

**Architetture dei Sistemi**

**Di Elaborazione**

**Computer Architectures**

**02LSEYG**

**extra\_point[0]**

**Max 2 points**



Expected delivery of extrapoint1.zip must include:

- zipped project folder
- the setup.docx in pdf
- A **4 minutes video** with audio (.mp4 or .avi) explaining how the project works; the video has to show a software debug session with all significant peripheral windows opened; the audio must be a voice recording of you describing (in Italian or English) the behavior of the running system.

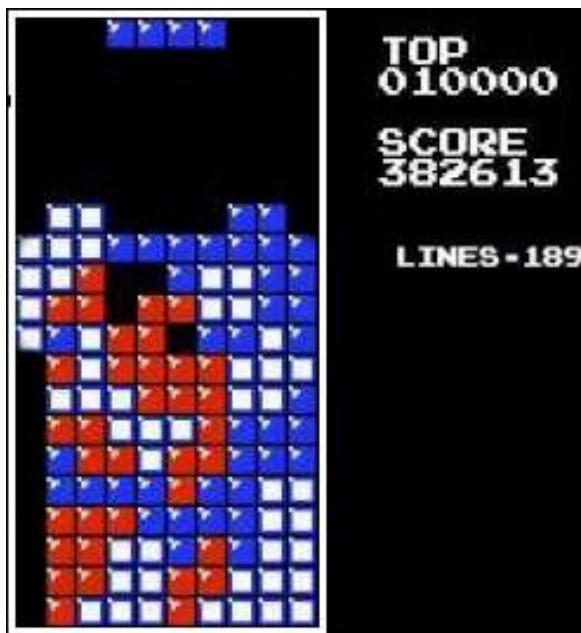
**Delivery deadline: 09/01/2026**

**Read carefully the request and deliveries!**

Purpose of Part 1: to acquire full confidence in the usage of the KEIL **software debug** (Compilation Target SW\_Debug) environment to emulate the behaviour of the LPC1768 and the LANDTIGER Board.

This part is evaluated to assign a maximum of 2 extra-points for qualified students taking the exam with a mark >= 18

# Tetris



Tetris' gameplay consists of a rectangular field in which pieces called **tetrominoes**, geometric shapes consisting of **four connected squares**, descend from the top-center. During the descent, the player can **move the piece horizontally** and **rotate** them until they touch the bottom of the field or another piece. The player's goal is to stack the pieces in the field to create horizontal lines of blocks. When a line is completed, it disappears and the blocks placed above fall one row. The game ends if the accumulated pieces in the field block other pieces from entering the field, a process known as "topping out". Common mechanics among Tetris versions include **soft drop** (increasing the descent of the piece) and **hard drop** (instantly placing the piece as far down as it can go).

The objective of Tetris is to collect as many points as possible during a gameplay session by clearing lines. The more lines cleared at once, the higher the score for a line clear. Clearing four lines at once using an I-shaped tetromino is referred to as a "Tetris".

You can try the official online version [here](#).

## Implementation details for LandTiger Board Emulator

In Keil µVision, use the **LANDTIGER emulator** (Compilation target: SW\_Debug) to implement a Tetris game.

If you have an available physical LandTiger Board you can directly develop the extra points without using the emulator (remember to change the compilation target, SW\_Debug is used only for the emulator). In that case, be a nice person and specify it in the box below.

I used the board to develop the game:

[ ] Yes [ ] No

Please deliver a zip folder with **all the files of your project** (you must save the project with all the compilation options you used).

Remember that if you import files in your project from another folder (e.g. when importing source files from another template) **Keil does not copy the file in your project folder automatically.**

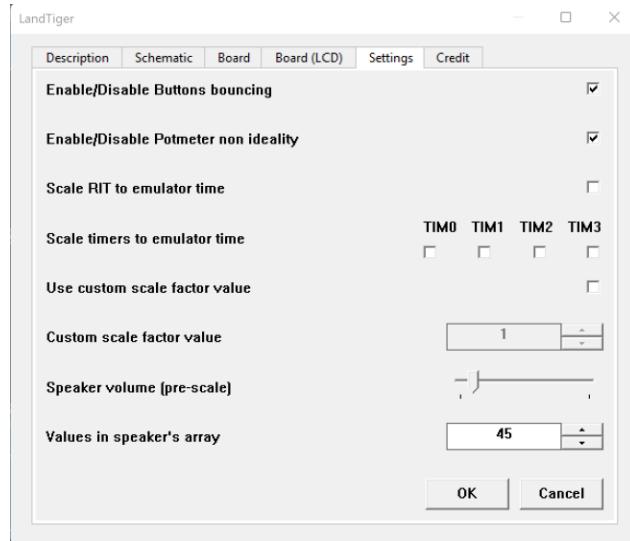
**Copy the files in your project folder first, then import them in Keil.**

Always check that all the source files are included in the project folder of the assignment before delivering it. If some files are missing, the project **will not compile on our computers** and you won't get any points for your project.

**The zip file should be named “extrapoint1.zip”.** Please use this exact name for the zipped folder and avoid exotic filenames such as: Extrapoint.zip, extra\_point.zip, extapont.rar, Progetto.tar.gz, tetris.7z, or extrapoint\_name\_surname\_999456\_final\_definitive\_version\_ishouldhavelearnedtousegit\_v4.20.zip.

Please attach any useful comments and a screenshot of your emulator configuration (**substitute the image below**) in an additional document (named setup.docx) .

**Your project will be evaluated according to your specification! The default configuration below will be used if you do not provide your configuration!**



Additional Comments (C-variable/defines defined in your code, e.g., scaling factors) :

**Very Important DISCLAIMER:** The use of AI is not forbidden neither strongly suggested. However, we are going to check the usage of AI in your project, and assign the score accordingly between you and the AI (i.e., if the AI wrote the 70% of the code, you will get only 30% of the total score).

Remember that AI agents will NOT be available during the exam 😊.

Additional Comments for the use of AI (If you did not use it, leave it empty):

Did I use the AI?

Am I going to be prepared for the exam either way?

What did I use? Gemini, Copilot, etc.

Which version did I use?

What did I use the AI for?

- First task
- Second task
- ...

Please add also a brief description in the video on the parts in which you used AI.

## Specifications (extrapoint 1)

Your task is to implement a simplified version of Tetris for the LandTiger board.  
**Take the image in the game description as reference.**

Please note that your project will not be evaluated based on your “artistic” skills, but on your programming skills. We appreciate nice graphics, but your primary objective is to make the game **playable**.

Use the display of the LandTiger to show:

- The playing field on the left side. You are free to choose the field size, but it should be big enough to fit 20 x 10 blocks
- Current score, high score and cleared line count on the right side

As for the game mechanics you are required to implement the following specifications:

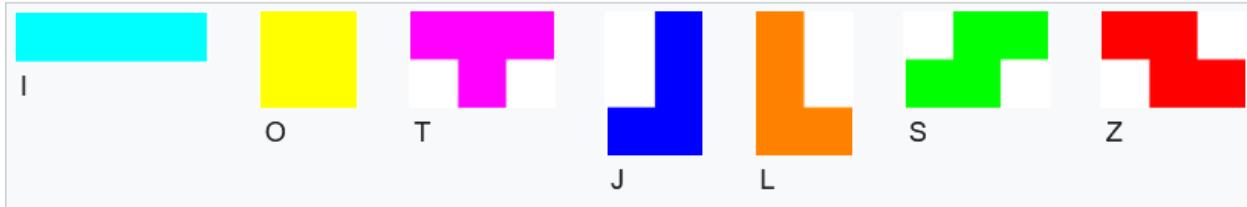
Spec. 1) **Playing field** - the playing field consists of a grid of 20 x 10 squares

Spec. 2) **Starting and pausing the game** – the game starts in the “paused” state with an empty grid, and the high score, current score and cleared line count set to 0. Pressing KEY1 after reset starts the game. Pressing KEY1 during the game toggles between the “paused” and “playing” states.

Spec. 3) **Tetrominoes** - tetrominoes are composed of **4 blocks** arranged in various shapes. Each block occupies the space of a square in the playing field grid.

You must implement the following tetromino shapes as a global variable:

- a. I-shaped
- b. O-shaped
- c. T-shaped
- d. J-shaped
- e. L-shaped
- f. S-shaped
- g. Z-shaped



Spec. 4) **Tetromino generation** - At the beginning of the game, a random tetromino appears on the top of the playing field. When a tetromino is placed on the playing field, a new random tetromino is generated in the same way

Spec. 5) **Tetromino movement** - Without other user interaction, the generated tetromino falls towards the bottom of the field at a speed of 1 square/second. The user can move the joystick left or right to move the tetromino and up to rotate it by 90° clockwise. When the tetromino cannot move down because it reached the bottom of the playing field or reached the top of another block, it stops there and is considered placed on the field.

Spec. 6) **Soft drop** - If the user holds the joystick down, the fall speed of the tetromino increases to 2 squares/second while the user is holding the joystick down. When the user releases the joystick, the speed returns to 1 square/second.

Spec. 7) **Hard drop** – If the user presses KEY2, the current tetromino falls instantly. The fall should follow the rules described in specification n. 5 regarding tetromino movement.

Spec. 8) **Clearing lines** – After a tetromino is placed on the field (see specification n. 5), the game should check if one or more horizontal lines of 10 squares have been completely filled with blocks. In that case, those lines are cleared, i.e., the blocks in them are eliminated, and the lines above them that still contain blocks are shifted down by an appropriate amount, so that there are no gaps between lines containing blocks. Multiple lines can be cleared with a single tetromino: when 4 lines are cleared (the maximum possible amount), the move is called a “tetris”

Spec. 9) **Scoring** – Placing a tetromino in the field awards 10 points.

Clearing a single line awards 100 points plus the 10 points from the tetromino placement.

Clearing multiple lines awards 100 points x number of cleared lines plus the 10 points from the tetromino placement.

A tetris awards 600 points plus the 10 points from the tetromino placement.

Spec. 10) **Lose conditions** - The game finishes when the placed tetromino goes over the top of the playing field.

If the generated tetromino that appears at the top of the field overlaps with another block, the game ends as well.

Spec. 11) **Game end** - When the player loses, if the score is higher than the current high score, it is saved as the new high score. The game then returns to the initial state described in specification

n. 2, the only exception being that the high score is not reset to 0 and reflects the new high score.  
The player can then start a new game by pressing KEY1.