

IoT project - Keep your distance

Luca Ferraro (10748116), Fabio Losavio (10567493),
Bernardo Camajori Tedeschini (10584438)
<https://github.com/LucaFerraro/IoT-HomeChallenge>

A.Y. 2019/2020

Contents

1	Introduction	2
2	Setup	3
3	Program description	5
3.1	TinyOS	5
3.2	NodeRed	5
4	Examples	6

List of Figures

Listings

1 Introduction

2 Setup

The first requirement to try this project is having an **IFTTT** account opened. This because in the project, when two nodes are too close one to the other, a notification is net via email through IFTTT. To make this possible, it's necessary to create an Applet in our IFTTT account. The steps to create it are (once registered to IFTTT):

- Creating a new applet.
- In the section **This**, select the voice *Webhooks* and to trigger the reception of a web request.
- Name the event *Proximity_alarm*; the name is fundamental for the proper working of the application.
- In the section **That**, choose *Email* and insert the email where you want to receive notifications.

Now we can start the simulation.

First of all, we need to compile the code. To do this, we need to navigate from the terminal to the directory of the project and run the command **make telosb**: the result is the creation of a folder (named **build**) containing the executable file that is needed to add the motes inside the simulation framework **Cooja**.

The next step is opening **Cooja**. We need to create a new simulation and add as *Skymotes* the number of motes we want to use in the simulation. Pay attention that the number of nodes can't be greater than the parameter **NUMBER_OF_MOTES** that is specified in the file **Project.h** (it's been initialized to 5 from us, but it can be changed). Also, the number of motes cannot be in general greater than 14, that is the number of TCP connections we have added in the **NodeRed** flow we have prepared. In case the simulation would require a greater number of motes, the file **flows.json** must be changed adding the needed *tcp in* blocks. Once added the motes, it's necessary to start a socket on each one of them (each mote acts as a server); the ports on which the sockets must be opened are (if the file **flows.json** is not changed from the user of the program) the ones from 60001 to 60014.

At this point, we need to start **NodeRed**. To do this we need to:

- Open the terminal.
- Type the command **node-red**.
- Open a browser and connect to the local host (*http://127.0.0.1:1880/*).

- Import the file `flows.json`.
- Deploy the flow (check if the tcp blocks corresponding to the port on which the socket has been opened in Cooja is actually connected to that socket).

Note that, by default, the block *Req Params* is connected to the block *Web Req*, that sends a mail through **IFTTT** to one of ours mail. In order to let a user receive the notification as a mail, the block *Req Params* must instead be connected to the block *INSERT YOUR IFTTT KEY TO GET EMAILS*, which must be filled with the key of the mail registered in **IFTTT**. (All of this is explained also in the block *IFTTT instructions* inside the Nodered flow.)

Now everything is ready to start the simulation: just go again in Cooja and click on the *Start* button. By moving the motes around, it's possible to see the messages exchanged by the motes both in Cooja and in NodeRed.

3 Program description

3.1 TinyOS

3.2 NodeRed

4 Examples