INGI2146 : Z1 motes project

Emergency system for falling objects

Group 3 :
Crombé Henri
Declercq Mallory

April 2016

1 Introduction

With the growth of the Internet of things, the need for small connected devices that can monitor and automate processes have exploded. Embedded systems, such as the Zolertia Z1, is a perfect example of such device. Indeed, the Z1 mote can supervise complex processes using its internal and external sensors. And it can communicate with other motes using its wireless transmitter. If a mote acts as a border router, the whole system can reach (or be reached from) the Internet. Given these characteristics, we decided to design, implement and test an emergency detector for falling objects that can be monitored from the Internet.

Our emergency system consists of two Z1 motes. One mote acts as an acceleration probe and the other acts as a server. When the first mote detects that it is falling, it sends an emergency message to the server. Upon arrival of such a message, the server launches its emergency sequence. There exists a large range of application for this kind of emergency system. For example, it can be used as an emergency system for elevators. When the probe detects that the elevator is falling down, it asks the server to activate the emergency brakes. Another example of application could be the monitoring of elderly people. Indeed, as people get older, they lose more easily balance and tend to fall down more. As they sometimes need assistance in these case, our system could be used to constantly check if the person is alright and, if the person falls down, an emergency sequence is started (call ambulance and family, etc). And why not imagine using the system to detect the door activation of a mouse trap (see figure 2 in Annexe section) ? The major differences between these scenarios is the way the fall is detected. In the case of the elevator, the fall must last at least one second in order to exclude the case of a false positive fall. In the case of the people falling, the time of the fall is less than one second so the system must react more quickly. Our system implements a way of tuning the accelerometer parameters of the probe to handle these different situation. More specifically, we implemented a way for the user to tune and monitor the system from the Internet.

The following sections of the report are dedicated to describe our solution and the components used to solve the problem as well as the measurements we made to ensure that the system behave correctly.

2 Developed solution

As explained in the introduction, the developed emergency system consists of two different Z1 motes. One mote, the acceleration probe, is used to detect the fall of an object and to notify the server that the object is falling. The other mote, the server, is used to launch an emergency sequence when it receives an emergency messages from the probe. The two motes communicate together with their radio transmitter and exchange pre-defined messages wrapped into UDP datagrams to inform the other mote of their current state.

The fall detection system works as follow. The server mote is first started and begins to listen for UDP datagrams on port 3000. When the probe mote is launched, it first enter in an initialization phase. During this phase, the probe periodically sends INIT message to the server until it receives back an INIT_ACK message from the server. When this phase is over, the two motes are aware of each other and the emergency control system is activated. At this point, the motes periodically sends ALIVE messages to each other in order to be sure that the other is still reachable. If the acceleration probe becomes unreachable for fifteen seconds, due to poor wireless connectivity for example, then the server automatically launches the emergency procedure. Moreover, if the probe detects a fall, it sends EMERGENCY messages to the server to notify that it has to trigger the corresponding procedure. In case everything went well and the user wants to disconnect to probe from the server, he just need to press a button and the probe will send a DISCONNECT message to the server. This will safely unlock the emergency control system and reset the state of the server in case the user wants to reconnect afterwards. It is also to note that the emergency system can be unlocked by the probe in case of false-alarm. Now that the basics of the system have been explained, let's see how to tune the accelerometer to detect the different kind of falls.

In order to detect the fall of the object attached to the probe, we rely on the built-in ADXL345 accelerometer of the Z1 mote. Contiki provides a convenient interface to manage this accelerometer and this interface allows the user to register for specific interrupts. These built-in interrupts are trigger when some specific activity is detected. For example, the Z1 mote allows to register for an ACTIVITY event. Then, when the accelerometer detects some activity (i.e. the mote move for 0.5 second), it stops the current execution of the mote and launch a predefined callback function. In the context of our project, we use the built-in FREE_FALL interrupt to detect when an object is falling. This interrupt is triggered when the accelerometer detects that probe acceleration exceeds a fixed threshold for a given period of time. The acceleration and the time threshold for the FREE_FALL interrupt can be modified to better fit our needs. By default, the acceleration and the time threshold are respectively set to 563 mg and 160 ms. This means that an interrupt will be triggered if the probe is subjected to an acceleration of more than 563 mg for more than 160 ms. These default values don't suit all situation. Indeed, if we take the example of the elevator, these default values will trigger an emergency every time the elevator starts moving down because the elevator will be in some sort of free-fall for some milliseconds. The same thing could happen with people going down the stairs. In both cases, the system would have triggered the emergency procedure due to false-positive detection. Note that the acceleration is computed by summing the acceleration on the 3 axis (X, Y, Z), so it doesn't matter in which direction the object is falling, the fall will be detected anyway. Therefore, if we want the system to detect different kind of falls, we just need to adjust the fall time threshold so that the interrupts aren't triggered for nothing while still being triggered in the right cases. The measurements section is in part dedicated to explain how we chose the thresholds for the scenarios described in the introduction.

In addition to the emergency control, the server also acts as a border router. Thus, the server also operates as a bridge between the local mote network and the Internet. Meaning that the motes can be reached from the Internet and that the mote can contact a remote server if needed. The server also simultaneously run a small HTTP server that is accessible from the Internet thanks to the border router. The HTTP server hosts a little web-site displaying the state of the network as well as the state of the emergency system. This gives the user an easy way to remotely observe the status of the system. Still to make the system more accessible, the probe mote operates a REST server that allows the user to conveniently change the thresholds of the accelerometer. The user just need to connect to the REST server using Firefox and the IPv6 address of the mote. When the user is connected, Firefox provides an helpful UI to send CoAP commands to the REST server. Then, the user can easily sends CoAP commands to the probe to change the accelerometer thresholds in order to better fit their needs.

3 Measurements

3.1 Fall detection

As said previously, the parameters of the accelerometer can be tuned to detect different kind of falls. In the context of this project, the parameter that we need to adjust is the time the fall lasts before triggering an emergency. We take as hypothesis that this time can be approximated using the free-fall law of physics :

$$z = \frac{1}{2}gt^2 \tag{1}$$

In this formula, z is the traveled distance, t is the time of the fall and g is the gravitational acceleration. Now, let's consider the three different scenarios described in the introduction (i.e. The elevator, people falling and mouse trap).

In the case of the elevator, the time the fall lasts before triggering an emergency must be high enough to limit the case of false-positive. However, the system must be reactive enough to stay safe. Indeed, as the formula states, the elevator would have travelled 5 meters after 1 second of fall. Then, the system mustn't take more than 1 second to detect the fall and to activate the emergency brakes. Therefore, we would recommend using a time to fall threshold between 400ms to 600ms. This threshold should eliminate the case of false-positive and should guarantee that the system remains safe.

The case of a person falling is more tricky. Indeed, the duration of the fall is shorter and depends on the height of the person falling. Moreover, the accelerometer must distinguish between true- and false-positives (walking, stairs, etc). We made some tests to approximates an acceptable value for the threshold in this scenario. The test subject is 180cm tall and wear the probe in the top pocket of his shirt. The test procedure is rather simple. We fix the threshold and see if the emergency procedure is triggered when the test subject falls down on a bed. If we apply the formula given above, we get that the fall for the test subject should lasts approximately 540ms (given that the mote is located ($\approx$ 150cm above the ground). We started our experiments with the threshold set to 500ms then we decreased it until having false-positive results (i.e. emergency triggered when walking down the stairs). For each threshold, we repeated the fall ten times and observed the results. The table below shows the number of detected falls per threshold.

	Threshold			
	200ms	300ms	400ms	500ms
Detected fall	9	3	0	0

Table 1: Number of fall(s) detected given a fixed time to fall threshold

As the results suggest, when the threshold is set to 400ms or 500ms, the falls are not detected. When the threshold is set to 300ms, 3 out of 10 falls are correctly detected. The best results comes when the threshold is set to 200ms. However, when the threshold is so low (200ms), the emergency system tends to activate itself when walking down the stairs. So, either the system is too permissive and don't detect all the falls or either the system is too sensitive and trigger false-positive emergencies. In both cases, the results are unacceptable. This makes us feel that the system cannot be used *as it* to perform such a critical tasks that is monitoring people safety.

The case of the mouse trap with a retractable door that is activated when the mouse walks inside the cage is more straightforward. The probe is fixed to the retractable door and the door only falls or move when the mouse has entered the cage. Therefore, there shouldn't exist false-positive activation in that scenario. For the sake of the experiment, let's say that the trap door have to travel 10cm before being closed. The experiment set-up for this scenario is rather simple. The mote is placed 10cm above a pillow and when the system is engaged, the probe is released. In that case, and according the formula, the door fall should last $\approx$ 140ms. Therefore, we started our experiment by setting the time to fall threshold to 100ms. The test procedure have been repeated ten times for each threshold. The table below shows the results of the experience for this scenario :

	Threshold	
	50ms	100ms
Detected fall	10	0

Table 2: Number of fall(s) detected given a fixed time to fall threshold

The experiments reveal that the system is particularly reliable in this scenario. However, the threshold must be set to 50ms to detect the falls. As for the people falling, the expected value for the threshold given by the formula is higher than the threshold really needed. Therefore, for any given scenarios, the correct value for the threshold needs to be evaluated by testing the system in the real condition.

3.2 Range of usability

The whole emergency system relies on wireless communication to be operational. Then, the range of usability must be accurately determined. We define the range of usability as the range at which the system always reacts to an emergency (i.e. the server receives the emergency message from the probe). The experiment procedure consists of testing the emergency system of the probe at different distance from the server. The experiments are repeated ten times for each distance and the number of emergency hits at the server for each distance are recorded. In addition, the RSSI values (signal strength) for each distance are gathered. This should given an idea of the signal strength required for the system to be functional.

	Distance						
	0m	50cm	1m	2m	3m	4m	5m
Detected fall	10	10	9	4	2	2	1

Table 3: Number of emergency hit(s) on the server given the distance of the probe

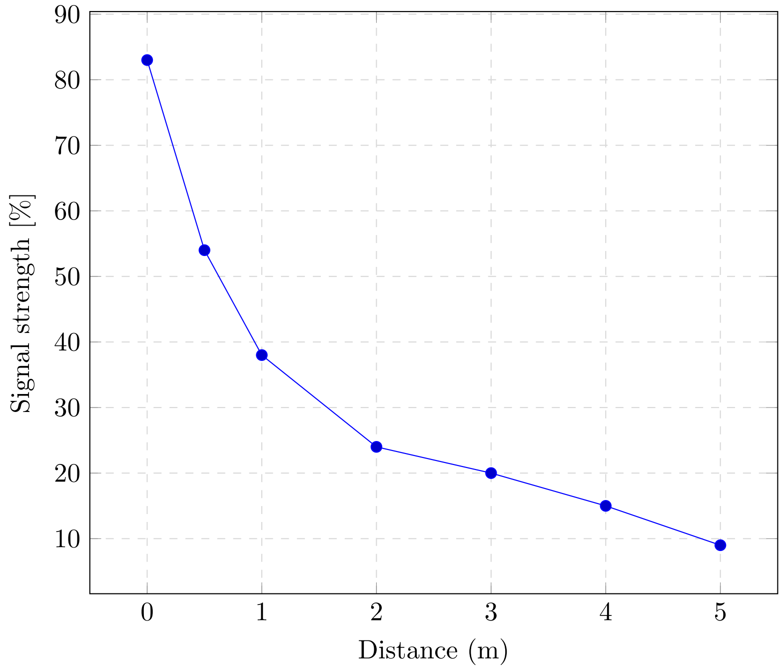

Figure 1: Mean Signal Strength (RSSI values) according to the distance in meter

Table 3 shows that the range of usability for the system is about 1 meter. This is clearly not satisfactory, even more if we consider that the test environment is relatively noiseless. Therefore we strongly recommend to add a proper antenna to Z1 mote. This should expand the range usability and make the communication more reliable. Furthermore, by correlation the results of the signal strength and the range of usability, we evaluate that the signal strength should at least exceeds 55% all the time for the system to properly operational. This shows again that a proper antenna on the two motes would expand the range of usage.

4 Further work

The system could be improved by using the signal strength value in addition with the ALIVE messages to detect when the probe becomes unreachable or if the system would become unreliable. In addition, we wanted to develop a low-battery emergency system for the probe but we didn't found enough documentation on the subject to implement this feature.

5 Annexe

Figure 2: Example of mouse trap with retractable door