AY 2021/2022

Project report of

Smart Bracelets

Marco Bendinelli

1 Introduction

The project consists on the realization of a pair of smart bracelets: the bracelet is worn by a child and her/his parent to keep track of the child's position/behaviour and trigger alerts both when a child goes too far and when he/she is falling.
Summarizing, the bracelet operates in 3 different phases:

- **Pairing**: the parent's bracelet and the child's bracelet broadcast a random key that is pre-loaded at production time. Upon reception of the key, a device checks whether the received key is equal to the stored one. If yes, it stores the address of the source device in memory. Finally, a special message is directly transmitted to the source device to stop the pairing phase.
- **Operation**: the child's bracelet, every 10 seconds, transmits INFO messages containing the coordinates of the child and his/her kinematic status.
- **Alert**: if the child's status is FALLING, the parent's bracelet sends a FALL alarm. If the parent's bracelet does not receive any message after one minute from the last received message, a MISSING alarm is sent. In both cases it reports also the last position received.

2 Architecture

The IoT devices are implemented with the Coniki operating system.
The unique PROCESS_THREAD checks for event of expired timers. In particular, the behavior of the program at such events depends on the current state of a finite state machine (STATE_PAIRING or STATE_OPERATION).
The communication between the bracelets is handled by a broadcast and an unicast channel that are opened at the beginning of the process.

2.1 Pairing phase

At startup, the FSM is in the pairing state: every 10 seconds the bracelet sends a broadcast message containing the key of the device (saved in the configuration file).
Upon receipt of the broadcast message, the *broadcast_recv(..., ..)* function is called. It checks if the received key is equal to the stored one and, if so, it stores the address of the sender and it sends to it an unicast special message to stop the pairing phase and move to the next step.
Finally, the FSM's state of the sender and of the receiver changes in to the operation state.

2.2 Operation phase

2.2.1 Child's bracelet

It starts to send unicast messages to the paired device, containing the coordinates and the status of the child. In particular, I used an etimer to trigger a timer event each 10 seconds.
The coordinates are generated at each time, randomly; instead, the status of the child is pulled with the following probabilities: P(STANDING) = P(WALKING) = P(RUNNING) = 0.3 and P(FALLING) = 0.1.
Hence, out of 10 times, the first three states are chosen each 3 times; and the last state is chosen 1 time.

2.2.2 Parent's bracelet

During this phase, it just reacts to the received messages: **if the packet is sent from the paired device**, it prints the status of the child and it checks if it is a FALLING message; in that case, the bracelet has to print an alert message to the parent.
The recv_uc(..., ...) function, that handles the received unicast messages, implements also the mechanism to discover if the child is missed or not: every time it receives a message it starts a ctimer of 60 seconds. If a new message is received and the timer is not yet expired, the ctimer is reset.
Instead, if the ctimer runs out, the *callback* function is executed: it prints to the user a MISSING message and it restarts the ctimer from the previous expiration time.

3 Simulation

The simulation of the bracelets is done with the Cooja O.S..
In the log file I would like to emphasize how the code is following the main design requirements:

- the first 10 messages (then, I moved away the 2 devices) follow the probability distribution previously described;
- the parent's bracelet starts a timer every time it receives an INFO message, hence, the alert is sent by taking into account the last message received.
  INote that in order to simulate a missing child, I just moved a device out of the range transmission of the other bracelet.

Note that at the start up of the simulation, the 2 devices are far away; after 2 broadcast messages I brought them closer together.