POLITECNICO
MILANO 1863

Project Report:
Smart Bracelets

Mirko Calcaterra
Nikolina Zallemi

August 28, 2022

TinyOs

In this Project we implemented a software using TinyOs and NodeRed.
This software is applied on smart bracelets, which are worn by children and their parents (2 couples), and help parents to keep track of the child's position. When the child goes beyond a communication ratio, an alert notifies the respective parent.
For the simulation we have used Cooja, making sure that every part of the code fulfills the project tasks. Each file making the simulation possible is explained below:

smartBracelet.h The first file which is important to be analyzed is smartBracelet.h. In this file we first defined all the constants, such the *number of possible connections*, the *length* of the *keys*, which we chose to be scalable (integer) with length of 20 byte, the *type* of the message (which can be BROADCAST [0], UNICAST [1] or for INFO [2]) and the timers.
Furthermore we have specified all the possible status code (that can be STANDING 3 WALKING 4 RUNNING 5 or FALLING 6) and the data structures, together with their components:

- **pairing datagram**: all the messages are sent in broadcast. Contains the type, key, address and ID
- **informative datagram**: containing the type, position of the child, with X,Y coordinates,status, and ID.
- **pairing acknowledgement datagram**: all the messages are in unicast. It contains the type and acknowledgement.

smartBraceletAppC.nc This is the file of the application where are defined all the components for the implementation.
Let focus on first on the **timer mechanism**:

- **TimerPairing**: Timer for the initial pairing. We randomly decided a value of 12500 ms, because it was not specified in the project requirements.
- **TimerTransmitting**: Timer for transmitting the position of the child and the kinematic status set to $10s = 10000ms$
- **TimerAlert**: If the parent's bracelet does not receive any message, after $60s = 60000ms$ from the last received message, a MISSING alarm is sent reporting the last position received.

The LED mechanism is explained in figure 4.

smartBraceletC.nc In this file we first manage the pairing between the bracelet, in particular we define who is the parent and who is the child. Then we have defined all the timers we needed.
In the second part of the file we put all the tasks which refer to broadcast and unicast messages, to transmit the info message from the child to the parent and the alarm message, because of the fall of the child or of the missing signal.

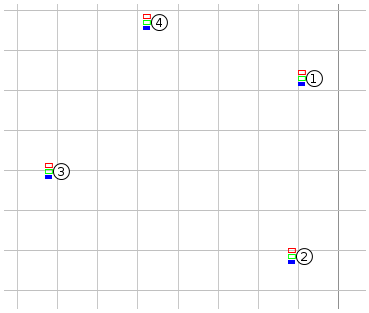

Figure 1: Pairing phase: If the parent and the child are in range one each other at least one time, the led becomes blue

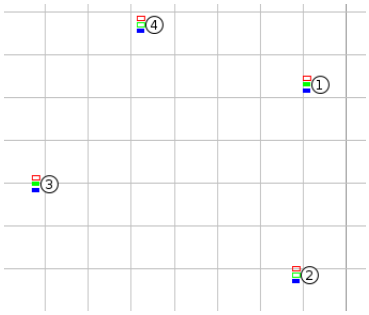

Figure 2: Info message phase: In this phase the parent's bracelet becomes green every time the child's bracelet transmits INFO messages.

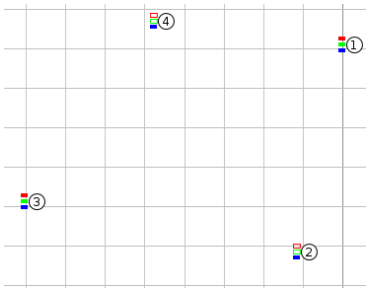

Figure 3: FALL and MISSING message: If the child falls or if he is out of range, the led of the dad becomes red.

Figure 4: LED mechanism. The image comes from the Cooja simulation

simulation.py Running the python code, the user can choose if he/she prefers to print only in terminal the simulation of the 4 bracelets or to save everything in a text file.

NodeRed Last but not least we connected in NodeRed our Cooja simulation. It was possible thanks to the **tcp node** (in NodeRed) and activating the Server Serial Socket in Cooja in Node 1 and 3, which are the parents motes.
Then thanks to a serial request node, every time the child is out of range, it is notified. Every minute it checks if it is still out of range or not. In the first case, an other notification is sent.
It is important We put all the code of Node Red in the file "*NodeRed_Simulation.txt*"

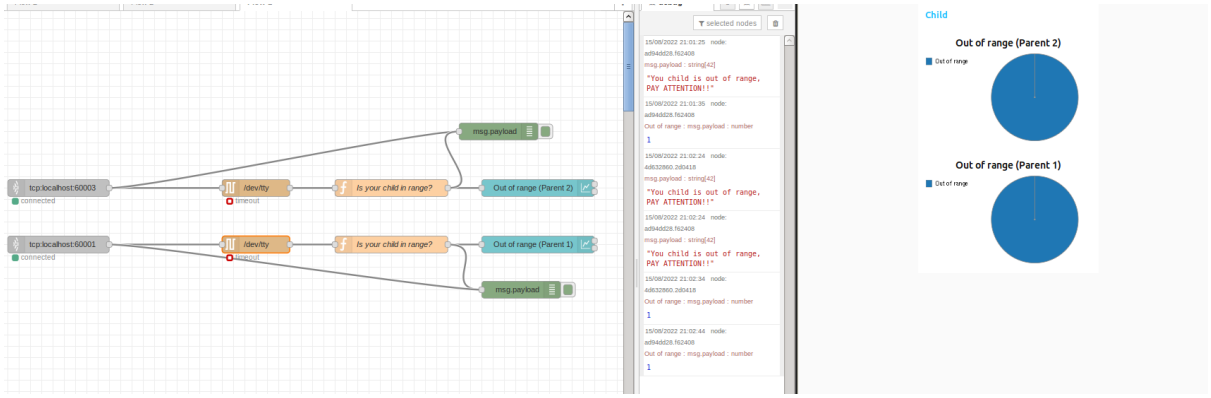

Figure 5: NodeRed scheme

If you want to check all the demonstration of the simulation, we provide a video called "Simulazione.mp4" on the link to the Github repository that you can download to see how it works