SMART TRAFFIC LIGHT WITH ARDUINO SIMULATION IN TINKERCAD ENVIRONMENT

Lecturer PhD Eng., **Gheorghe GÎLCĂ,** "Constantin Brâncuși" University from Tîrgu-Jiu**,** gheorghe.gilca@yahoo.com, ROMANIA
Assistant professor PhD Stud. Eng., **Lupulescu (Machidon) Alina-Elena,** "Constantin Brâncuși" University from Tîrgu-Jiu**,** alinalupulescu@yahoo.com, ROMANIA

Abstract: *This paper presents the design and simulation of a smart traffic light using the Arduino UNO platform in the Tinkercad virtual environment. The project implements an adaptive regulation system of the duration of the light signals depending on the traffic density, simulated by means of two potentiometers. The goal is to optimize the waiting time at intersections and streamline road traffic.*

Keywords: smart traffic light, Arduino UNO, tinkercad virtual environment

1. INTRODUCTION

In the context of the rapid development of urban centers and the exponential increase in the number of vehicles, road traffic management has become one of the most pressing challenges of the 21st century. Modern cities are faced daily with traffic jams, delays, increased pollution and a decrease in the quality of life of citizens, mainly caused by the inefficiency of transport infrastructure and traditional road signaling systems [1].

Conventional traffic lights, based on preset times, are not able to respond dynamically to fluctuating traffic volumes, which leads to significant losses of time and fuel. For example, a vehicle can wait at a red traffic light in the absence of other traffic participants from adjacent directions, which denotes an inefficient use of signaling time. In this context, smart traffic lights come as a modern and scalable solution for optimizing road flow and reducing the negative impact of traffic on the environment [2].

Intelligent traffic light systems are capable of monitoring traffic in real time and adapting signaling based on data collected from various sensors: proximity sensors, video cameras, ultrasonic sensors, magnetic sensors embedded in the roadway or even communication modules with intelligent vehicles [3]. This data is processed by microcontrollers or embedded systems, which determine the optimal signaling times, so as to reduce waiting times, fuel consumption and, implicitly, polluting gas emissions [4].

In this context, Arduino represents an ideal platform for the development and prototyping of these intelligent systems, due to its low cost, flexibility and accessibility. Arduino is an open-source development platform, based on microcontrollers (such as ATmega328), which can be easily programmed using the C/C++ language and the Arduino IDE development environment [5]. Arduino projects can include multiple components, such as LEDs, sensors, LCD displays, wireless communication modules, and more, making this platform a popular choice in both academia and industry for rapid prototyping and functional applications [6].

A typical Arduino-based smart traffic light system involves several essential components: a microcontroller that coordinates the operation of the system, vehicle detection sensors for each lane or direction of travel, RGB LEDs to represent the traffic light, communication modules (such as ESP8266 for Wi-Fi or NRF24L01 for RF), and, optionally, a display to count the time remaining until the traffic light color changes [7]. Through programmed code, the system can collect data from the sensors and decide, in real time, how long a certain color should remain active based on the existing traffic.

Moreover, modern smart traffic light systems can be integrated into wider IoT (Internet of Things) networks by connecting them to centralized monitoring platforms. Thus, data can be collected from multiple intersections and traffic lights can be coordinated to form so-called "green waves", which allow the smooth transit of vehicles on congested routes, depending on the time intervals and predominant travel directions [8].

The integration of these technologies into urban infrastructure brings multiple benefits:

- Reducing waiting times at traffic lights;
- Reducing fuel consumption and polluting emissions;
- Increasing road safety by reducing conflicts at intersections;
- Prioritizing emergency vehicles (ambulances, firefighters) by automatically recognizing them using RFID or radio signals [9].

Also, smart traffic light systems can be extended to include pedestrian signaling, pedestrian detection at crossings, integration with mobile applications or audible signaling for people with disabilities, thus contributing to the development of smart and accessible cities [10].

In conclusion, Arduino-based smart traffic lights represent a feasible and effective solution for modern urban traffic management. They combine the advantages of embedded technology with the real needs of transport infrastructure and provide an excellent platform for research, development and education in the field of smart systems.

This project aims to explore the architecture, operation and practical implementation of such a system, demonstrating its viability in an experimental context.

2. MATHERIALS AND METHODS

2.1. COMPONENTS USED

The electronic components used in the project are the following:

- ✓ Arduino UNO;
- ✓ 6 LEDs (red, yellow, green for two directions);
- ✓ 2 potentiometers (traffic A and traffic B);
- ✓ Breadboard; 220 ohm resistors;
- ✓ Connection wires;

Arduino UNO is an open-source development board based on the ATmega328P microcontroller. It features 14 digital pins (of which 6 can be used as PWM), 6 analog inputs, a 16 MHz crystal, a USB connection, a power jack, and a reset button. Arduino UNO is ideal for educational projects and rapid prototyping due to the simplicity of programming in the Arduino C/C++ language and the rich ecosystem of resources. In table 1 is presented a comparison with other Arduino boards:

Table 1. Presentation of characteristics of arduino types

Features	**Arduino UNO**	**Arduino Nano**	**Arduino Mega 2560**
Microcontroller	ATmega328P	ATmega328P	ATmega2560
Digital I/O Pins	14	14	54
Analog Input Pins	6	8	16
Flash Memory	32 KB	32 KB	256 KB
USB Port	USB Type B	Mini USB	USB Type B
Dimensions	68.6 x 53.4 mm	45 x 18 mm	101.5 x 53.3 mm

2.2. CONNECTION DIAGRAM

Figure 1 shows the connection diagram between the electronic components.

- ✓ The LEDs are connected to the digital pins D2-D7;
- ✓ The potentiometers are connected to the analog pins A0 and A1.

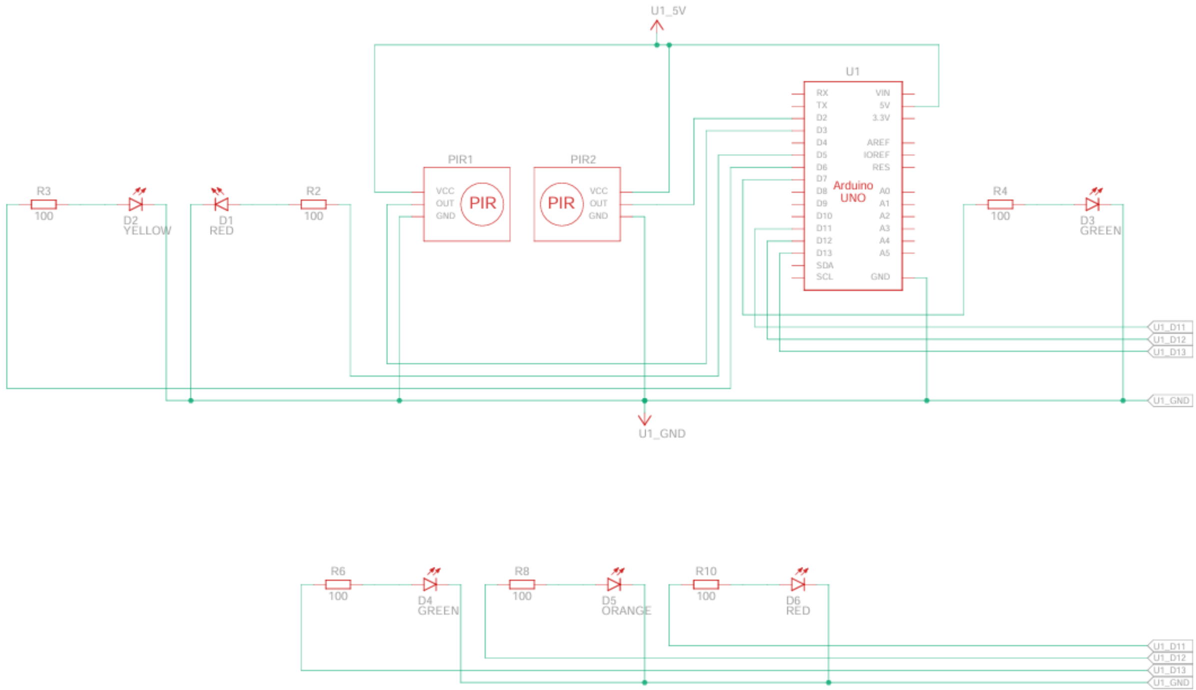

Fig. 1. The connection diagram

2.3. THE CONTROL ALGORITHM

The algorithm reads the values from the potentiometers and determines, based on the difference between them, the duration of the green signal for each direction. Thus:

- ✓ If traffic A >> traffic B => green A 7s, green B 3s;
- ✓ If traffic B >> traffic A => green A 3s, green B 7s;
- ✓ If traffic is balanced => 5s for both directions.

The Arduino program was developed in the Tinkercad environment and is presented below. It implements the described logic and controls the LEDs based on the received analog values.

```
// Pin sensor Pir
int sh = 2;
int sv = 3;
// Valori de la senzori
int valh = 0;
int valv = 0;
// Pin semaphore vertical
int rv = 5;
int av = 6;
int vv = 7;
// Pin semaphore horizontal
int rh = 11;
int ah = 12;
int vh = 13;
void setup()
{
      for (int i = 5; i <= 13; i++) {
        pinMode(i, OUTPUT);
  }
  pinMode(sh, INPUT);
  pinMode(sv, INPUT);
  Serial.begin(9600);
}
void loop()
{
  valh = digitalRead(sh);
  valv = digitalRead(sv);
  Serial.print("Horizontal: ");
  Serial.println(valh);
  Serial.print("Vertical: ");
  Serial.println(valv);
  Serial.println();
  int th1 = 3000;
  int th2 = 400;
  int tv1 = 3000;
  int tv2 = 400;
```

```
 if (valh == HIGH && valv == LOW) {
      Serial.println("Aumentar
Horizontal");
   th1 = th1 * 2;
   th2 = th2 * 2;
 } else if (valh == LOW && valv ==
HIGH) {
   Serial.println("Aumentar
Vertical");
   tv1 = tv1 * 2;
      tv2 = tv2 * 2;
 }
 semaphoreHorizontal(th1, th2);
 semaphoreVertical(tv1, tv2);
}
void semaphoreHorizontal(int t1, int
t2) {
 digitalWrite(rv, HIGH);
      digitalWrite(vh, HIGH);
 delay(t1);
 digitalWrite(vh, LOW);
 digitalWrite(ah, HIGH);
 delay(t2);
 digitalWrite(rv, LOW);
 digitalWrite(ah, LOW);
}
void semaphoreVertical(int t1, int
t2) {
 digitalWrite(rh, HIGH);
 digitalWrite(vv, HIGH);
 delay(t1);
 digitalWrite(vv, LOW);
 digitalWrite(av, HIGH);
 delay(t2);
 digitalWrite(rh, LOW);
 digitalWrite(av, LOW);
```

The project was built and tested in Tinkercad Circuits. The platform allows for the simulation of electronic circuits and testing of code in real time. The traffic light behavior was correct, with changes in the potentiometers being directly reflected in the duration of the green signal.p

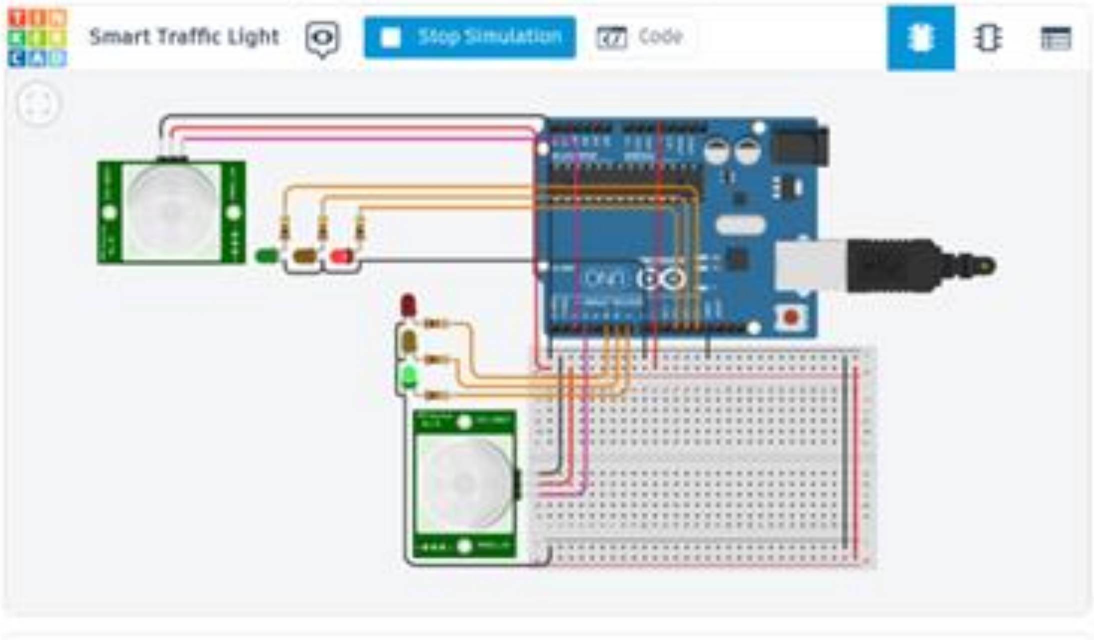

Fig. 2. Horizontal semaphore red light on - vertical semaphore green light

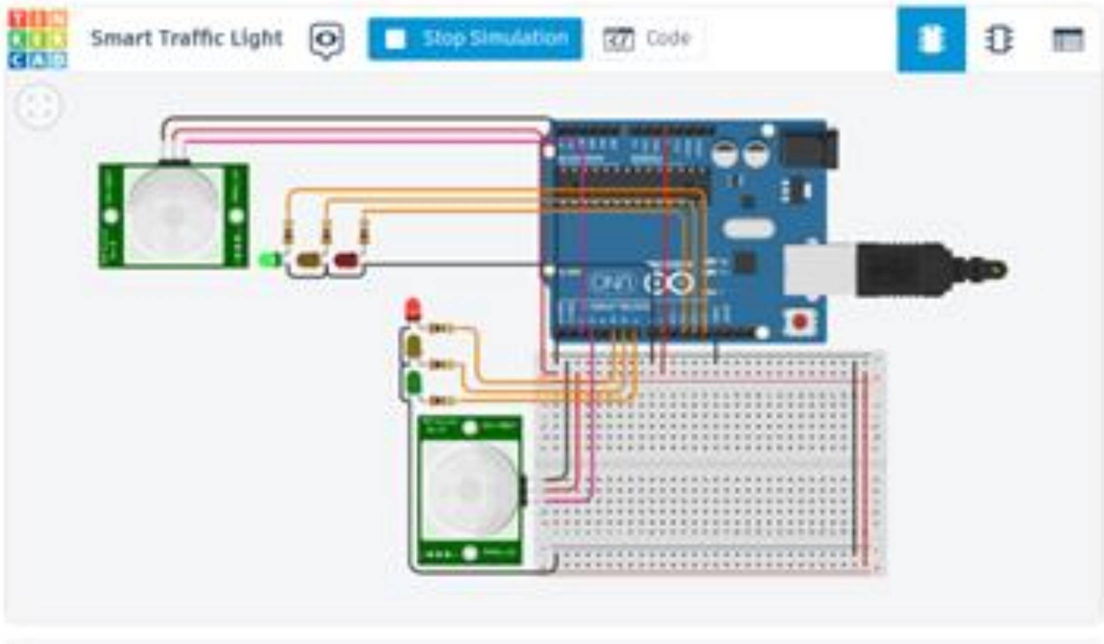

Fig. 3. Horizontal semaphore green light on – vertical semaphore red light

3. CONCLUSIONS

The project demonstrates the feasibility and efficiency of a simple smart traffic light system. The use of Arduino and Tinkercad provides an accessible platform for learning and testing. The model can be extended by including real sensors (IR, ultrasonic) or integrating into an IoT system.

4. REFERENCES

[1] **Chowdhury, M., & Sadek, A.** (2003). *Fundamentals of Intelligent Transportation Systems Planning*. Artech House.

[2] **Baskar, L. D., et al.** (2011). *"Traffic management for automated highway systems using model-based predictive control."* IEEE Transactions on Intelligent Transportation Systems, 13(2), 838–847.

[3] **Asadi, H., & Vaziri, M.** (2015). *"Design and Implementation of an Intelligent Traffic Control System Using Wireless Sensor Networks."* Journal of Advanced Transportation, 49(4), 427–438.

[4] **Zhang, Y., et al.** (2017). *"A survey on smart traffic light systems."* Procedia Computer Science, 109, 987–992.

[5] **Banzi, M., & Shiloh, M.** (2014). *Getting Started with Arduino* (3rd ed.). Maker Media.

[6] **Singh, A., & Kumar, R.** (2021). *"Design of a Smart Traffic Light Control System Using Arduino UNO and IR Sensors."* International Journal of Emerging Technology and Advanced Engineering, 11(6), 55–60.

[7] **Rahman, M. A., et al.** (2018). *"IoT-based Smart Traffic Signal Control System Using Arduino."* International Journal of Computer Applications, 182(29), 1–5.

[8] **Patel, D., & Shah, R.** (2019). *"RFID Based Traffic Signal System for Emergency Vehicles."* International Journal of Innovative Research in Computer and Communication Engineering, Vol. 7, Issue 5.

[9] **Kumar, S., & Sahu, A.** (2022). *"AI-powered Traffic Management Using Embedded Systems."* Journal of Embedded Systems and Applications, Vol. 10, No. 1, 14–22.

[10] **Bhargava, N., & Gupta, M.** (2020). *"Implementation of Adaptive Traffic Signal Using Arduino and Ultrasonic Sensors."* International Journal of Scientific Research in Engineering and Management (IJSREM), Vol. 4, Issue 9.