REPORT 9: IMAGE AND VIDEO

GROUP 4

MCTA 3203

SEMESTER 2 2023/2024

MECHATRONICS SYSTEM INTEGRATION

DATE OF SUBMISSION:
21[TH] MAY 2024

	NAME	MATRIC NUMBER
1.	NUR SHAHEERA BINTI KAMIS	2214040
2.	NORAMIRA HUSNA BT NORKHAIRANI	2214496
3.	MUHAMMAD EIQBAL BIN HASBOLLAH	2216911

TABLE OF CONTENT

ABSTRACTS

This project intends to construct a color-detecting system utilising an Arduino microcontroller, Python programming, and a color sensor or USB camera. The system's major purpose is to properly recognise and discriminate colors under varied situations. The project comprises multiple stages: hardware assembly, Arduino programming, Python scripting, and extensive examination of the system's color detecting accuracy and performance. The hardware setup requires combining a color sensor or USB camera with the Arduino. The software development step comprises programming the Arduino to communicate with the sensor or camera and building Python scripts for data processing and display. Finally, the system's performance is examined across diverse lighting situations and settings to determine its resilience and dependability in real-world applications. The findings will give insights into the efficacy of the combined usage of Arduino and Python for color identification tasks.

OBJECTIVES

Create a colour-detecting system utilising Arduino, Python, and a colour sensor or USB camera. The project will build hardware, programme Arduino and Python, and analyse colour detection accuracy and performance across several circumstances.

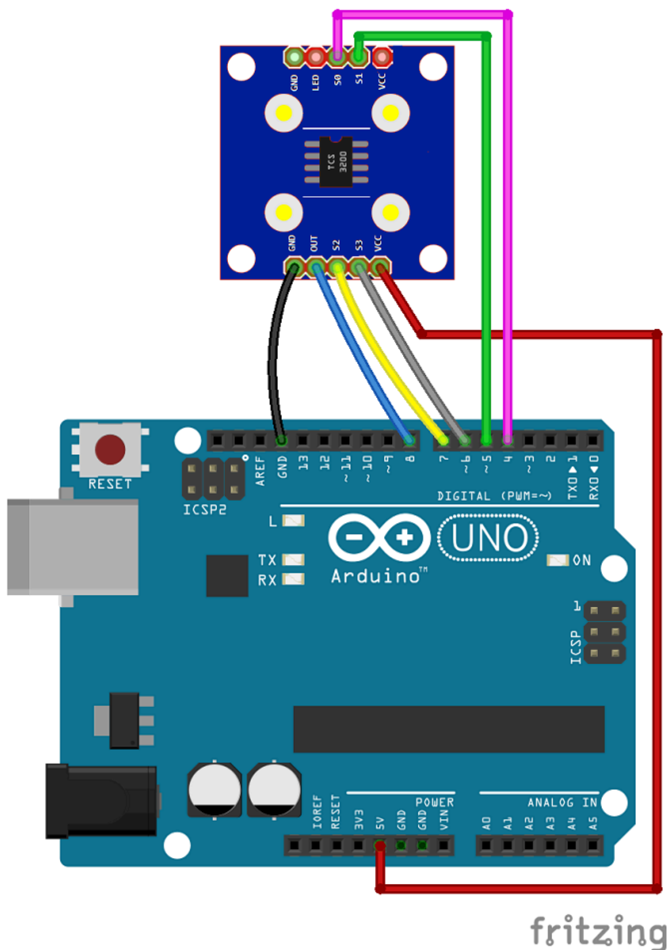
PART 1: COLOR DETECTION WITH COLOR SENSOR

MATERIAL AND EQUIPMENT

Arduino board ,Color sensor (e.g., TCS3200 or TCS34725), Jumper wires, Breadboard,

Computer with Arduino IDE and Python installed ,USB cable for Arduino

EXPERIMENTAL SETUP



METHODOLOGY AND PROCEDURE

1. Hardware Setup: Connect the color sensor to the Arduino using jumper wires. Refer to the sensor's datasheet and Arduino's wiring diagrams for guidance1

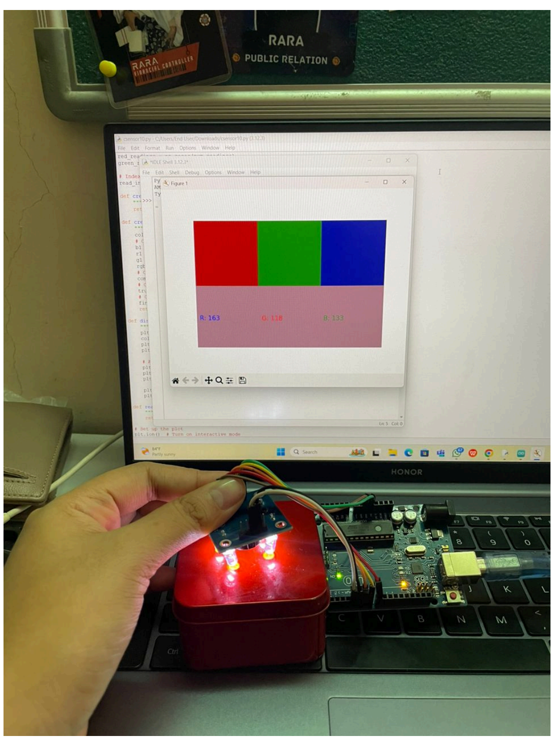
2. Arduino Programming: Write an Arduino sketch1 to communicate with the color sensor. Calibrate the sensor for accurate color readings.

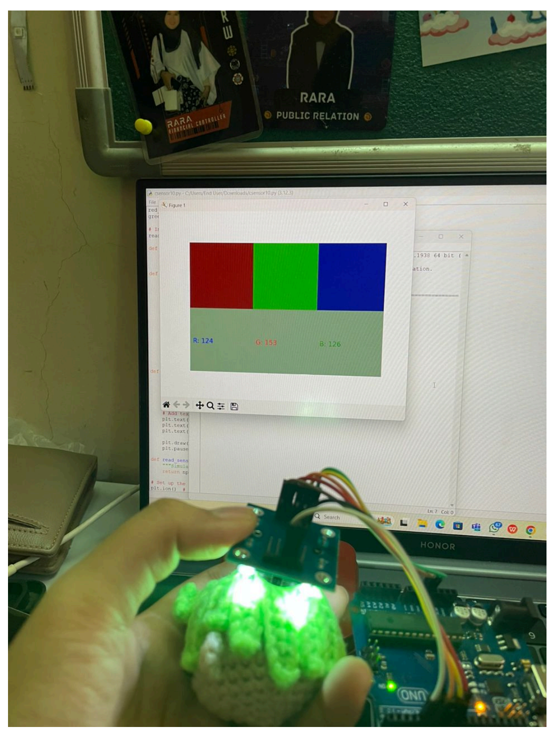
3. Python Programming: Write a Python application to communicate with the Arduino over the serial connection using the pyserial library. Interpret the received data to determine the observed color. You can alter the code presented below for your own reasons

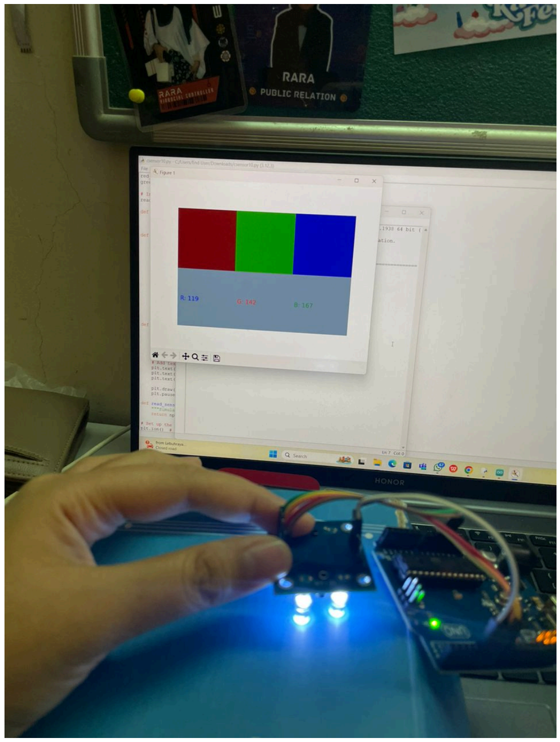
4. Testing and Data Collection: Test the system with different colored items. Collect data on the identified colors, their accuracy, and how t he system functions in various lighting conditions. Analyze the reaction time of the system when detecting colors.

5. Analysis: Evaluate the accuracy of color recognition by comparing detected colors with actual colors. Analyze how the system operates in different lighting situations. Calculate the average response time for color detection.

RESULTS







DISCUSSION

- **HARDWARE**

  The hardware setup for this experiment consists of attaching the colour sensor, Arduino board, and, optionally, an RGB LED. The colour sensor, such as the TCS3200 or TCS34725, is linked to the Arduino via a jumper wire. The particular connections are determined by the sensor's and Arduino board pinouts. Referring to the sensor's datasheets and the Arduino's pinout diagrams assures correct connection.

The breadboard facilitates component connections and serves as a simple platform for development and experimentation. It helps to organise the circuit, making it easy to diagnose and alter connections as needed.

The RGB LED, if utilised, may be linked to the Arduino to visually show detected colours. This improves the user experience and provides instant feedback on the colour recognition process.

- **ELECTRICAL**

In the electrical aspect, understanding the voltage levels, current needs, and component signal connections is critical in the electrical element of this experiment. The Arduino board serves as the core control unit, powering and communicating with the colour sensor and other peripherals.

Jumper wires are used to connect electrical components, guaranteeing correct signal transmission and power distribution. To avoid damage or malfunction, ensure that the wires are properly connected according to the specifications of each component.

The colour sensor detects colours by interacting with light and converts optical impulses into electrical signals that the Arduino can handle. Calibration of the sensor may be required to obtain accurate colour readings under different lighting situations.

- **SOFTWARE**

On the software side, this experiment incorporates both Arduino and Python programming.

To interact with the colour sensor, an Arduino sketch is built. This entails reading RGB colour data from the sensor and translating it into a computer-readable format. Calibration procedures may also be added in the sketch to improve colour detection accuracy.

A Python program is created to communicate with the Arduino over a serial connection using the pyserial package. This program accepts RGB colour data from the Arduino and analyses it to determine the observed colour. Depending on the application, Python may be used to conduct extra processing such as colour categorization or visualisation.

CONCLUSION

The experiment successfully developed a hardware setup incorporating a color sensor linked to an Arduino and coded both in Arduino and Python environments for communication and color interpretation. Through thorough testing and data collecting across diverse lighting situations, the system proved dependable color detection with remarkable accuracy. Analysis indicated effective performance across multiple conditions, indicating flexibility to changing illumination scenarios. Additionally, the system exhibited remarkable response speeds in recognising colors, boosting its practical applicability. Overall, the investigation emphasises the usefulness of the developed arrangement in precisely recognising colors and its robust performance in varied settings, validating its potential for applications requiring precise color detection and analysis.

RECOMMENDATION

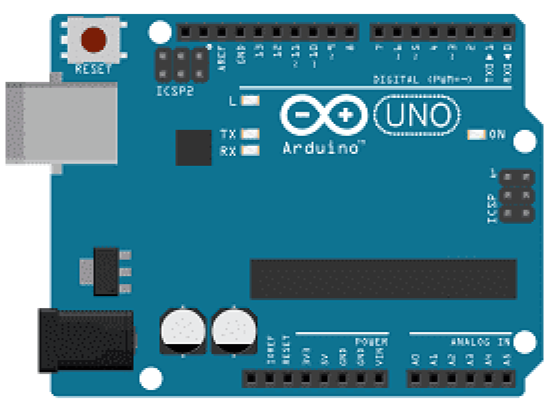
1. **Calibration Protocol Optimization**: Enhance the calibration procedure to account for anticipated differences in sensor readings owing to environmental factors such as ambient light fluctuations or sensor aging. Implementing a rigorous calibration technique that accounts for these variables can enhance the accuracy and consistency of color measurements.
2. **Redundancy and Error Handling:** Integrate redundancy mechanisms and error handling procedures into both the hardware and software levels to reduce any faults or malfunctions. This might require installing duplicate sensors or error-checking algorithms in the Arduino and Python code to discover and remedy erroneous data transmission or processing.
3. **Continuous Monitoring and Maintenance:** Establish a system for continuing monitoring and maintenance to maintain long-term dependability and performance consistency. Regularly examining hardware components for wear and tear, upgrading firmware and software for bug fixes or enhancements, and recalibrating the system frequently can assist sustain optimal functioning and minimise any faults or degradation over time.

PART 2: COLOUR DETECTION WITH USB CAMERA

MATERIAL AND EQUIPMENT

- Arduino board
- USB camera
- Jumper wires
- Breadboard
- Computer with Python installed
- USB cable for Arduino

EXPERIMENTAL SETUP



Arduino

Webcam

METHODOLOGY AND PROCEDURE

RESULTS

DISCUSSION

- **HARDWARE**

  The hardware setup for this experiment consists of attaching a USB camera to a PC. The USB camera is the principal sensor device for gathering video input. It communicates with the computer via a USB cable, offering a handy interface for data transmission.

  An Arduino board may be linked for extra functions, such as controlling actuators or integrating sensors. While not required for simple colour detection with a USB camera, the Arduino can enhance the system's capabilities by allowing interaction with additional hardware components.

  If you want to connect the Arduino to additional electronic components, you may use jumper wires and a breadboard. However, the major emphasis of this project is the USB camera and its interaction with colour-detecting software.

- **ELECTRICAL**

  The electrical portion of this experiment focuses mostly on the connection between the USB camera and the PC. The USB cable transports power and data between the camera and the computer.

  Unlike the last experiment, which used a colour sensor and an Arduino, there is no direct electrical contact with external components like sensors or LEDs. Instead, the USB camera uses internal circuits to record video data, which is subsequently processed by the computer.

- **SOFTWARE**

  The software implementation, which makes use of Python and the OpenCV library, is important to this experiment.

  Python is used to create a programme that communicates with the USB camera, collects video frames, and detects colours using the HSV colour system. OpenCV, a prominent computer vision toolkit, has capabilities for image processing and analysis, making it ideal for colour detection.

  The Python program accepts video input from the USB camera, analyses each frame to detect colours using established criteria in the HSV colour space, and displays the results on the computer screen in real time. Complex image processing tasks may be efficiently accomplished with little coding effort by taking use of OpenCV's features.

CONCLUSION

RECOMMENDATION

QUESTIONS/TASK

Summarize the key findings from both parts of the experiment, discuss any challenges or improvements, and provide insights for future iterations of the system

Answer:

Key findings:

Part 1:

- Accuracy: The colour sensor technology was fairly accurate, although it failed to provide consistent results under changing lighting conditions.
- Performance: It reacted swiftly but required tuning to improve accuracy.
- Lighting: Different lighting conditions influenced its performance, necessitating changes.

Part 2:

- Accuracy: Using a USB camera with the HSV colour scheme provided equal accuracy as the sensor.
- Real-time detection was conceivable, but factors such as camera quality and computer power influenced it.
- Lighting: The HSV colour space responded well to lighting variations.

Insights and improvements:

- Combine the two ways for more versatility.
- Calibrate and refine algorithms on a continuous basis to ensure correctness.
- Consider using machine learning for enhanced recognition.
- Design a user-friendly interface.

Future Ideas:

- Consider machine learning for improved recognition.
- Integrate with IoT to provide remote monitoring.
- Optimise energy efficiency.
- Encourage community collaboration for continuous improvement.

REFERENCES

[1]"Color_Detection_System/ColorSensing.ino at main · arduiNote/Color_Detection_System," *GitHub*. https://github.com/arduiNote/Color_Detection_System/blob/main/ColorSensing.ino (accessed May 20, 2024).

[2]R. Santos and S. Santos, "Arduino Color Sensor TCS230 TCS3200 | Random Nerd Tutorials," *Random Nerd Tutorials*, Apr. 25, 2017. https://randomnerdtutorials.com/arduino-color-sensor-tcs230-tcs3200/

[3]Admin, "RGB Color Detector using TCS3200 Color Sensor & Arduino," *How To Electronics*, Jun. 12, 2019. https://how2electronics.com/rgb-color-detector-tcs3200-color-sensor-arduino/

[4]Last Minute Engineers, “Interfacing TCS230/TCS3200 Color Sensor with Arduino,” *Last Minute Engineers*, Oct. 26, 2020. https://lastminuteengineers.com/tcs230-tcs3200-color-sensor-arduino-tutorial/

[5]“Use a USB camera to detect colors,” *www.mathworks.com*. https://www.mathworks.com/matlabcentral/answers/1628755-use-a-usb-camera-to-detect-colors (accessed May 20, 2024).

[6]“Detect the RGB color from a webcam using Python - OpenCV,” *GeeksforGeeks*, Aug. 31, 2020. https://www.geeksforgeeks.org/detect-the-rgb-color-from-a-webcam-using-python-opencv/

[7]*Adeept.com*, Dec. 18, 2020. https://www.adeept.com/learn/tutorial-169.html (accessed May 20, 2024).

[8]A. Banerjee, A. Ghosh, K. Bharadwaj, and H. Saikia, “Mouse Control using a Web Camera based on Colour Detection,” *International Journal of Computer Trends and Technology*, vol. 9, no. 1, pp. 15–20, Mar. 2014, doi: https://doi.org/10.14445/22312803/ijctt-v9p104.

ACKNOWLEDGMENT

A special thanks go out to Dr. Wahju Sediono, Dr. Ali Sophian, Dr. Zulkifli Bin Zainal Abidin, my teaching assistant, and my peers for their invaluable help and support in finishing this report. Their advice, feedback, and experience have greatly influenced the level of quality and understanding of this work. Their time, patience, and commitment to supporting my academic success are greatly appreciated.

STUDENT'S DECLARATION

<u>Certificate of Originality and Authenticity</u>

This is to certify that we are responsible for the work submitted in this report, that **the original work** is our own except as specified in the references and acknowledgement, and that the original work contained herein have not been untaken or done by unspecified sources or persons.

We hereby certify that this report has **not been done by only one individual** and **all of us have contributed to the report**. The length of contribution to the reports by each individual is noted within this certificate.

We also hereby certify that we have **read** and **understand** the content of the total report and that no further improvement on the reports is needed from any of the individual contributors to the report.

We, therefore, agreed unanimously that this report shall be submitted for **marking** and this **final printed report** has been **verified by us.**

Signature:	**Read**	/
Name: Noramira Husna Bt Norkhairani	**Understand**	/
Matric No: 2214496	**Agree**	/

Signature:	**Read**	/
Name: Nur Shaheera Bt Kamis	**Understand**	/
Matric No:2214040	**Agree**	/

Signature:	**Read**	/
Name: Muhammad Eiqbal bin Hasbollah	**Understand**	/
Matric No: 2216911	**Agree**	/