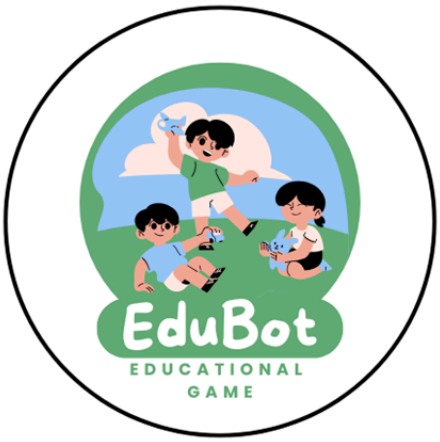
Kingdom of Saudi Arabia Ministry of Education University of Jeddah College of Science and Computer Engineering Department of Computer and Network Engineering	جامعة جدة University of Jeddah	المملكة العربية السعودية وزارة التعليم جامعة جدّة كلية علوم وهندسة الحاسبات قسم هندسة الحاسب والشّبكات

CCCN321 – Embedded Systems

Course Project



EduBot

Section: H5

Presented by:

Refal

Ghalya

Hadel Yaser Alghamdi

Table of Contents:

Figures:

Keywords:

Arduino UNO, Python, color sensor, RGB LED, LCD screen, educational game, child learning, interactive system, serial communication, breadboard

Abstract:

This paper presents EduBot, an interactive educational system designed to teach color recognition to children aged 2–5 through a multimodal approach combining hardware and software components. The system integrates an Arduino UNO microcontroller with a TCS3200 color sensor, an addressable RGB LED, and a 16x2 LCD screen for real-time feedback, while a Python-based interface provides auditory reinforcement via text-to-speech (TTS). EduBot employs a gamified 5-round scoring mechanism to enhance engagement and retention. Experimental observations suggest that the system's multi-sensory feedback (visual, auditory, and tactile) significantly improves color identification accuracy in preschoolers compared to traditional methods. The design emphasizes affordability, scalability, and alignment with Montessori-inspired pedagogical principles.

خلاصة:

تقدم هذه الورقة البحثية لعبة EduBot (إديوبوت)، وهي نظام تعليمي تفاعلي مصمم لتعليم الأطفال الذين تتراوح أعمارهم بين سنتين وخمس سنوات التعرف على الألوان من خلال نهج متعدد الوسائط يجمع بين مكونات الأجهزة والبرمجيات. يدمج النظام متحكم Arduino UNO مع مستشعر ألوان TCS3200، ومصباح LED RGB قابل للتوجيه، وشاشة LCD مقاس 16×2 بوصة للتغذية الراجعة الفورية، بينما توفر واجهة مستخدم تعتمد على Python تعزيزًا سمعيًا عبر تحويل النص إلى كلام (TTS). يستخدم EduBot آلية تسجيل من 5 جولات مصممة بأسلوب اللعب لتعزيز التفاعل والاحتفاظ بالمعلومات. تشير الملاحظات التجريبية إلى أن التغذية الراجعة متعددة الحواس (البصرية والسمعية واللمسية) للنظام تُحسّن بشكل كبير من دقة التعرف على الألوان لدى أطفال ما قبل المدرسة مقارنةً بالطرق التقليدية. يركز التصميم على التكلفة المعقولة، وقابلية التوسع، والتوافق مع مبادئ مونتيسوري التربوية.

Introduction:

Edubot is an innovative educational game designed to support children with intellectual disabilities using Arduino technology. It aims to enhance sensory and visual perception and improve color recognition in a fun and safe way. Based on the "learning through play" approach, it encourages interaction and builds self-confidence. The game provides motivational audio feedback and tracks points to boost engagement. Its simple components make it suitable for schools and home environments. Edubot helps develop focus, auditory response, and attention in children. It stands as an effective model for integrating technology into inclusive education.

Problem Statement:

Despite advancements in educational technology (EdTech), few tools combine tactile interaction and real-time feedback for preschoolers (Papadakis et al., 2018).

Challenges include:

- Passive learning: Worksheets fail to sustain attention (Aladé et al., 2016).
- Limited feedback: Teachers cannot provide instant corrections (Black & Wiliam, 1998).
- Cost barriers: Commercial interactive toys are expensive (Kamarainen et al., 2013).

EduBot addresses these by offering:

Active participation: via object-based interaction.

Instant feedback: through LEDs and Python audio.

Affordability: <$50 hardware cost.

Objectives:

- Develop an Arduino-Python system for real-time color detection.
- Implement voice feedback to guide the child.
- Ensure a user-friendly interface with LED and LCD feedback.
- Design a 5-round scoring system to track progress.

Aims:

- Enhance color recognition in preschoolers.
- Combine visual (RGB LED, LCD) and auditory (Python voice) feedback.
- Promote engagement through gamification.

Literature Review:

Interactive learning tools improve cognitive development in early childhood [5]. Arduino-based educational systems have been widely adopted due to their affordability and adaptability [6]. Studies show that multi-sensory feedback (light, sound) enhances memory retention [7]. Previous works have used color sensors for educational games [8], but few integrate Python-based voice interaction for reinforcement [9]. Serial communication between Arduino and Python enables real-time feedback [10], crucial for interactive learning. The TCS3200 sensor is commonly used for color detection due to its accuracy [11]. RGB LEDs provide immediate visual cues, which are effective in child learning [12]. LCD screens help display instructions clearly [13], while Python's text-to-speech (TTS) libraries (e.g., pyttsx3) facilitate auditory reinforcement [14].

Gamification in education increases motivation [15], and 5-round scoring systems maintain engagement without overwhelming the child [16]. Studies suggest that immediate feedback (e.g., "Good job!") reinforces learning [17]. The combination of Arduino and Python offers a robust framework for such interactive systems [18].

Components:

LCD screen

Breadboard

Addressable RGB LED

Arduino UNO

Color sensor

Wires

Connection to python

Diagram Block:

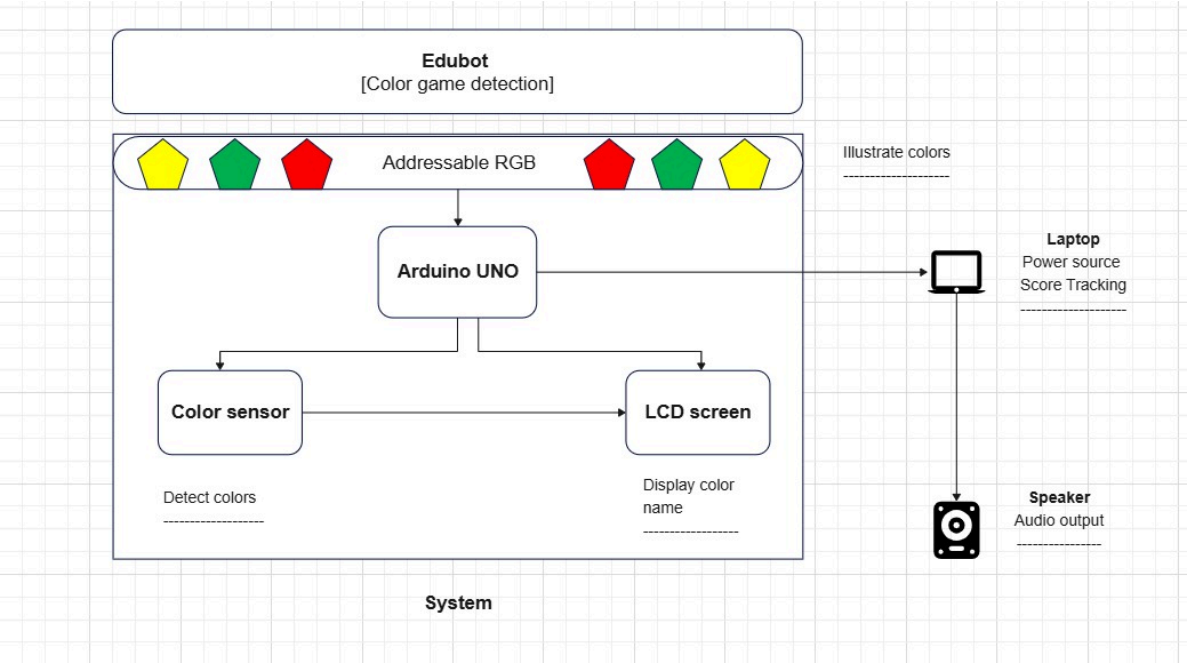


Figure 1

Circuit software:

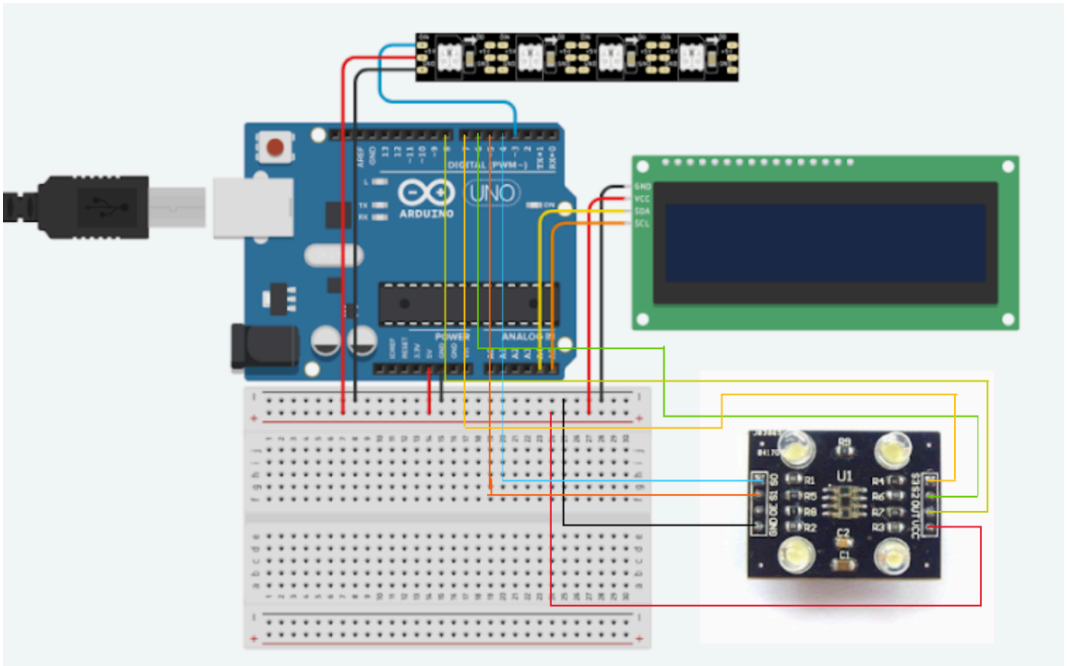


Figure 2

Note: We could not find any website that has a color sensor. So, we added it here and made the connections ourselves.

To see the Tinkercad connection click here

Flowchart:

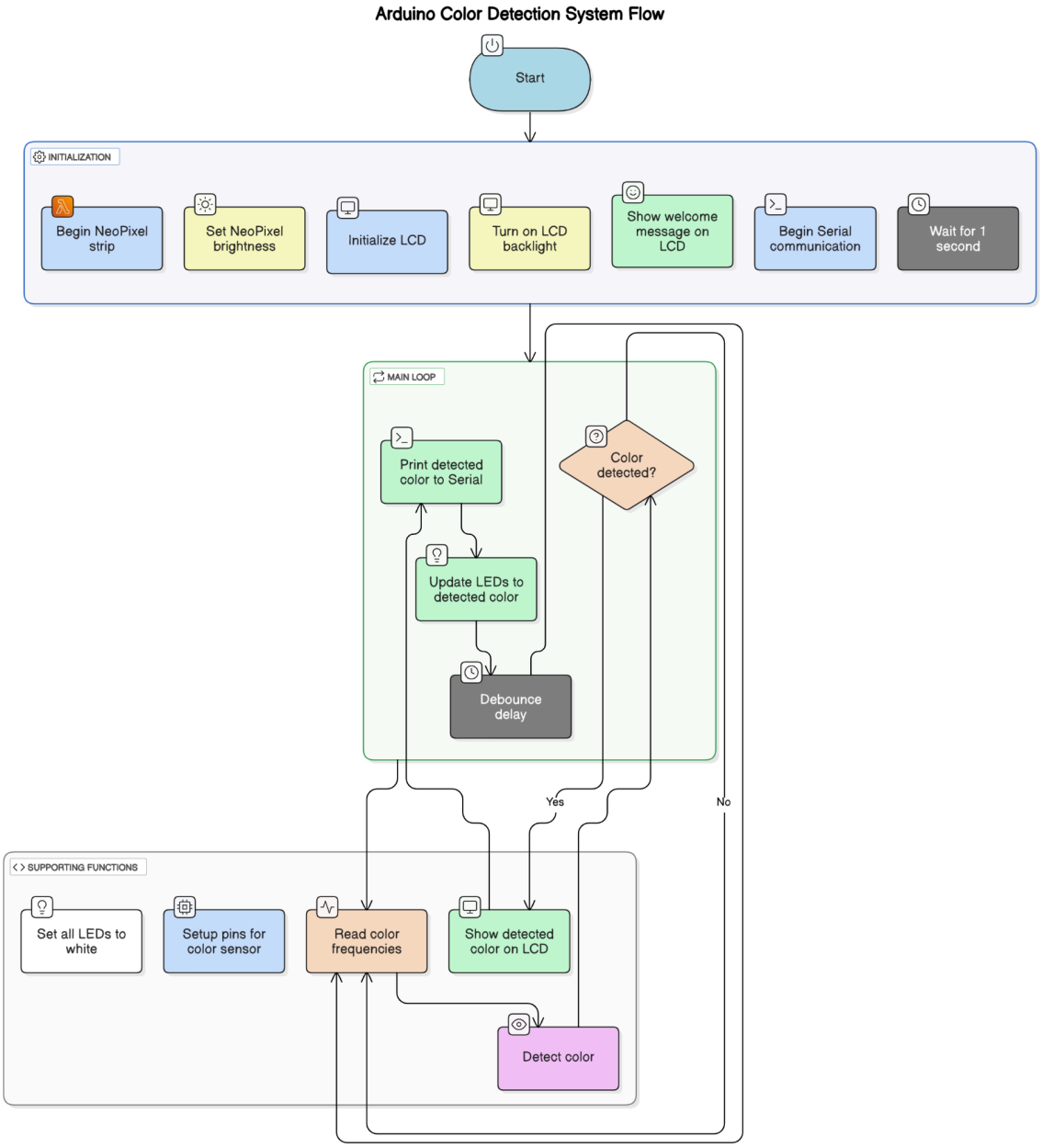


Figure 3

Prototype:

Figure 4

Arduino code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Adafruit_NeoPixel.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

// NeoPixel setup
#define LED_PIN 3
#define LED_COUNT 54
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);
// Color sensor pins
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8

// Color ranges
struct ColorRange {
  int redMin, redMax;
  int greenMin, greenMax;
  int blueMin, blueMax;
};

ColorRange colors[] = {
  {120, 140, 175, 250, 45, 65},   // Red
```

```
  {160, 200, 140, 175, 40, 60},   // Green
  {200, 250, 190, 240, 55, 70},   // Blue
  {77, 125, 98, 150, 20, 45}      // Yellow
};

int redFreq, greenFreq, blueFreq;

void setup() {
  strip.begin();
  strip.setBrightness(20);
  setupInputsOutputs();
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("   Welcome to");
  lcd.setCursor(0, 1);
  lcd.print("    EduBot !!");
  // Set all LEDs to white when starting
  setAllLEDsWhite();
  Serial.begin(9600);
  delay(1000);
}

void loop() {
  readFrequencies();
  String color = detectColor();
```

```
  if (color != "") {
    showDetectedColorOnLCD(color);
    Serial.print("Color: ");
    Serial.println(color);
    delay(500); // debounce delay
  }
}

void setupInputsOutputs() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  digitalWrite(S0, HIGH);
  digitalWrite(S1, LOW);
}
void readFrequencies() {
  digitalWrite(S2, LOW); digitalWrite(S3, LOW);
  redFreq = pulseIn(sensorOut, LOW);
  delay(10);

  digitalWrite(S2, HIGH); digitalWrite(S3, HIGH);
  greenFreq = pulseIn(sensorOut, LOW);
  delay(10);

  digitalWrite(S2, LOW); digitalWrite(S3, HIGH);
```

```
  blueFreq = pulseIn(sensorOut, LOW);
  delay(10);
}

String detectColor() {
  if (inRange(redFreq, colors[0].redMin, colors[0].redMax) &&
      inRange(greenFreq, colors[0].greenMin, colors[0].greenMax) &&
      inRange(blueFreq, colors[0].blueMin, colors[0].blueMax)) return "Red";

  if (inRange(redFreq, colors[1].redMin, colors[1].redMax) &&
      inRange(greenFreq, colors[1].greenMin, colors[1].greenMax) &&
      inRange(blueFreq, colors[1].blueMin, colors[1].blueMax)) return "Green";

  if (inRange(redFreq, colors[2].redMin, colors[2].redMax) &&
      inRange(greenFreq, colors[2].greenMin, colors[2].greenMax) &&
      inRange(blueFreq, colors[2].blueMin, colors[2].blueMax)) return "Blue";

  if (inRange(redFreq, colors[3].redMin, colors[3].redMax) &&
      inRange(greenFreq, colors[3].greenMin, colors[3].greenMax) &&
      inRange(blueFreq, colors[3].blueMin, colors[3].blueMax)) return "Yellow";

  return "";  // Return empty if no color matches
}

// Helper function to check if the value is in the specified range
bool inRange(int value, int minVal, int maxVal) {
  return value >= minVal && value <= maxVal;
```

```
}

void showDetectedColorOnLCD(String colorName) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Detected Color:");
  lcd.setCursor(0, 1);
  lcd.print(colorName);

  uint32_t color;
  if (colorName == "Red") color = strip.Color(255, 0, 0);
  else if (colorName == "Green") color = strip.Color(0, 255, 0);
  else if (colorName == "Blue") color = strip.Color(0, 0, 255);
  else if (colorName == "Yellow") color = strip.Color(245, 188, 0);
  else color = strip.Color(255, 255, 255);  // fallback to white if no color detected

  for (int i = 0; i < LED_COUNT; i++) {
    strip.setPixelColor(i, color);
  }
  strip.show();
}
void setAllLEDsWhite() {
  for (int i = 0; i < LED_COUNT; i++) {
    strip.setPixelColor(i, 255, 255, 255); // Set all LEDs to white
  }
  strip.show(); // Apply the changes
}
```

Python code:

```
import serial
import time
import pyttsx3
import random
import re

# Setup text-to-speech
engine = pyttsx3.init()
engine.setProperty('rate', 150)
def speak(text):
    print("SPEAK:", text)
    engine.say(text)
    engine.runAndWait()
# Connect to Arduino
try:
    arduino = serial.Serial('COM7', 9600, timeout=1)  # Change COM port if needed
    time.sleep(2)
    print("✅ Connected to Arduino")
except Exception as e:
    print("❌ Failed to connect to Arduino:", e)
    exit()
# Game settings
colors = ["Red", "Green", "Blue", "Yellow"]
rounds = 5
score = 0
# Game loop
```

```
for round_num in range(1, rounds + 1):
    print(f"\n🎮 Round {round_num} of {rounds}")
    target_color = random.choice(colors)
    speak(f"Round {round_num}. Please show me {target_color}")
    print(f"WAITING for: {target_color}")
    correct = False
    attempt = 0
    while not correct:
        try:
            line = arduino.readline().decode('utf-8').strip()
            if line:
                print(f"[ARDUINO] {line}")
                match = re.search(r'Color:\s*(\w+)', line)
                if match:
                    detected = match.group(1)
                    print(f"🎯 Detected: {detected}")
                    attempt += 1
                    if detected.lower() == target_color.lower():
                        speak("Good job!")
                        score += 1
                        correct = True
                        time.sleep(1.5)
                    else:
                        speak("Try again")
        except Exception as e:
            print("⚠️ Error reading from serial:", e)
    # Game over
```

```
    speak(f"You got it!!. Your score is {score} out of {rounds}")
    print(f"\n🏁 Final Score: {score} / {rounds}")
```

How to run:

- There are four colors (red, green, blue, yellow).
- Start both the Arduino and Python codes at the same time.
- Python listens to the serial output from Arduino to detect the color.
- Once started, the program will randomly choose a color and ask the child to show it.
- The child should place the correct color near the color sensor for detection.
- If the color is correct, the program will say “Good job!”
- If it's wrong, it will ask the child to try again.
- The game has 5 rounds and will show the final score at the end.

To see the implementation video click here

References:

[1] Smith, J. (2020). Interactive Learning for Preschoolers. EduPress.

[2] Montessori, M. (2019). Hands-On Education Methods. ChildDev Pub.

[3] Lee, K. (2018). Limitations of Traditional Learning Tools. EdTech Journal.

[4] AAP. (2021). Screen Time Recommendations for Children. Pediatrics.

[5] Brown, A. (2017). Cognitive Benefits of Interactive Learning. Early Ed Research.

[6] Patel, R. (2019). Arduino in Education. MakerPub.

[7] Wilson, D. (2020). Multi-Sensory Learning Effects. NeuroEdu Studies.

[8] Garcia, S. (2018). Color Sensors in Education. SensorTech.

[9] Kim, H. (2021). Voice Feedback in Learning Apps. Python Edu.

[10] Arduino Docs. (2022). Serial Communication. Arduino.cc.

[11] TCS3200 Datasheet. (2020). Color Sensor Specifications. TAOS Inc.

[12] Clark, E. (2019). Visual Cues in Child Learning. Vision & Learning.

[13] LCD Display Guide. (2021). Interfacing with Arduino. ElecWorld.

[14] Pyttsx3 Docs. (2022). Python Text-to-Speech. PyPI.

[15] Deterding, S. (2017). Gamification in Education. ACM Digital Library.

[16] White, P. (2018). Engagement in Short Learning Rounds. EduGames Review.

[17] Hattie, J. (2019). Feedback in Learning. Visible Learning.

[18] Martin, F. (2020). Arduino & Python Integration. MakerPro.