POLITECNICO
MILANO 1863

IoT Project Report

KEEP YOUR DISTANCE

Mohammadreza Bakhshizadeh Mohajer 913233 - 10647452 | August 2020

Introduction

This project was created due to the situation established for Corona virus pandemic. It is aimed at helping people keep their distance in enclosed areas such as museums and galleries. The model is based on TinyOS as operating system, Cooja as the simulation environment, Node-red as connection between the nodes and the internet for sending notifications to a mobile phone.

In the simulation phase, if a node comes to a certain range from another mote an alarm procedure is triggered as follows.

1. The Node ID of the other node in the proximity is stored in the log memory of the mote.
2. The red LED on the node starts to blink while showing on its own screen an alarm message containing the Node ID of the violating node and then it sends the alarm message to Node-red.
3. In Node-red the message is displayed and then converted to a message format suitable for mobile phone notification and sent as one using the IFTTT platform.
4. The notification that contains first the ID of the violating node and then the ID of the other node is sent to be viewed on the handset of the phone.

Implementation

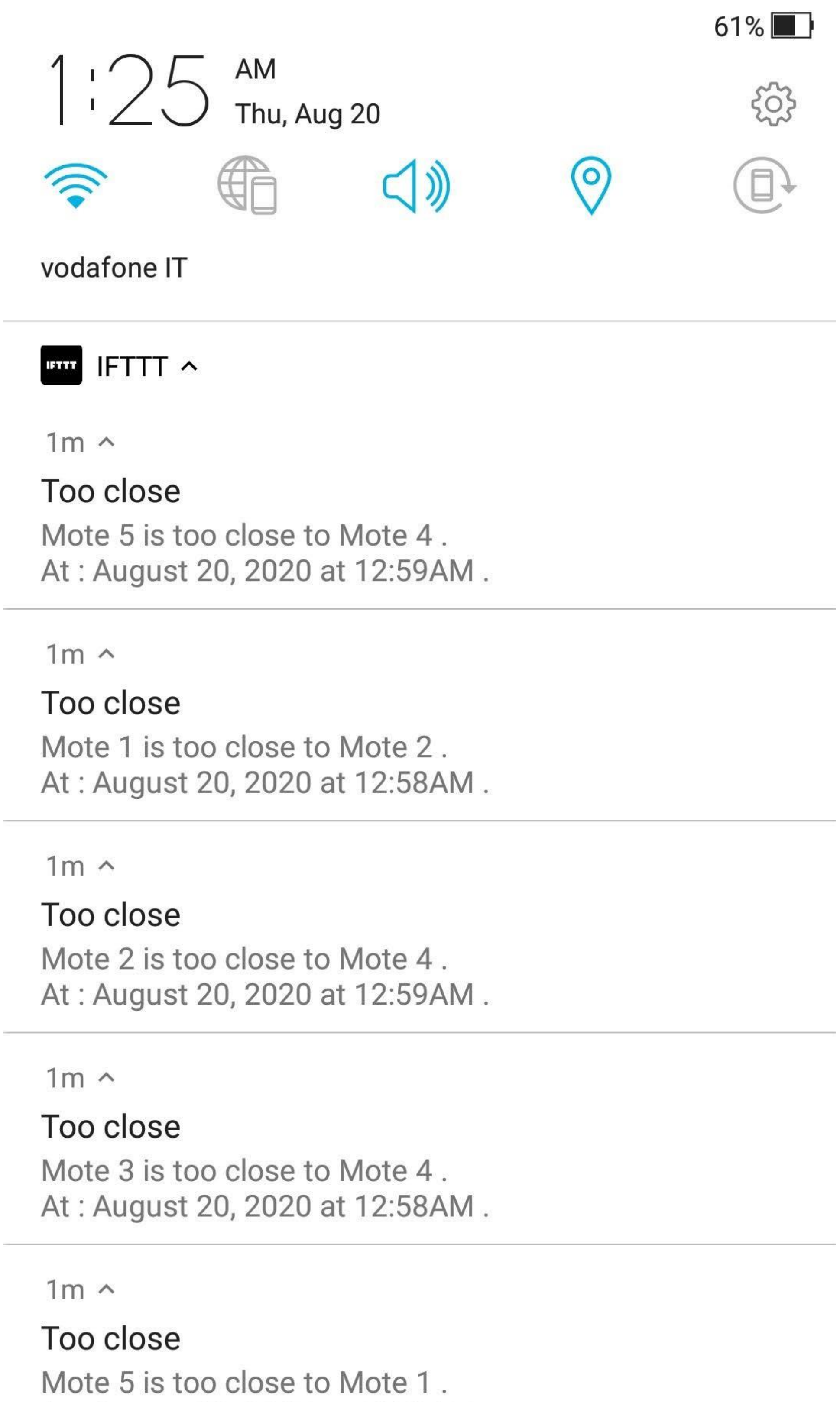
One code has been used for the nodes, which starts by booting the device then turning on the radio interface. Since we need to use the log memory on the mote, we must first make it ready. Instead of block memories that need mounting, log flash memories are ready to use only by first reading them, so the node calls for memory read. After making sure that the radio is on and the flash is ready to use the mote starts a periodic timer of 500ms. As the timer expires a message containing the Node ID of the mote is broadcasted in its proximity. Soon as two motes become close to each other, they receive the broadcast message so they extract the Node ID of the other node and first store it in log memory while toggling on and off the red LED of the mote as a sign of alarm and using the blue LED as an indicator that the log writing process has been executed correctly with success.

The alarm message is shown on the screens of the involved motes that breached the distance limit. For the case of two nodes involved, the incident is updated in the memory every 3 seconds until the nodes become separated and distanced again. But in the case that more than two motes are involved, memory writing rate becomes as fast as the messages received by the mote to keep up with the critical changes in the situation and log the incident with more details.

After recording the violating node ID in the memory, using a unique socket for each mote they are connected to Node-red. In Node-red, first the received messages from each TCP socket must be cleaned from the unwanted characters due to the bogus function of printf in Cooja. After trimming the messages, they are displayed in Node-red and then a switch statement classifies them according to which node is the conflicting mote, then we use a rate limiter to control the amount of the flow of notifications to the phone. The function after the limiter creates a template for each notification providing the event name and values which are the Node IDs to trigger the notification using the IFTTT platform. All the outputs of the template creator function are collected in a log_node_red.txt file in the directory of the program. In the end using a HTTP POST node we post the request to trigger the notification on the IFTTT website. The screenshot of the notifications received on the cellular phone handset are shown in Figure 1.

Please note that to see the full debug in Cooja, the commented printf functions must be uncommented. But due to the bogus functionality of printf in Cooja which adds unwanted characters to the messages, please run the whole program as it is to see the full performance without errors, then modify and uncomment the printfs.

All the log files in both Cooja, and Node-red plus the screenshots of the IFTTT applet notifications, IFTTT's activity log, Cooja implementation model of Cooja (.csc file), and Node-red model code are all included in the project file.



1m

Too close

Mote 5 is too close to Mote 4 .
At : August 20, 2020 at 12:59AM .

1m

Too close

Mote 1 is too close to Mote 2 .
At : August 20, 2020 at 12:58AM .

1m

Too close

Mote 2 is too close to Mote 4 .
At : August 20, 2020 at 12:59AM .

1m

Too close

Mote 3 is too close to Mote 4 .
At : August 20, 2020 at 12:58AM .

1m

Too close

Mote 5 is too close to Mote 1 .

Figure 1. Screenshot of the mobile phone notifications.