

# Applying Arduino and individually addressable LED strips to the development of interactive materials to promote computer science undergraduate course

Marlon Fabichacki Pereira\*, Marcio Seiji Oyamada\*

\* Computer Science – State University of Western Paraná (UNIOESTE) – Cascavel, Brazil  
{marlon.pereira1, marcio.oyamada}@unioeste.br

**Abstract**—Arduino is a hardware and software platform known for its ease of use and accessibility, and can be operated by people with different levels of technical knowledge. With such a powerful tool, the goal of this research was to develop interactive and engaging materials that combine Arduino and individually addressable LED strips to attract people to computer science course in exhibitions and showcases by demonstrating the capabilities and accessibility of the technology. This paper presents two demos created with addressable LED. The first is an LED race based on the Open LED Race project, and the second is a Connect-4 game using an LED matrix.

**Keywords**—Interactivity; Games; Open Source Software.

## I. INTRODUCTION

The research was inspired by the Open LED Race Project (OLR) [1], a game created during Arduino Day Sevilla 2019 and subsequently developed as an independent open source project. The game itself consists of a simulation of a slot car track using an individually addressable LED (Light-Emitting Diode) strip, a different kind of LED strip where each LED or section of LEDs can be individually controlled in aspects such as brightness and color. Players are represented by different colors that light up on the LED strip, simulating a car moving forward and receiving an acceleration pulse each time a button connected to the Arduino is pressed. Our motivation with the Open LED Race was the lack of prominent materials available to promote our computer science course at exhibitions in which we frequently participate. This resulted in low interaction with the public and thus lower utilization of these events. The Open LED Race Project provided an opportunity to develop not only the race, but also other games and interactive prototypes that conveyed the possibilities of codifying and representing our world through technology using simple materials such as the individually addressable LED strips.

In this work, the Open LED race project was adapted and a completely new project was developed based on the acquired expertise, implementing the traditional Connect-4 game into

an addressable LED matrix. This work is organized as follows: Section II presents the basics of addressable LED, Section III describes the Arduino platform, IV describes the Open LED Race implementation, and Section V describes the Connect-4 implementation. Section VI presents the conclusions and future works.

## II. INDIVIDUALLY ADDRESSABLE LED

The individually addressable LED strips used in this research are based on two different drivers, the WS2811 [2] and the WS2812 [3], which use a similar 5050 LED (5050 indicates the size of the LED, which is 50 millimeters high and 50 millimeters wide). The WS2811 driver is a more primitive version, with the memory located outside the 5050 LED along the strip. It can drive three different consecutive LEDs and offers an economical but less precise solution when its resolution is one-third the number of LEDs in the strip (see Figure 1). In the WS2812 driver shown in Figure 2, the memory is located within the 5050 LED, creating a strip where each LED is addressable, resulting in one-to-one resolution.



Figure 1. A WS2811 LED strip section - One memory per three LEDs.

Both LED drivers are compatible with Neopixel [4], Adafruit's brand of individually addressable LEDs that come in many shapes such as strips, circles and matrices, allowing for a variety of representations powered by this technology.

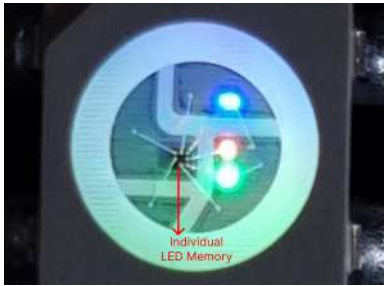


Figure 2. A WS2812 LED strip single LED - One memory per LED.

### III. ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software [5]. Its language is based on C and C++ and includes several libraries that facilitate the development of projects through its own integrated development environment, Arduino IDE. One of the libraries available for Arduino is the Adafruit Neopixel Arduino Library, which includes several functions for easy setup and control of Neopixel LEDs. This interface enabled the development of the projects using an Arduino Nano V3 and an Arduino Uno R3 board, whose technical specifications can be seen in Table I and Table II [6] below, respectively.

Table I  
ARDUINO NANO V3 SPECIFICATION

Microcontroller	ATmega328
Clock speed	16MHz
SRAM	2KB
EEPROM	1KB
Flash memory	32KB
USB connector	Mini-B USB
I/O voltage	5v
Input voltage (nominal)	7-12v
DC current per I/O pin	20mA
Built-in LED pin	13
Digital I/O pins	14
Analog input pins	8

In the Arduino IDE, the Adafruit Neopixel Arduino Library provides some simple introductory and well-documented examples of how the individually addressable LED strips work. The implementation is essentially limited to the two main functions used in an Arduino, `setup()` and `loop()`, as shown in Figure 3. The `for()` loop in the code is responsible for setting the color of the LEDs in the strip step by step through the

Table II  
ARDUINO UNO R3 SPECIFICATION

Microcontroller	ATmega328p
Clock speed	16MHz
SRAM	2KB
EEPROM	1KB
Flash memory	32KB
USB connector	USB-B
I/O voltage	5v
Input voltage (nominal)	7-12v
DC current per I/O pin	20mA
Built-in LED pin	13
Digital I/O pins	14
Analog input pins	6

`setPixelColor()` method of the `Adafruit_NeoPixel` class, selecting a specific `n` LED of the strip at a time.

```
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

#define DELAYVAL 500 // Time (in milliseconds) to pause between pixels

void setup() {
  pixels.begin(); // INITIALIZE NeoPixel strip object (REQUIRED)
}

void loop() {
  pixels.clear(); // Set all pixel colors to 'off'

  // The first NeoPixel in a strand is #0, second is 1, all the way up
  // to the count of pixels minus one.
  for(int i=0; i<NUMPIXELS; i++) { // For each pixel...

    // pixels.Color() takes RGB values, from 0,0,0 up to 255,255,255
    // Here we're using a moderately bright green color:
    pixels.setPixelColor(i, pixels.Color(0, 150, 0));

    pixels.show(); // Send the updated pixel colors to the hardware.

    delay(DELAYVAL); // Pause before next pass through loop
  }
}
```

Figure 3. Simple example of an individually addressable LED strip controller code.

### IV. OLR IMPLEMENTATION

The Open LED Race (OLR) project [1] was implemented in an Arduino Nano V3 driving a WS2811 LED strip, contrary to the project's indication to use a WS2812 LED strip, which was not available in the laboratory. The limitation imposed by the lower resolution of the LED strip led us to readjust the open-source OLR code, minimizing animations and constants such as acceleration, friction and gravity, thus making the race functional with an LED strip of 129 LEDs controlled by only 43 addressable memories.

In addition to modifications to the default parameters, there were user suggestions that led to new features that were

implemented using a 128x64 OLED (Organic Light-Emitting Diode) panel with Adafruit's SSD1306 driver [7] and controlled by the Adafruit's SSD1306 Arduino library. The OLED display was used to show an options menu before the start of each race, allowing configuration of the lap count and enabling or disabling the physics calculation of the track slope in the next race. This interaction is completed by using the same two buttons that players use to race. One button is used to cycle through the options, and the other is used to change the values of the options. When the race starts, the display will show a countdown, followed by informations of the race, such as the current winner, the number of laps each player has completed, and the duration of the fastest lap in seconds. When the race is over, the display shows the winner along with the fastest lap and returns to the options menu.

The OLED was supposed to display the duration of the race in real time, but due to the limited capabilities of the Arduino Nano V3 and the processing required to output information to a screen, this idea was not feasible, so we opted for the fastest lap status, once it has to be calculated and displayed only when the lap is complete, resulting in less output and more stable behavior.

To ensure the safety of the project during an exhibition, it was enclosed in a wooden box with exits for the cables, buttons and OLED display, as shown in Figures 4 and 5.

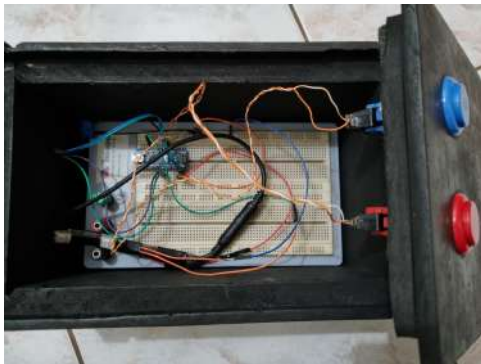


Figure 4. OLR Arduino project set up inside a wooden box.

## V. CONNECT-4 IMPLEMENTATION

After completing the OLR implementation and familiarizing ourselves with the operation of the individually addressable LED strips and their libraries, our research was aimed at discussing what other games could be displayed using the various form factors of the individually addressable LEDs, which led us to study the model of the game Connect-4, a simple board game for two players.

In Connect-4, players choose between two different sets of tiles,



Figure 5. OLED display in front of the box showing the race status.

usually red and yellow, and begin the game by taking turns placing their tiles in an upright board with 7 columns and 6 rows. The tiles drop into the lowest unoccupied space in the column, filling the board from bottom to top. The player's goal is to be the first to occupy four contiguous spaces with his tiles either vertically, horizontally or diagonally, and thus win the game.

To easily perform column selection during the game, we used a DFRobot LCD keypad shield [8] through the Arduino Uno R3, adding 5 buttons that send information through the analog input pin A0 and a 2x16 LCD (liquid crystal display) to configure and display information about the current game. To simulate the board with individually addressable LEDs, we used an 8x8 LED matrix and reserved one row as a column selector, leaving a board area 8 columns wide by 7 rows high, larger than the original Connect-4 but still functional. You can see the materials described in the figures 6 and 7.



Figure 6. Arduino Uno mounted with the LCD Keypad Shield.

### A. Code development

Knowing the proposal and the materials to be used, we proceeded with the coding of the game itself. To avoid interference from hardware problems, an initial version of the Connect-4 was developed using only the C programming language, with the terminal as the interface for input and

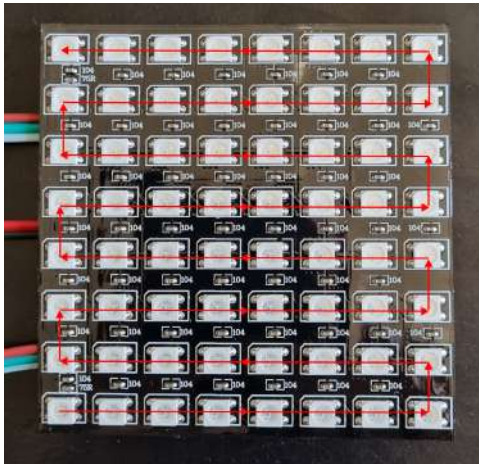


Figure 7. 8x8 LED Matrix, the red lines indicate the wiring sequence.

output of data. The prototype uses an 8x8 matrix with data type *char* filled with '\*' characters to represent the empty slots. The top row is the selection row and is filled with the 'X' character of player one or the '+' character of player two, who selects one of the columns, he can move cyclically through the selection row when the characters 'a' or 'd' are entered, moving the selected field to the left or right respectively. Then the player can enter the character 's' to confirm his choice and end his move, repeating the process between players until the game is won or all the squares are filled, resulting in a tie.

When a column is selected, the code checks all its rows and inserts the player's piece into the last empty slot. If the column is full, the operation becomes invalid and the move does not end until another non-full column is selected. After each move is successfully completed, four validations are performed, starting with the last selected slot, to check whether a win is established in one of the vertical, horizontal or both diagonal axes. If up to three more consecutive characters of the same type can be counted in the same axis, it means that the last placed character was able to connect four consecutive characters, making the last player the winner of the match.

Once the game was functional in the terminal implementation, a customization process began to adapt the inputs and outputs to the Arduino interfaces. The input function *scanf* was replaced with an analog read function that extracts data from pin A0 to determine which key was pressed on the LCD keypad shield. The same Adafruit Neopixel Arduino Library used to control the OLR LED strip was also used to control the LED matrix, making it behave like a sequential vector and requiring an arithmetic pointer calculation to match the positions of the

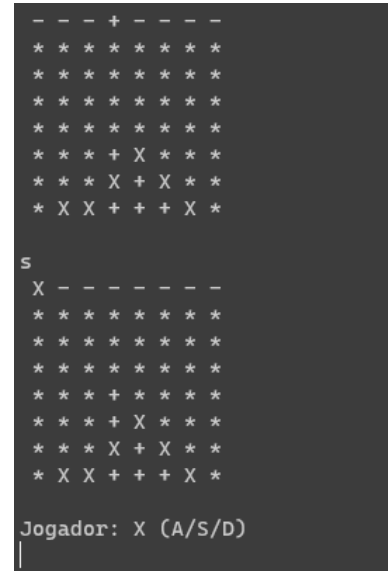


Figure 8. Last two turns of an ongoing Connect-4 match implemented with a terminal.

character matrix to the LED matrix, assigning each character as a color in the process. Due to the wiring sequence, which can be visualized in Figure 7, the LED matrix does not behave like a regular matrix, requiring an adjustment to the arithmetic pointer calculation to count each odd row as inverse and match the mapping to the zig-zag pattern of the LED matrix.

Once the Arduino project was working properly, a few different functions were added to the code to take advantage of the LCD display that was available according to suggestions collected from users. Once the Arduino starts, it prompts each player to choose their color from a list of pre-selected colors, flashing the entire matrix with the chosen color, then asks how many matches will be played. When the game starts, the LCD keeps track of the moves and shows which player's turn it is and how many matches are left, with a counter showing each player's wins. When all the matches are over, the LCD and the LED matrix indicate the winner, and the game goes back to the color-choice phase. The final implementation can be seen in the figure 9.

## VI. CONCLUSIONS AND FUTURE WORKS

As Gleitman notes in his dissertation [9] on the development of a sorting game that also uses WS2812 LED strips to display game information, it is more difficult to get people excited about Computer Science since its questions and results tend to be abstract. Therefore, projects that aim to present the subject in a tangible way can help make it fun and accessible to the general public, results we were also able to identify in our

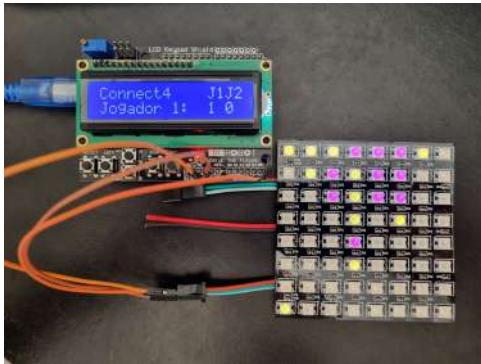


Figure 9. An ongoing Connect-4 match on the Arduino Uno displayed in an individually addressable LED matrix.

projects. To fulfil their purpose, the projects were taken to a total of 4 exhibitions and career fairs during the research process, where they served to capture the public's attention to the Computer Science course. Throughout our participation, we noticed an increase in public engagement, specially with the Open LED Race, as visitors would often return to compete against different friends, seen in the Figure 10. We also received many *feedbacks* that led to the improvement of the interactivity of both projects, as the OLED display in the Open LED Race and the color selector in the Connect-4.



Figure 10. Visitors racing each other in the Open LED Race during a careers fair.

These interactions were important not only because they brought more visitors at our booth, but also because they allowed us to engage in a conversation with them and raise their interest about technology and open-source projects, making the developed research successful in its objective of raising the effectiveness of our presence in exhibitions and careers fairs, allowing us to actively engage with the visitors. As future work, in order to engage the public's interest in

the project, it could be developed into a more educational approach where the participants could interact with the code and see the effects of their changes in real time, putting them in touch not only with the outcome of the project but also with the functioning of it. It is also essential to collect the results of our research, using surveys or other methods with prospective students, to obtain measurement data on the effectiveness of the project.

All of the code developed during the research is hosted in publicly available Github repositories [10] [11].

#### ACKNOWLEDGMENTS

To MEC-SESU, for funding this scientific research project through the Tutorial Education Program.

#### REFERENCES

- [1] L. Borsari. (2019) Open led race project. [Online]. Available: <https://openledrace.net/>
- [2] *WS2811 Datasheet*. [Online]. Available: <https://cdn-shop.adafruit.com/datasheets/WS2811.pdf>
- [3] *WS2812 Datasheet*. [Online]. Available: <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>
- [4] P. Burgess, *Adafruit NeoPixel Überguide*, Adafruit, 2013.
- [5] Arduino. (2018) What is arduino? [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>
- [6] *Arduino Documentation*. [Online]. Available: <https://docs.arduino.cc/>
- [7] *SSD1306 Datasheet*, 2008. [Online]. Available: <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>
- [8] *DFRobot LCD Keypad Shield DFR0009 Datasheet*. [Online]. Available: [https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0009\\_Web.pdf](https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0009_Web.pdf)
- [9] A. Gleitman, "The sorting game : a new way to teach computer science in outreach and museum settings," 2014.
- [10] M. Pereira. (2023) Arduinoconnect4. [Online]. Available: <https://github.com/Marlonfpereira/ArduinoConnect4>
- [11] ——. (2023) Openledrace. [Online]. Available: <https://github.com/Marlonfpereira/openLedRace>