

Tic Tac Toe

Introduction

Name: Manea Andrei Iulian

Group: 331CB

Assistant: Alex Văduva

Description

The project is a physical implementation of the game Tic Tac Toe. Additionally, a speaker will play music in the background for better ambiance, there will be a buzzer to introduce a time limit for each move; it will signal when the player has 5 seconds left. Also, there will be a LED strip for a better visual effect.

Motivation

I think this is a fun game that I always used to play as a kid, so it's an additional motivation for me to implement it. It's simple, yet effective, and the fact that I can have „a pocket version” of it would mean a lot to me.

General description

I used two Arduino UNO R3 as μ Cs. For inputs, I will have a keyboard with four buttons for moving around the Tic Tac Toe grid and an additional button for selecting in which tile a symbol will go. The music will be played using microSD card reader for the input and a speaker for the output. A buzzer will also be used to signal the last five seconds until the player will have to make a move and the LED Strip will be used for visual enhancement. Finally, an LCD will be used to display the actual game.



Hardware Design

Materials:

- 2 x Arduino UNO R3
- 2 x mini breadboard
- 1 x button
- 1 x keyboard with 4 buttons
- 1 x LCD 1.44" SPI and ST7735 controller
- 1 x microSD/SDHC card module reader
- 1 x SDHC card

1 x speaker
1 X buzzer
1 X 220Ohm resistor
1 x XPT8871 mono audio amplifier
plenty of male-male and male-female wires

Circuit Layout



Circuit Design



Circuit Implementation

1. Arduino Uno (Rev3)

- **Role:** Central controller. Drives SPI for the display and SD, outputs PWM for audio/buzzer and scans the keypad.
- **Power:** Can use its onboard 5 V regulator (from VIN) or the two external AMS1117 regulators shown.

2. Voltage Regulators (U1 & U3)

- **U1** (AMS1117-5 V):
 - **Input:** 7-12 V
 - **Output:** 5 V rail feeding:
 - Arduino 5 V pin
 - WS2812 LEDs
 - PAM8403 audio amp
 - Buzzer (through its series resistor)
- **U3** (AMS1117-3.3 V)
 - **Input:** same VIN
 - **Output:** 3.3 V rail feeding:
 - TFT module
 - MicroSD module

3. TFT Display

- **Power & Backlight**
 - VCC → 3.3 V
 - GND → common ground
 - LED + (backlight) → 5 V
- **SPI Signals** (sharing the SPI bus)
 - SCK → D13 (hardware SCK)
 - MOSI → D11
 - CS → D10
 - DC → D9 (Data / Command select)
 - RST → D8

4. MicroSD Card Module

- **Power**

- VCC → 3.3 V
- GND → common ground
- **SPI** (same bus, separate CS)
 - MOSI → D11
 - MISO → D12
 - SCK → D13
 - CS → D4
- Separate CS allows the Arduino to select either the TFT or the SD card independently.

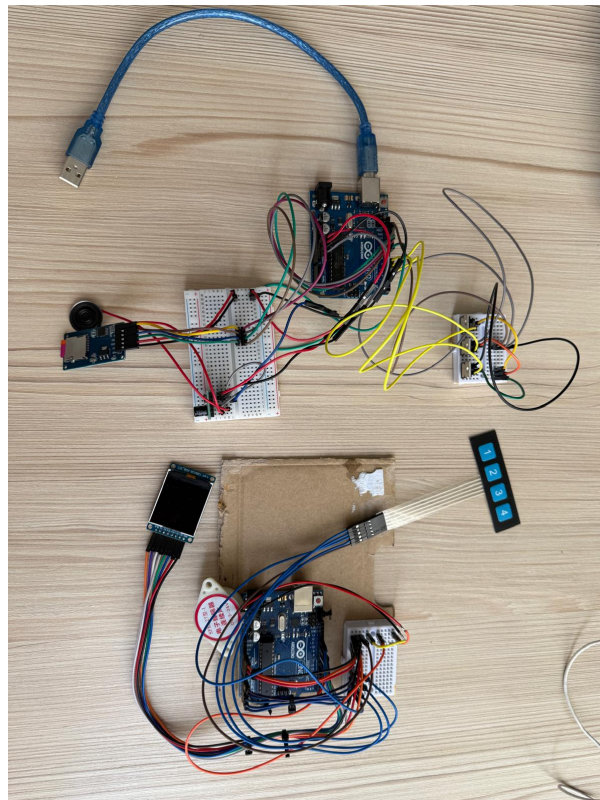
5. Audio Amplifier Module (PAM8403) + Speaker

- **Power**
 - VCC → 5 V
 - GND → common ground
- **Inputs**
 - L IN / R IN → PWM-filtered audio from Arduino
- **Outputs**
 - L OUT / R OUT → speaker
- **Function:** Boosts the Arduino's low-level PWM audio to drive a few-watt speaker.

6. Passive Buzzer

- **Connections**
 - One leg → D3 (PWM) through a 220 Ω resistor
 - Other leg → GND
- **Function:** Generates tones / alerts by toggling PWM at audio frequencies.

Real-life Circuit



Software Design

- Environment: Arduino IDE
- Libraries: Adafruit_GFX; Adafruit_ST7735; TMRpcm

Given the fact that I have used two Arduino UNO, I have two separate codes. The first one runs the actual game logic and handles the input from the keyboard, allowing the player to move on the board. It also handles the timer for the buzzer, which makes a sound when the player has 2 seconds left for his turn. The second one is used to handle the speaker, taking the input from the microSD card reader.

1. Tic-Tac-Toe Game

- `setup()`
 - Initializes the TFT display via SPI (`tft.initR()`), clears the screen.
 - Configures buttons (GPIO inputs with pull-ups), the buzzer pin (GPIO output), and seeds the random number generator using an ADC read.
 - Calculates the grid's starting coordinates, resets the game board, and starts the first turn timer.
- `loop()`
 - End-of-Game Handling: If `gameover` is true, it shows either "Player X wins!", "Player O wins!", or "Draw!"; then waits `resetDelay` ms before clearing and restarting.
 - Turn Timer
 - Between `warningThreshold` and `moveTimeout`: sounds the buzzer via `tone()` (PWM) and displays a countdown once per second.
 - After `moveTimeout`: automatically picks a random free cell for the current player, draws the mark, and checks for win/draw. If the game continues, it swaps players and resets the turn timer.
 - Otherwise: silences the buzzer and clears any warning message.
 - Input & Rendering: Reads button presses (debounced in software) to move the cursor or place a mark, then highlights the current cell and redraws any X/O.
- Supporting Functions
 - Board management: `resetBoard()`, `drawGrid()`, `highlightCell()`, `drawMark()`
 - Game logic: `moveCursor()`, `placeMark()`, `checkWin()`, `isBoardFull()`
 - Turn management: `startTurn()`, `clearWarningArea()`, `displayTimerWarning()`, `displayWinner()`, `displayDraw()`
 - Random fallback: `randomMove()` for timeouts

2. Speaker

- `setup()`
 - Initializes the SD card over SPI (chip-select on pin 10).
 - Configures the TMRpcm library's CS and speaker pins and sets playback volume.
- `loop()`
 - If no audio is currently playing, calls `audio.play("powerup.wav")`.

Here is the full code: https://github.com/LilAndy2/PM_Project

Results

Here is a short demo of the game in action.

https://youtube.com/shorts/_YSpUI_ECTo?feature=share

Conclusions

- It was a cute project which took me a lot of time to make.
- I had to learn the basics of Arduino from scratch and try to make something functional.
- While the aesthetic can be improved by adding colors and decorations, it is the best I could do in the time given.
- While the experience was very frustrating at times, it was also fun to learn something new.

Download

Here is an archive with all the files of the project.

[pm_project_andrei_manea.zip](#)

Journal

06.05.2025 - decided on the project, wrote the description and the hardware materials
11.05.2025 - materials bought, starting on the hardware design
18.05.2025 - finished hardware design
19.05.2025 - uploaded circuit design and circuit layout
23.05.2025 - decided that LED strip component does not fit with the current hardware, so I removed it
26.05.2025 - implemented the software
27.05.2025 - filmed the demo of the game

Bibliography / Resources

Hardware Resources

- <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>
- <https://shorturl.at/ccQdk>
- https://www.electrokit.com/upload/product/41015/41015739/41015739_-_SD_Card_Module.pdf

Software Resources

- <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-gfx-graphics-library.pdf>
- <https://www.tinkercad.com/projects/Tic-Tac-Toe-Game-on-Arduino-for-Beginners>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2025/avaduva/andrei_iulian.manea



Last update: **2025/05/27 11:18**