Interactive Touch Me Not Game: Arduino UNO and Closed Circuit Principles

L Vinay Kumar Reddy
(Dept. of Computer Science.)
Christ University

Bangalore, Karnataka, India
vinaykumar.reddy@mca.christuniversity.in

***Abstract*—This research paper explores the innovative integration of electronics and gaming by implementing a closed circuit current passing mechanism within the popular game "Touch Me Not" using the Arduino Uno microcontroller. "Touch Me Not" is a classic electronic game that challenges players to complete a circuit without touching the wire in the middle with the copper handle. Traditionally, this game relies on passive components like wires and metal contacts. In this study, we present a novel approach by incorporating the Arduino Uno platform to enhance the gameplay experience and educational value**

***Keywords*—** Arduino UNO, circuit, component, pins

I. Introduction

In this paper, we present an implementation of the current passes through a closed circuit principle using Arduino UNO. Arduino UNO is a microcontroller board that can create interactive electronic devices.

Our implementation of the current passes through a closed circuit principle using Arduino UNO consists of a simple circuit with a switch, a buzzer, and an LED. When the switch is closed, the current flows through the circuit and the buzzer beeps and the LED turns on. When the switch is open, the circuit is broken and the buzzer stops beeping and the LED turns off.

The objectives are to implement the "current passes through a closed circuit" principle in a Touch Me Not game using an Arduino Uno, which can be an educational and engaging project. The objective of this project is to teach users about the basic principles of electrical circuits and how they can be used in a fun and interactive way.

II. Arduino UNO

A. Introduction to Arduino Uno

Arduino UNO is a microcontroller board developed by Arduino.cc and is based on the Atmega328 Microcontroller. The first Arduino project was started at Interaction Design Institute Ivrea in 2003 by David Cuartielles and Massimo Banzi with the intention of providing a cheap and flexible way for students and professionals to learn embedded programming. [1]

Arduino UNO is a very valuable addition to electronics that consists of a USB interface, 14 digital I/O pins(of which 6 Pins are used for PWM), six analog pins, and an Atmega328 microcontroller. It also supports three communication protocols: Serial, I2C, and SPI protocol [1] .The Fig. 1. shows how an Arduino UNO looks like with the a connecting USb port cable.

Arduino UNo has Multiple componenets and pins through which other devices are connected to it. The table 1 shows an overview of all the components on an Arduino UNO board.

- The software used for writing, compiling, and uploading code to Arduino boards is called Arduino IDE (Integrated Development Environment), which is free to download from Arduino Official Site.
- It has an operating voltage of 5V while the input voltage may vary from 7V to 12V.
- Arduino UNO has a maximum current rating of 40mA, so the load shouldn't exceed this current rating or you may harm the board.
- It comes with a crystal oscillator of 16MHz, which is its

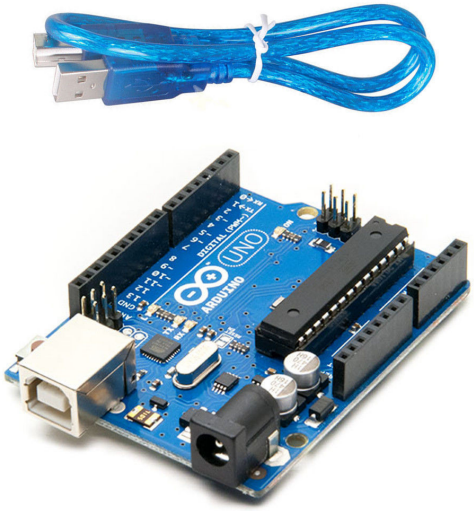

Fig. 1. Arduino UNO.

TABLE I
Arduino UNO Features and Technical Specs [2]

No.	Parameter Name	Parameter Value
1	Microcontroller	Atmega328
2	Crystal Oscillator	16MHz
3	Operating Voltage	5V
4	Input Voltage	5-12V
5	Digital I/O Pins	14 (D0 to D13)
6	Analog I/O Pins	6 (A0 to A5)
7	PWM Pins	6 (Pin # 3, 5, 6, 9, 10 and 11)
8	Power Pins	5V, 3.3V, Vin, GND
9	Communication	UART(1), SPI(1), I2C(1)
10	Flash Memory	32 KB (0.5KB is used by bootloader)
11	SRAM	2 KB
12	EEPROM	1 KB
13	ICSP Header	Yes
14	Power sources	DC Power Jack & USB Port

operating frequency.

- Arduino Uno pinout consists of 14 digital pins starting from D0 to D13.
- It also has six analog pins starting from A0 to A5.
- It also has 1 Reset Pin, which is used to reset the board programmatically. In order to reset the board, we need to make this pin LOW.
- It also has 6 Power Pins, which provide different voltage levels.
- Out of 14 digital pins, 6 pins are used for generating PWM pulses of 8-bit resolution. PWM pins in Arduino UNO are D3, D5, D6, D9, D10 and D11.

B. How to run the code

[3]

- Download Arduino IDE from Arduino's official site
- After installing IDE, open it by double clicking the shortcut icon or type IDE in the Windows search bar.
- After the IDE is open, it's time to set up Arduino hardware. Now connect the Arduino board using USB B cable through the COM port on the PC.
- Once the Arduino board is connected, it will automatically install the necessary drivers needed and will select the port at which it is connected.
- Connection is established between Arduino board and PC. It's time to upload an Arduino sketch. As an example, we will use LED Blink code to upload it to the Arduino board. Go to, File¿Examples¿Basics¿Blink:
- Blink code will open in a new window. It's time to compile the code. Click Verify button on top left window corner.
- Last step is to upload this sketch in Arduino. To do this click the right-pointing arrow known as the Upload button present on the right side of the verify button. Once the upload is finished, a Done Uploading message will appear in the output window.

III. Building the Hardware

A. Components required

- Arduino UNO.
- Sensor shield(SST technologies) with Buzzer, RGB light.
- Jumper wires.
- 2sq,6sq-mm Wires.

B. Defining and connecting the inputs

The 6sq-mm wires are used for the start, path, and finish points. These points are connected to different pins in the sensor shield of SSL technologies. The start point is covered with a jumper foil to connect it with PIN 6. The Path is connected with another wire, which is connected to the PIN 7. The finish point is covered with foil along with a jumper wire, which is connected to PIN 8.

The 2sq-mm wire is used for the handle, which acts as the ground that completes the circuit, letting the current pass through the circuit. The handle is connected to a jumper wire, which is connected to the ground pin in the sensor shield board.

IV. Explaining the Game

A. Start Point

The Handle is brought in contact with the start point to start the game. The Handle when touched at the start point, the LED turns green and indicates that the player is ready to start the game.

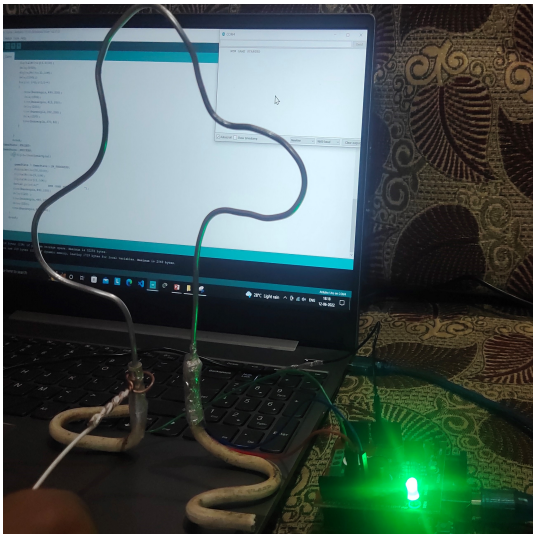

Fig. 2. Example Picture when Game is Started.

B. Fail Point

The Player, after the game has started, has to move the handle along the track without touching the way. When the player touches the track, the LED turns red and a buzz is heard from the buzzer meaning that the player has failed to reach the finish point without touching the track and has to restart the game by going back to the start point. In this state, even when the player reaches the final point, there is no celebration or any buzzer from the board as it is in the fail state now. To reset it, the player has to restart the game from the start point only.

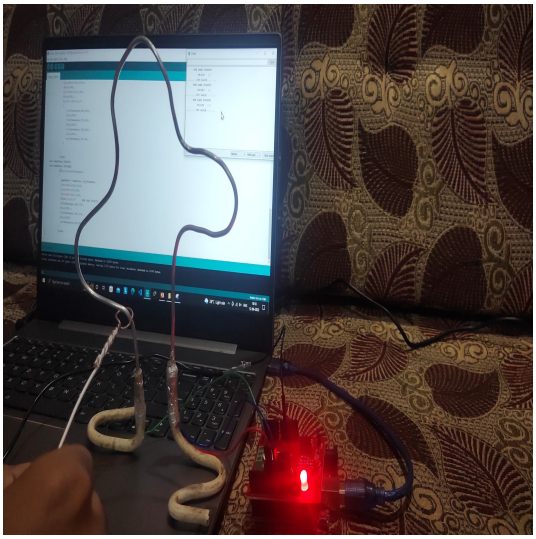

Fig. 3. Example Picture when Game is Lost.

C. Finish Point

The Player, after starting the game and without touching the track with the handle, reaches the final point and touches the final point with the handle. The LED turns blue, red, and green with multiple buzzes, indicating that the player has won the game.

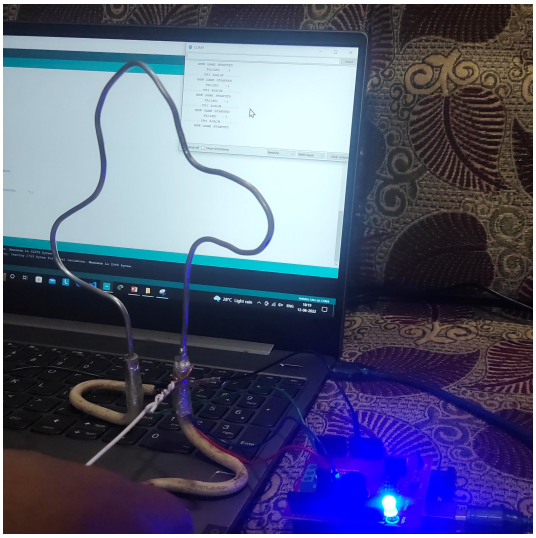

Fig. 4. Example Picture when Game is Won.

D. Code and Output

The first section of the code defines the pins for the buzzer, Start, Final, and Fail states and initializes the default state of the game to fail. it also defines the the type of the pin whether it is an input pin or an output pin.The other remaining section of the code in the void loop changes states from Start, Fail, and Finish according to the current passed through the defined pins and changes the buzzer sounds and the LED colors according to the state.

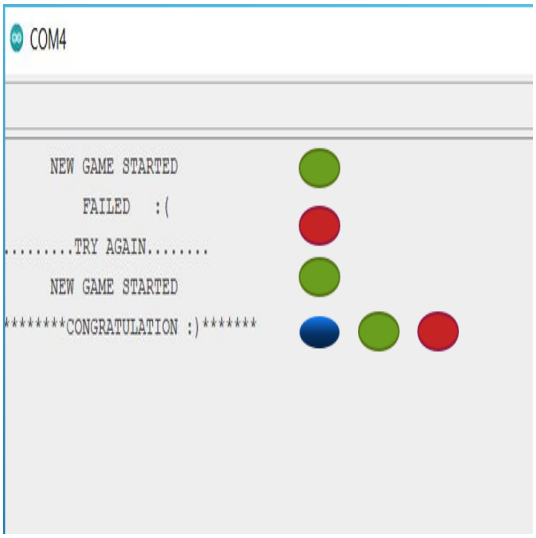

Fig. 5. Output on three different states

When the game is in the start state, the message ”NEW GAME STARTED” is displayed on the output screen with the LED on the board as green. When the state of the game changes the corresponding state message is displayed on the output screen. When the game is in a Fail state, the message ”FAIL :(” is displayed on the screen with the LED in red. So the player goes back and starts a new game. When the game is in Finish state, the message ”***CONGRATULATIONS :)***” is displayed on the output screen with the LED toggling between blue, red, and green colors.

```
Buzzer_Game §
const byte startpin=6;      //The " Start Zone " Which Players must touch to start the game each time

const byte failpin=7;       // Touching any part of the wire itself causes a failure

const byte endpin=8;        // The "Win Zone" At the end of the wire

const byte buzzerpin=3;     // A piezo buzzer chirps to signify sucess/failure

// Globals
// Keep track of the current states of the game.
enum GameState{FAILED,IN_PROGRESS,SUCCESS};
GameState gameState = GameState::FAILED;

void setup() {
  // put your setup code here, to run once:
pinMode(startpin,INPUT_PULLUP);
pinMode(failpin,INPUT_PULLUP);
pinMode(endpin,INPUT_PULLUP);
pinMode(buzzerpin,OUTPUT);
pinMode(11,OUTPUT);
pinMode(10,OUTPUT);
pinMode(9,OUTPUT);
Serial.begin(9600);
}
```

```
Buzzer_Game §
void loop() {
  // put your main code here, to run repeatedly:
switch(gameState)
{
  case GameState::IN_PROGRESS:
        if(!digitalRead(endpin))
        {
              gameState = GameState::SUCCESS;
              digitalWrite(9,HIGH);
              digitalWrite(10,LOW);
              digitalWrite(11,LOW);
              delay(1000);
              Serial.println("********CONGRATULATION :)*******");
              tone(buzzerpin,440,50);
              delay(60);
              tone(buzzerpin,587,250);
              for(int i=0;i<3;i++)
              {
                 digitalWrite(9,HIGH);
                 delay(100);
                 digitalWrite(9,LOW);
                 delay(100);

                 digitalWrite(10,HIGH);
                 delay(100);
                 digitalWrite(10,LOW);
                 delay(100);

                 digitalWrite(11,HIGH);
                 delay(100);
                 digitalWrite(11,LOW);
                 delay(100);
```

```
Buzzer_Game §
              tone(buzzerpin,440,50);
              delay(60);
              tone(buzzerpin,587,250);


            }
        }
        else if(!digitalRead(failpin))
            {
            gameState = GameState::FAILED;
            digitalWrite(11,HIGH);
            digitalWrite(9,LOW);
            digitalWrite(10,LOW);
            delay(100);
            Serial.println("          FAILED  :(");
            Serial.println(".....TRY AGAIN.....");
            tone(buzzerpin,440,200);
            delay(200);
            tone(buzzerpin,415,200);
            delay(200);
            tone(buzzerpin,392,200);
            delay(200);
            tone(buzzerpin,370,40);
            for(int i=0;i<5;i++)
            {
               digitalWrite(11,HIGH);
               delay(100);
               digitalWrite(11,LOW);
               delay(100);}
```

Fig. 6. Code used for the Arduino board

V. Conclusion

In conclusion, this research paper highlights the development of an interactive "Touch Me Not" game using Arduino UNO as an educational tool for teaching closed circuit principles. The game offers an engaging and effective learning experience by combining theoretical concepts with hands-on experimentation, making it a valuable addition to STEM education. It promotes active engagement, accessibility, interdisciplinary learning, scalability across different age groups, and the demonstration of real-world applications. This innovative approach has the potential to reshape how closed-circuit principles are taught, preparing students for the digital age and fostering a lasting passion for learning in the fields of science and engineering.

References

[1] Wikipedia contributors, (2023, September), Sbmeirow, In Wikipedia, The Free Encyclopedia, Retrieved on 20th September 2023, from "https://en.wikipedia.org/w/index.php?title=Arduino_Uno&action=history"

[2] components101 contributors, In components101 website, retrieved on 23rd September 2023, from "https://components101.com/microcontrollers/arduino-uno"

[3] Community Member in Linuxhunt, Kashif, retrived on 23rd September 2023 from "https://linuxhint.com/run-arduino-code/"

[4] Baji, Sayyed Ayesha, Mangalagiri Lavanya, Sirasani Jhansi, Naladimmu Dhanalakshmi Lakshmi, and P. Narayana Swami. "Home Automation using IOT Application."

[5] Hadwan, H. Hamid, and Y. P. Reddy. "Smart home control by using Raspberry Pi and arduino uno." Int. J. Adv. Res. Comput. Commun. Eng 5.4 (2016): 283-288.

[6] Shejwal, C. K., Mr VV Mhaske, Mr SC Thorat, and Miss JD Chavhan. "ARDUINO-BASED ACCIDENT PREVENTION SYSTEM USING EYE BLINK SENSOR."

[7] Rabrenović, Radovan, Marko Tadić, and Radovan Stojanović. "ARDUINO BASED SMART FARMING FOR THERAPEUTICAL MICRO PLANTS."

[8] Lusterio, Vincent, Robert Christian Montecer, and Neil Balba. "Multiple Laser Alarm System using Arduino Uno." Lyceum of the Philippines–Journal of Engineering and Computer Studies 4,no. 3 (2020): 1-1.

[9] Omar Imaeeg, Massoud A., Ismail M. Elkafrawi, Tarek RS Al Aga, Hussam MI Alkafrawi, and Yaser SA Shaheen. "Arduino Mega Based Smart Traffic Control System." Asian Journal of Advanced Research and Reports (2021): 43-52.

[10] Chinnasami Sivaji, M. Ramachandran, et al. "Application of Arduino Devices in various IOT Application." Renewable and Nonrenewable Energy 1 (2022): 39-4.

[11] Kaswan, Kuldeep Singh, Santar Pal Singh, and Shrddha Sagar. "Role of Arduino in real world applications." Int. J. Sci. Technol. Res 9.1 (2020): 1113-1116.

[12] Taufiq, A. J., I. H. Kurniawan, and T. A. Y. Nugraha. "Analysis of Arduino Uno Application on Control System Based on Industrial Scale." IOP Conference Series: Materials Science and Engineering. Vol. 771. No. 1. IOP Publishing, 2020.

[13] Ismailov, Alisher Shakirovich, and Zafar Botirovich Jo'Rayev. "Study of arduino microcontroller board." Science and Education 3.3 (2022): 172-179.

[14] Ismailov, Alisher Shakirovich, and Zafar Botirovich Jo'Rayev. "Study of arduino microcontroller board." Science and Education 3.3 (2022): 172-179.

[15] RAO, Y., and G. RAVINDRANATH. "Implementation of DTC-SVM (Sample Reference Phase Voltages) of Induction Motor Using Arduino UNO." Journal of Active & Passive Electronic Devices 16.2 (2021).

[16] "https://www.theengineeringprojects.com/2018/40/introduction-to-arduino-uno.html"