
# Arduino Tetris Game:
## Code & Implementation

**Student:** Ciocian Alexandru-Nicolae

**Index/Title/Items:**

# 1. Overview

The Tetris Game Arduino Code implements a simplified version of the classic game Tetris using an Arduino microcontroller. This documentation provides an overview of the code structure, its theoretical background, and the hardware implementation details.

While doing this project I have encountered multiple challenges such as: the LED Matrix Module is 90$^{o}$ flipped, meaning that the x and y axis are now reversed, or finding a solution to keeping the tetrominos placed and creating another one to fall. These challenges have been overcome and the project is done.

# 2. Theoretical aspects

*"Tetris is a puzzle video game created in 1985 by Alexey Pajitnov, a Soviet software engineer.[1] It has been published by several companies for multiple platforms, most prominently during a dispute over the appropriation of the rights in the late 1980s. After a significant period of publication by Nintendo, in 1996 the rights reverted to Pajitnov, who co-founded the Tetris Company with Henk Rogers to manage licensing.*

*In Tetris, players complete lines by moving differently shaped pieces (tetrominoes), which descend onto the playing field. The completed lines disappear and grant the player points, and the player can proceed to fill the vacated spaces. The game ends when the uncleared lines reach the top of the playing field. The longer the player can delay this outcome, the higher their score will be. In multiplayer games, players must last longer than their opponents; in certain versions, players can inflict penalties on opponents by completing a significant number of lines. Some versions add variations on the rules, such as three-dimensional displays or a system for reserving pieces."*
[Wikipedia]

## Tetris Mechanics

- Grid Size:
  - The number of vertical LED matrix modules utilized determines the size of the grid on which the game is played, in this case 8xN.
- Tetris Pieces:
  - There are seven different kinds of tetrominoes in the game, and each can rotate in four ways by 90$^{o}$.
  - A 4x4 matrix is used to represent tetrominoes, by binary representation of the ON/OFF state of the LEDs.
- Game Controls:
  - The output of an Analog joystick is used to move the falling tetrominoes horizontally and the descent is automatically done; No user input is needed.
  - The SEL_PIN button is used to rotate the falling tetromino.
- Game logic:
  - Any block that reaches the end of the 8xN grid is considered as placed.

- If a block is placed, a Boolean matrix of dimension 8xN is initiated at those values with true.
- After any block has been successfully placed, the game check if any rows are full so that it can be deleted and the rows are shifted downwards. This also means that the grid is re-evaluated with the true/false Boolean values.

# 3. Implementation

The implementation has firstly been done in a simulated environment called WOKWI. This digital simulated environment has been chosen because of its simplicity that allows for complex code implementation even with a simple electronical circuit.

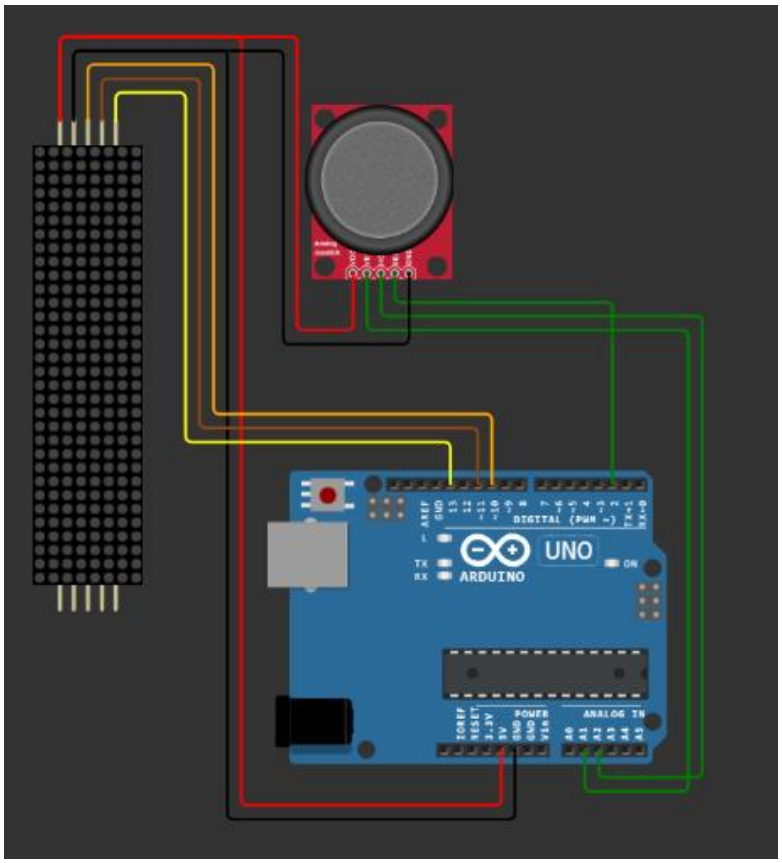



*Figure 1 - Components used in WOKWI to simulate the project*

<u>Components used for building this project:</u>

- Arduino Uno microcontroller
- Analog Joystick
- Max7219 LED Matrix Module
- Mother-Father cable connections

## Hardware Setup

LED Matrix:

- Pinout:

| Pin Name | Pin Nr. |
|---|---|
| SS_1_PIN | 11 |
| DIN_1_PIN | 10 |
| CLK_PIN | 13 |

Library Used: SPI, MD_MAX72xx

Analog Joystick:

- Pinout:

| Pin Name | Pin Nr. |
|---|---|
| SEL_PIN | 2 |
| VERT_PIN | A1 |
| HORZ_PIN | A2 |

- Pin “A1” and “A2” are analogical pins that must be used because of the nature of the Analog Joystick.

## Software code setup

Game Constants

- Grid Size:
  - GRID_WIDTH: 8
  - GRID_HEIGHT: Defined based on the number of vertical LED matrix modules used.

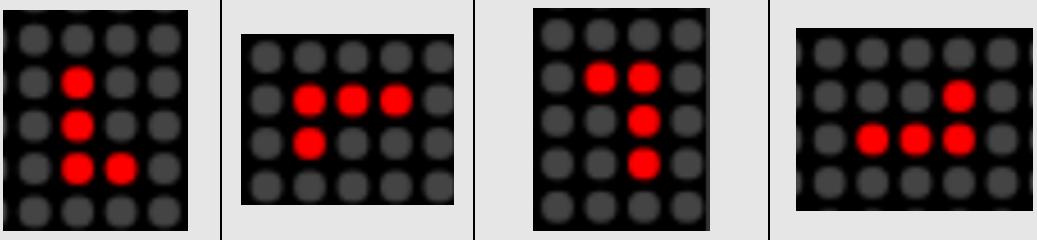
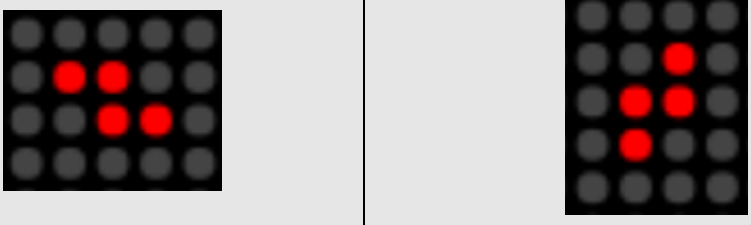
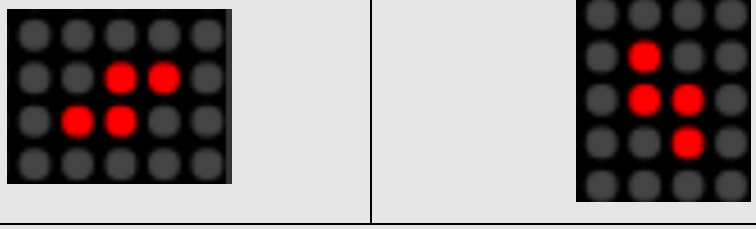
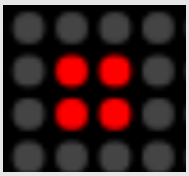
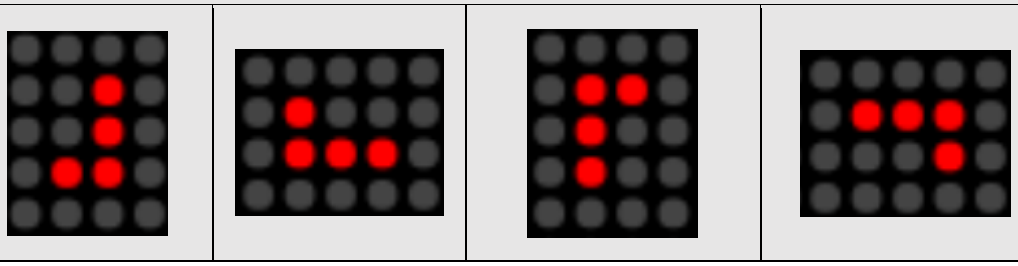
Initial Cursor Position:

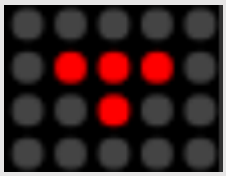
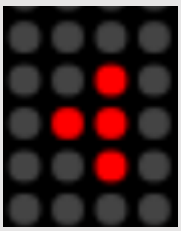
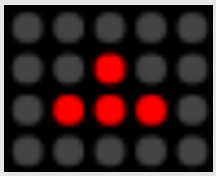
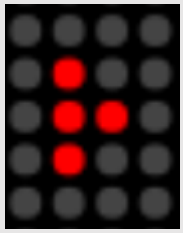
- cursorX: 0
- cursorY: 0

Tetris Block Types:

The game defines seven tetromino types, each with four rotations.

The tetrominos are: “I”, “L”, “Z”, “S”, “O”, “J” and “T”.

| Tetris block | LED representations – Rotations 1 - 4 |
|---|---|
| “I” |  |
| “L” |  |
| “Z” |  |
| “S” |  |
| “O” |  |
| “J” |  |

| | | | | |
|---|---|---|---|---|
| “T” |  |  |  |  |

Main Game Loop

The loop() function controls the game flow, handling block creation, movement, and user inputs.

Methods used for the game logic (no particular order):

- *getRandomCursor():* Randomly sets the initial cursor position within the grid.
- *getRandomPiece():* Randomly selects a tetromino type and rotation.
- *placeTetrisBlock():* Places the falling tetromino on the grid when it reaches the bottom or collides with other blocks.
- *checkPlacement():* Checks whether the current tetromino can be placed at its current position.
- *checkAndDeleteFullRows():* Checks for and deletes full rows, shifting upper rows down.
- *handleGameControls():* Handles user input from the analog joystick to move the falling tetromino.
- *rotateFallingBlock():* Handles tetromino rotation based on user input.

Display Functions

- *displayFallingBlock():* Displays the falling tetromino on the LED matrix.
- *clearFallingBlock():* Clears the previous position of the falling tetromino.
- *placedBlocksController():* Updates the LED matrix display with placed blocks.

# 4. Conclusions

In conclusion, the Tetris game implemented with the Arduino and the LED Matrix Module simulates a very basic Tetris game with only the falling and placing of blocks.

During the making of the project I have learned how to use the SPI library and utilize it for the LED Matrix Display, but it can easily be used for other specific implementations or projects/mini-projects.

# 5. Bibliography


- Tetris - Wikipedia
- https://wokwi.com/
- MD_MAX72xx LED Matrix Arduino Library: MD_MAX72XX Class Reference (majicdesigns.github.io)

# 6. Annexes

- GitHub Repository: https://github.com/Ciocian-Alex/Proiecte-Personale/tree/16a066d098a3dfd368363ca829a779734ced12be/Arduino-Projects/Tetris