

USER MANUAL AND TESTS: BEAT HERO

Execution:

To test the Beat-Hero game you simply need to choose which device to run it on (nRF52840 or LPC2105), compile the project (flash it to the board if using nRF), and enter a "Debug Session" to execute the code. To test other parts of the project, in main.c there is a constant called SESSION, where you can choose which part of the functions developed during the course you want to execute (all options are commented).

Controls:

The controls are very simple: button 1 and button 2 are used to play. With button 1 you control the left column (LEDs 1 and 3), and with button 2 the right column (LEDs 2 and 4). The LED will move down its respective column (first LED 1 or 2, then LED 3 or 4 respectively). When the bottom LED disappears, you have a limited time to press the appropriate button; if you do not press it, press the wrong one, or do not press it quickly enough, you will lose points.

As for button 3 (and 4 if using nRF), it is used to end the game when pressed once, but also to restart the game if held down for more than 3 seconds.

Interface:

When starting a game, all LEDs blink 2 times.

When losing a game, or when turning off the program by pressing buttons 3 or 4, LEDs 1 and 4 blink 4 times (in a \ shape).

When winning a game, all LEDs blink 3 times.

To verify the requirements of the assignment, perform the following tests:

1st Normal Gameplay Test:

Start the game. Play by pressing button 1 and button 2 when the light goes down.
Verify that every 7 measures the difficulty increases (Level 2 introduces silences, Level 3 double notes, Level 4 speed).

2nd Defeat Test:

Start the game and do not press anything (or intentionally fail). After failing 5 notes in a row, the game ends (defeat animation) and goes to sleep.

3rd Inactivity Test:

Since if you stop playing you will lose due to points, to test how the game goes to sleep due to inactivity you must modify line 309, which has the following form:

```
"if (j.compases_jugados >= JUEGO_COMPASES_MAX || j.puntuacion <=
    JUEGO_PUNTOS_MIN) {"
```

Here you must temporarily remove (or comment out):

```
"|| j.puntuacion <= JUEGO_PUNTOS_MIN"
```

By removing this line, we prevent the system from shutting down when losing due to points, making it so that if we do not press anything, after 10 seconds of inactivity the system goes to sleep and only wakes up after a new button press.

4th Reset Test (Long Press):

During a game, press and hold button 3 (or 4 if using nRF).

You will see that the LEDs turn off immediately.

Keep holding the button: after 3 seconds, the game restarts automatically.

5th Randomness Test:

Restart the game several times (long press on buttons 3 or 4 for 3 seconds) and verify that the initial note sequence is different each time.

6th Watchdog Test:

You can simulate a system error by uncommenting line 328 of the game file "juego_beat.c", thus causing an active wait that stops feeding the watchdog; after a few seconds the system restarts. After uncommenting, simply run the game and press button 3 or 4 to trigger the "error".

Authors:

Fernando Pastor Peralta (897113)

Guillermo Ledesma Uche (896594)

Group: TuesdayMorning K

Computer Engineering - University of Zaragoza, Academic Year 2025-2026