

The Bomb

Real-Time systems project report

spring 2014

Authors:

Juraj Bielik 396329

Martin Šoltisík 395964

Adéla Otte 357635

Marek Tomášтик 374575

The aim of this project was to design and build a system which would simulate a bomb with several effects using real-time programming.

Description of the device

It is strongly recommended to read the whole description before activating the bomb.

Countdown

When the bomb is turned on, a countdown starts. The countdown is accompanied by dramatic light and sound effects, specifically, beeping and flashing signals every second. During the last 10 seconds of the countdown, the flashes are more frequent in order to increase the tension.

Without any attempts to defuse the bomb, it lasts exactly 30 seconds from activation to explosion. With certain attempts to defuse the bomb by an unacquainted user, the bomb is probably going to explode much sooner. Namely, several mechanisms are built-in to guarantee bomb explosion when someone is trying to defuse the bomb without additional knowledge. See chapter “Protection against unauthorized deactivation”.

Deactivation

Apart from “traditional” methods to defuse the bomb (cut off specific wire, transport the bomb to the orbit, etc.), one mechanism is implemented to safely defuse the bomb. During the countdown there are also displayed three characters on the display, apart from unpleasantly quickly decreasing remaining time. To defuse the bomb, it is necessary to enter these three characters in specific form (see chapter “Entering the deactivation code”). It is recommended to know Morse code.

Warning! Don't try to use arrows or other keys on the main device (see chapter “Protection against unauthorized deactivation”).

Entering the deactivation code

To enter a deactivation code it is necessary to use the touch sensor built-in in the bomb hardware. As written above, knowledge of Morse code is required. Here we refer to other resources, search “radioman's guide” or “telegraphic communication in 19th century”).

A “dot“ is to be entered by short push of mentioned sensor, a “dash“ by longer. Specifically, “dot“ should last at most 350ms, “dash“ should last more than 350ms. Recommended values are 150ms for “dot“ and one second for “dash“.

After successful deactivation of the bomb the flag signalization occurs.

Protection against unauthorized deactivation

In the case some unauthorized person comes to the active bomb and tries to deactivate it, there are several mechanisms to guarantee the bomb explosion.

The ultrasonic sensor

There is built-in ultrasonic sensor in hardware, which ensures that active bomb cannot be moved. In case ultrasonic sensor records movement of the device, the bomb detonates immediately.

Buttons on the device

There are arrow keys on the main device. Pushing them will make countdown go twice as fast. With faster countdown are associated faster light and sound effects. This could make an unauthorized user quite nervous, and also subsequent deactivation of the bomb very unlikely.

Entering wrong deactivation code

While entering deactivation code using touch sensor (see chapter “Deactivation“) it is strongly recommended to enter correct code and to enter “dots“ and “dashes“ precisely. Error is unacceptable! When an incorrect code is entered, the bomb detonates immediately.

Explosion

For the purpose of presentation we do not use explosive shots.

After countdown reaches zero, the bomb explodes. Explosive shot should be fired from the shooting mechanism, which is triggered by a motor.

Explosion is accompanied by sound effect and displayed image.

Design and programming

Cooperation

We worked in a team of four members. We did not have any problem with arranging the meetings – we met every week. There we have discussed the project, have done some work together and divided the rest of the work between us.

Everybody took Lego Mindstorms to home for some time, played with it, connected it to his/her computer and made a part of the source code.

Programming

We used NXC for programming. Although we had some problems with installing it and connecting with the device, in the end we have successfully connected it to all computers.

Concurrency

There are several concurrent tasks, scheduled with the `Precedes`, `ExitTo` and `StopTask` API functions. First we considered using a mutex `bombMutex` for mutual exclusion between tasks. But after identifying shared resources in our program we found out, that there is only one shared variable `speed`, which is read from the countdown task and written to from the buttons task. Since this is a case of a read-write “conflict“, no mutual exclusion is necessary.

Commented source code is attached.