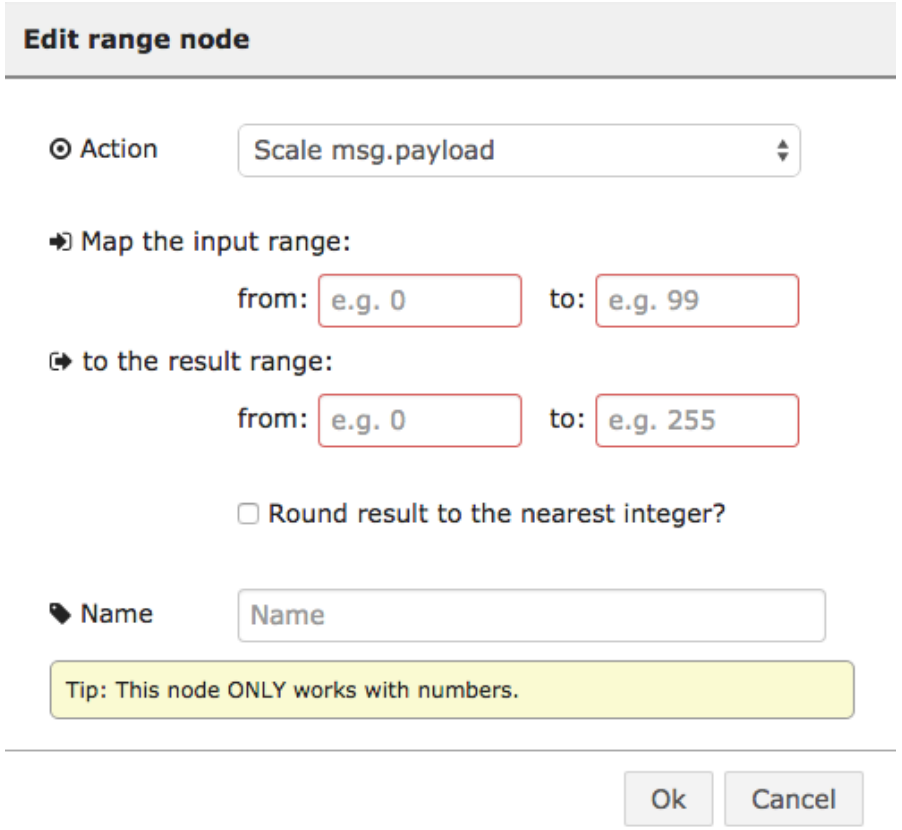
Modifies the properties of a message



# Range

Scales the payload

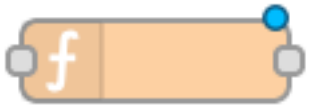
# Function

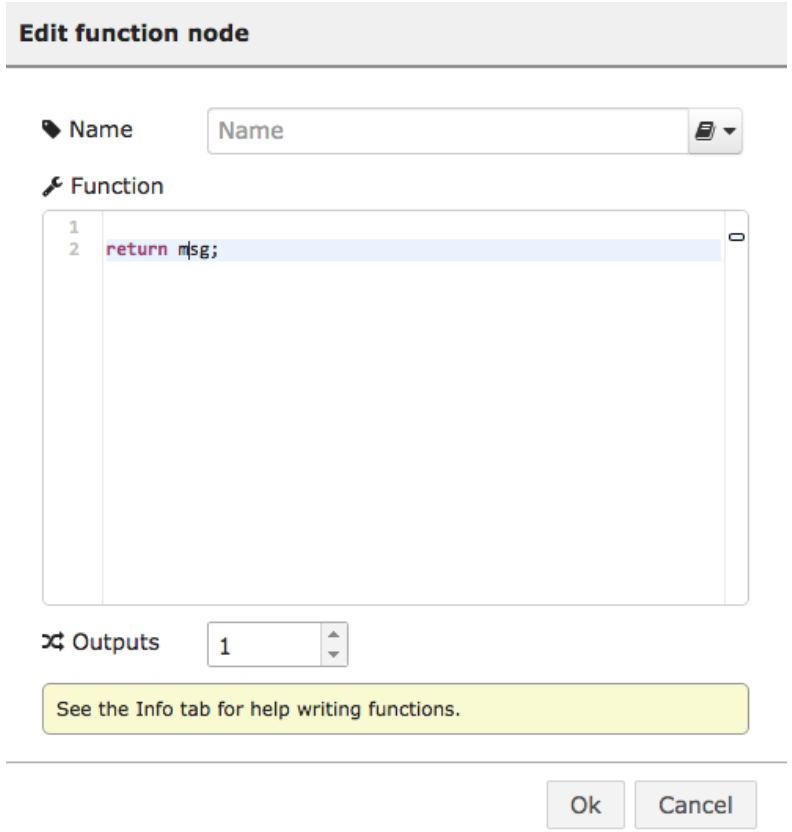
Javascript function

- msg the message received
- context.global values stored with the value node
- return sends the message

May have multiple outputs

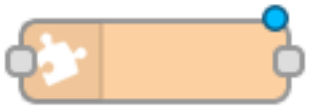
- return [o1, o2, o3]

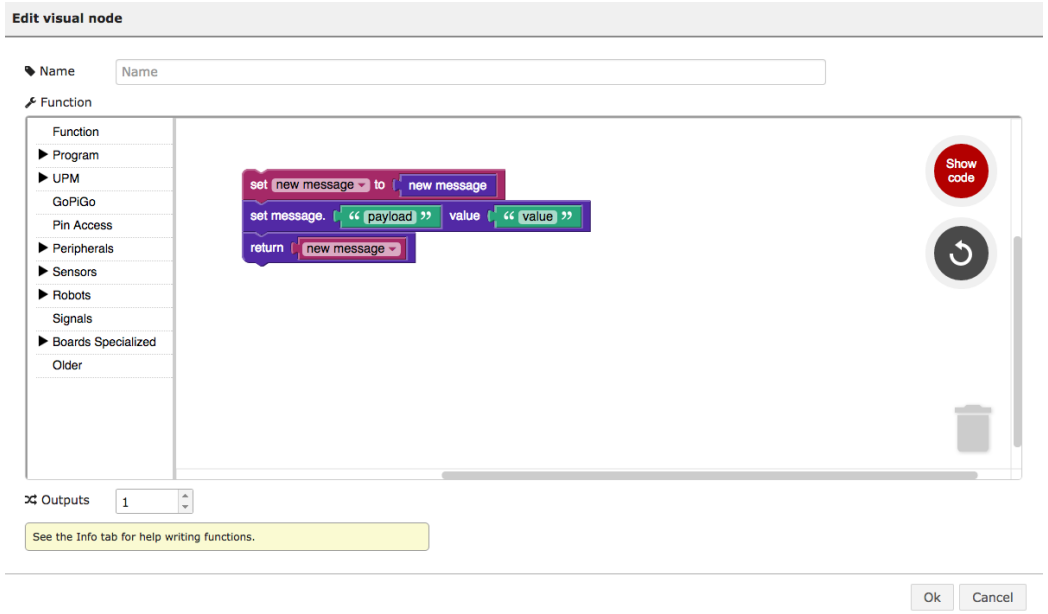




# Visual

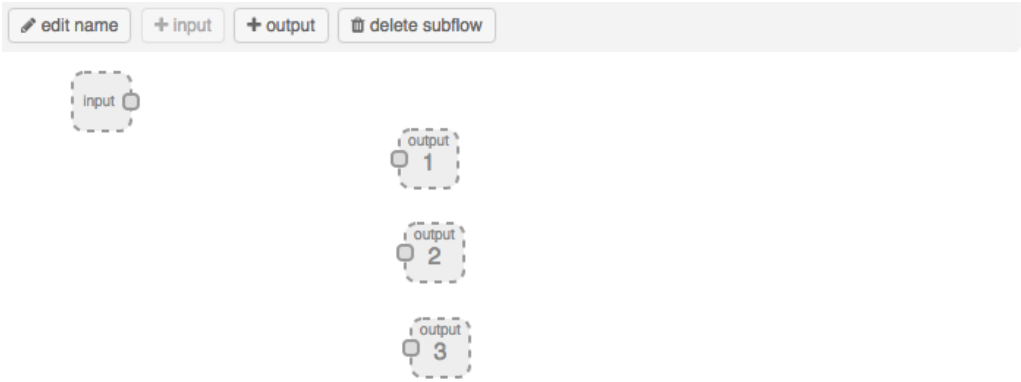
## Visual Programming Function

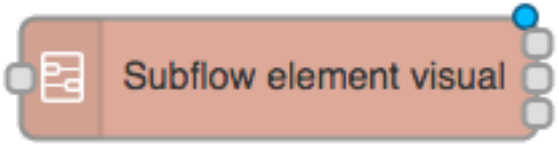




# Subflows

Subflows

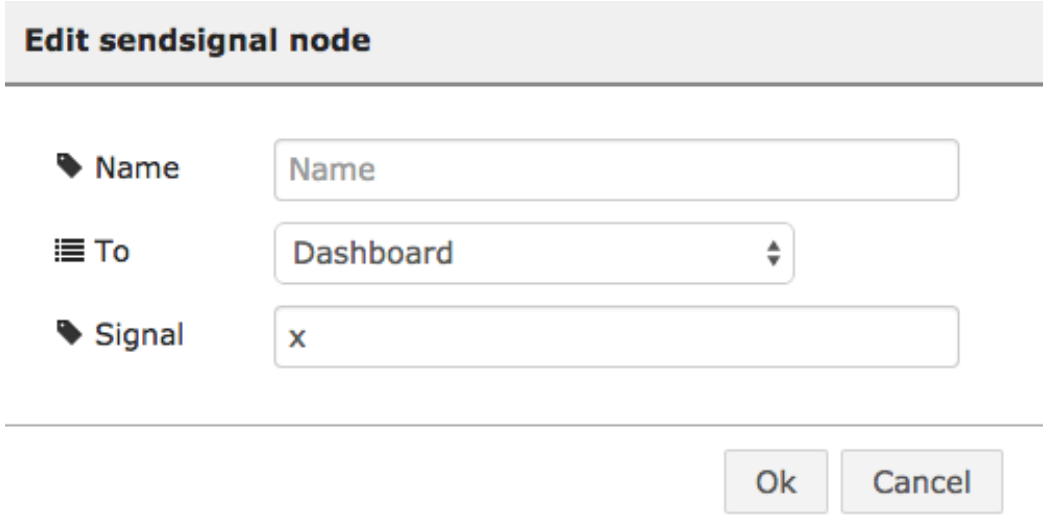


# Send signal

Sends a signal

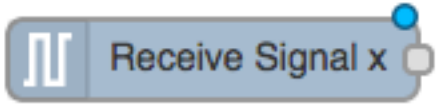
- To the dashboard
- To another board

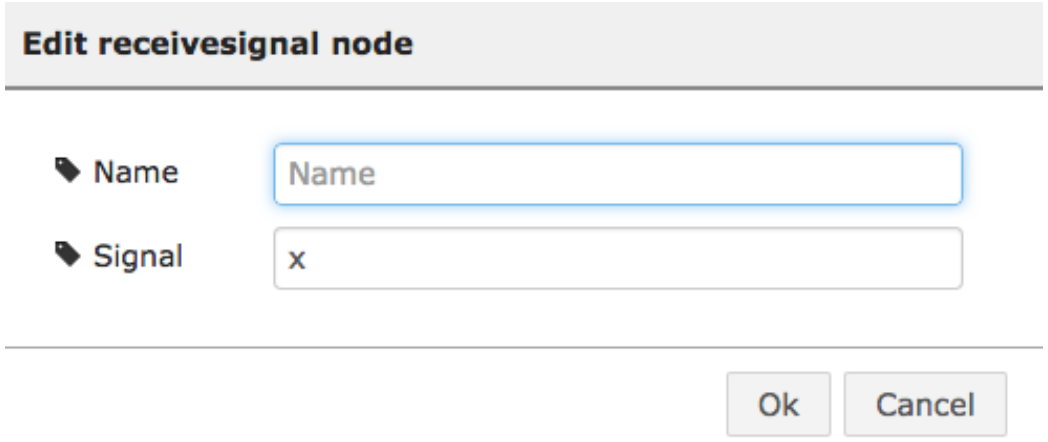



# Receive signal

Receives a signal

- Sends a message with the signal received
- Payload is the value
- Topic is the name
- Sender
  - User id if it comes from the dashboard
  - Board id if it comes from a board

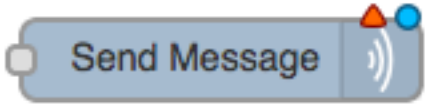


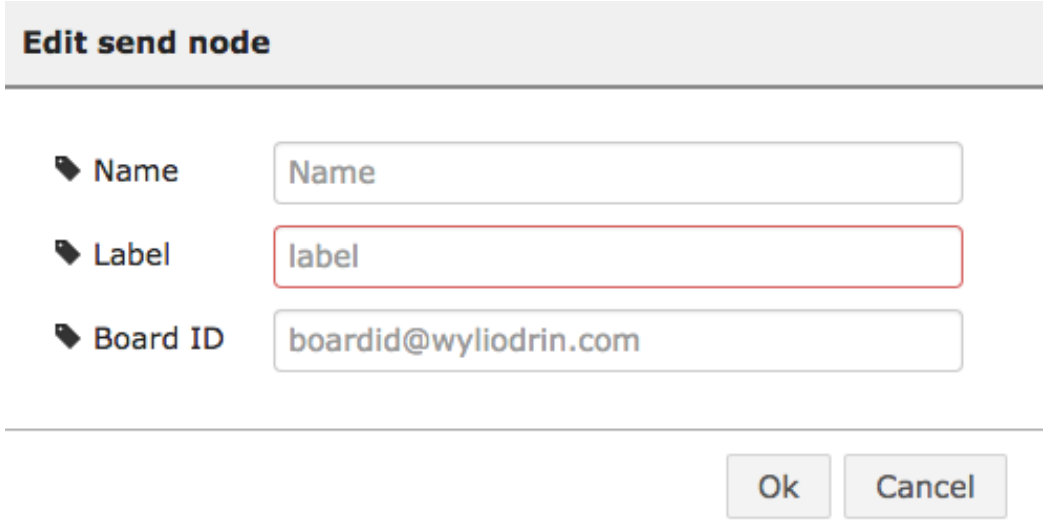



# Send message

Sends a message to another board

- The message is the payload
- Uses label instead of port



# Receive message

Receives a message from another board

- The message is the payload
- Sender is the board's id
- Uses label instead of port