

# Project 1. Keep Your Distance

Giacomo Sguotti

10667547

June 2021

## 1 Introduction

The purpose of this project was to simulate a set of motes that periodically send a message (500ms). If a mote receives 10 consecutive messages from the same mote, it would trigger an alarm through a push notification to the smartphone.

All the project files can be found at <https://github.com/LoSgu8/KeepYourDistance>.

## 2 TinyOS

The maximum number of motes is defined in the `MOTE_NUMBER` `#define` directive.

Since it cannot be assumed that the timers of the nodes are synchronized, I thought about adding an auxiliary variable to the periodically sent message.

In particular, this variable acts as an incremental sequence number which uniquely identifies a message sent from a certain mote. In this way, every time a mote receives a message it is able to check if it is subsequent to the one received previously and therefore increase a local counter which, upon reaching 10, will trigger the alarm.

Whereas the most reasonable application of this project does not last for a few minutes but for an indefinite time and being the sequence numbers limited, I decided to use a 16-bit unsigned integer variable instead of a more energy-efficient 8-bit integer in order to reduce the probability of false alarm. These false alarms occur because when the maximum value of the variable ( $2^{16} - 1$ ) is reached, it restarts from 0.

Each mote also generates debug messages upon receipt of each message which will then be filtered in the subsequent phases.

Debug message are printed in the following form:

```
M <printing mote> R <seq number of the received msg> F <msg source>
```

## 3 Cooja

All the simulations are run using Sky motes.

I run multiple simulations with different configurations (a simple one can be found in 'sim' folder) and sent output to NodeRED opening a Serial Socket(SERVER) in each mote with different Listening Ports.

## 4 Node-RED

Node-RED is used to interface the motes in Cooja with IFTTT Applet. The flow used in this project is shown in the Figure below.

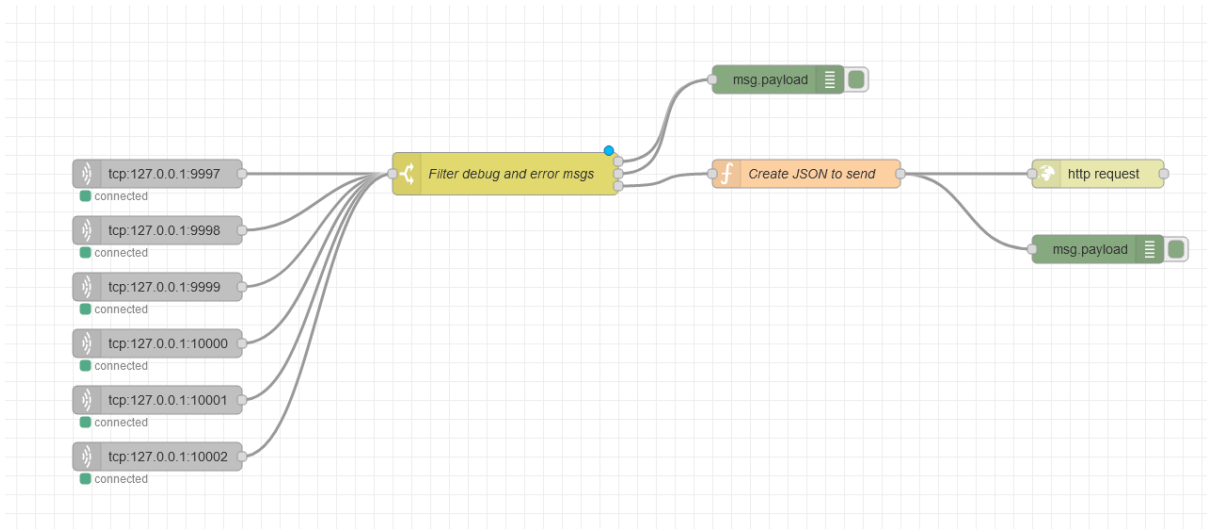


Figure 1: Node-RED Flow with 6 motes configuration.

Used nodes:

- **tcp in** node: used to connect and retrieve data generated by Cooja sockets. Each tcp in node connects to a single mote in Cooja.
- **switch** node separates the debug and error messages from alarm
- **function** node contains a javascript script which extracts sender and receiver IDs from the and creates a JSON file in the following form

```
{ "value1" : "<receiverID>", "value2" : "<senderID>" }
```

as required by WebHooks service in IFTTT described in IFTTT section.

- **http request** node performs a POST request to [https://maker.ifttt.com/trigger/mote\\_alarm/with/key/b8tv-8GQocm3YB3g-XSmyB](https://maker.ifttt.com/trigger/mote_alarm/with/key/b8tv-8GQocm3YB3g-XSmyB) with JSON created before as body, where *mote\_alarm* is the event and *b8tv-8GQocm3YB3g-XSmyB* is the personal API key provided by the service.
- **debug** node for debugging purposes.

## 5 IFTTT

It has been used the following applet to receive alarms through push notifications.

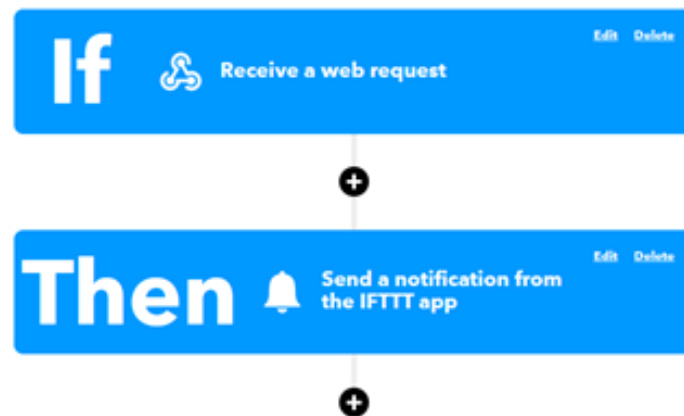


Figure 2: IFTTT Applet.

Each time the WebHooks service receives a POST http request with event `mote_alarm`, IFTTT app shows a push notification to smartphone. An example of received push notification can be seen in the figure below.

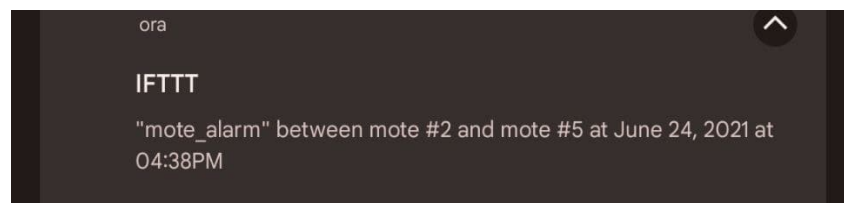


Figure 3: IFTTT Push Notification.