REPORT 9a: COLOR DETECTION AND ANALYSIS USING PIXY CAMERA

GROUP 4

MCTA 3203

SEMESTER 2 2023/2024

MECHATRONICS SYSTEM INTEGRATION

DATE OF SUBMISSION:
21TH 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

ABSTRACT

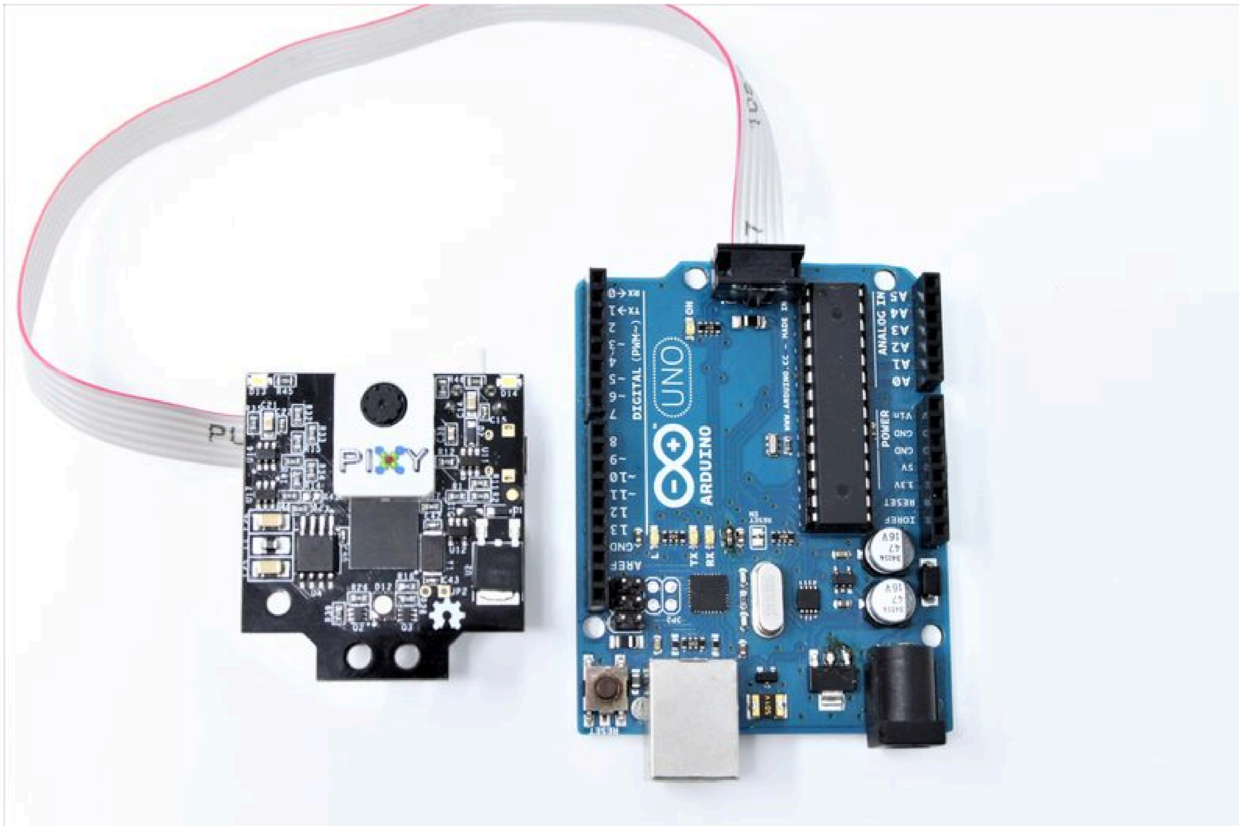
This experiment demonstrates an organised approach to creating an object detection system with a Pixy Camera and an Arduino Uno. The system's goal is to reliably identify between three coloured objects via hardware setup, colour calibration, code development, and iterative testing. Success is dependent on the system's capacity to dependably recognise objects, which necessitates changes to parameters such as colour signatures and threshold levels. Challenges such as false positives and illumination fluctuations highlight the importance of constant improvement.

Keywords: Object Detection, Pixy Camera, Arduino Uno, Colour Calibration, Threshold Optimisation, Iterative Testing.

MATERIAL & EQUIPMENT

- Pixy Camera
- Arduino Uno
- Jumper wire
- Type Mini USB
- Type B USB
- 3 different coloured Object

EXPERIMENTAL SETUP



METHODOLOGY/PROCEDURE

1. **Step 1: Hardware setup.**

   Connecting Pixy Camera to Arduino Uno:

   - Connect Pixy camera to Arduino Uno. For the I2C interface, connect the GND, +5V, SDA, and SCL pins. If you're using UART, connect GND, +5V, TX, and RX.

   Power Supply:

   - Connect the Arduino Uno to a USB or external power supply.

   Color Calibration:

- Before starting, calibrate the Pixy camera for three different coloured objects. To use PixyMon, launch the software, attach the Pixy camera, and set colour signatures for three items.

2. **Step 2: Install Pixy Library**

   Download and install the Pixy library for Arduino.

   - You may access it via the Pixy website via the Arduino Library Manager.

   Include Pixy Library in the Arduino IDE.

   - In the Arduino IDE, navigate to Sketch > Include Library > Pixy.

3. **Step 3: Write Arduino Code**

   Write code to detect objects.

   - Launch Arduino IDE and build a new sketch.

4. **Step 4: Upload Code to Arduino**

   To connect the Arduino Uno to your computer

   - use a USB cable.

   To pick the board and port in the Arduino IDE

   - navigate to Tools -> Board -> Arduino Uno and Tools -> Port -> [pick the port where Arduino is attached] respectively.

To upload code to the Arduino Uno

- use the Arduino IDE's "Upload" button.

5. **Step 5: Test and Debug**

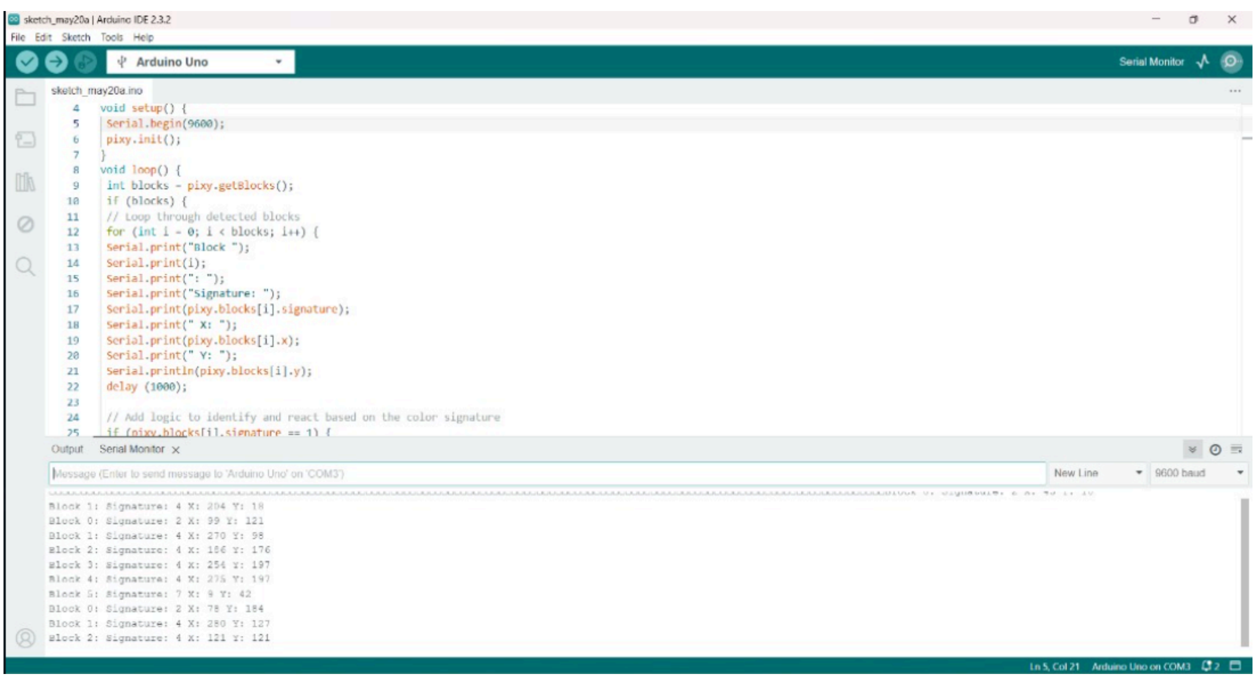
To see the output and diagnose problems

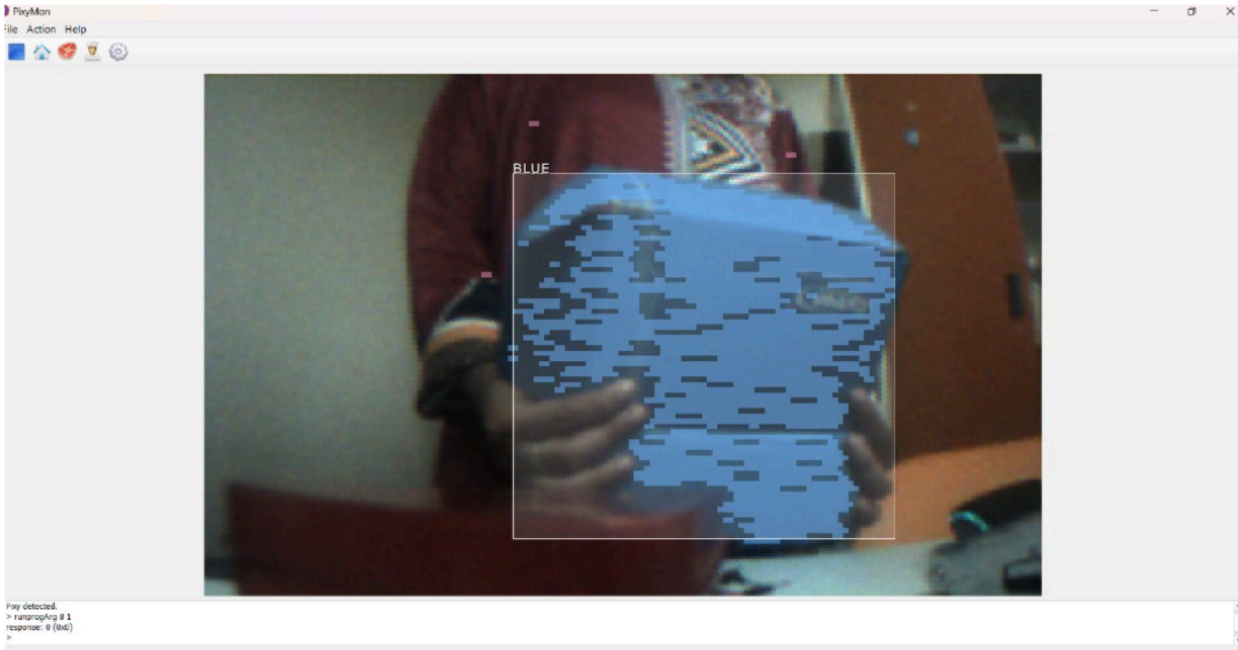
- use the Serial Monitor in the Arduino IDE (Tools -> Serial Monitor).

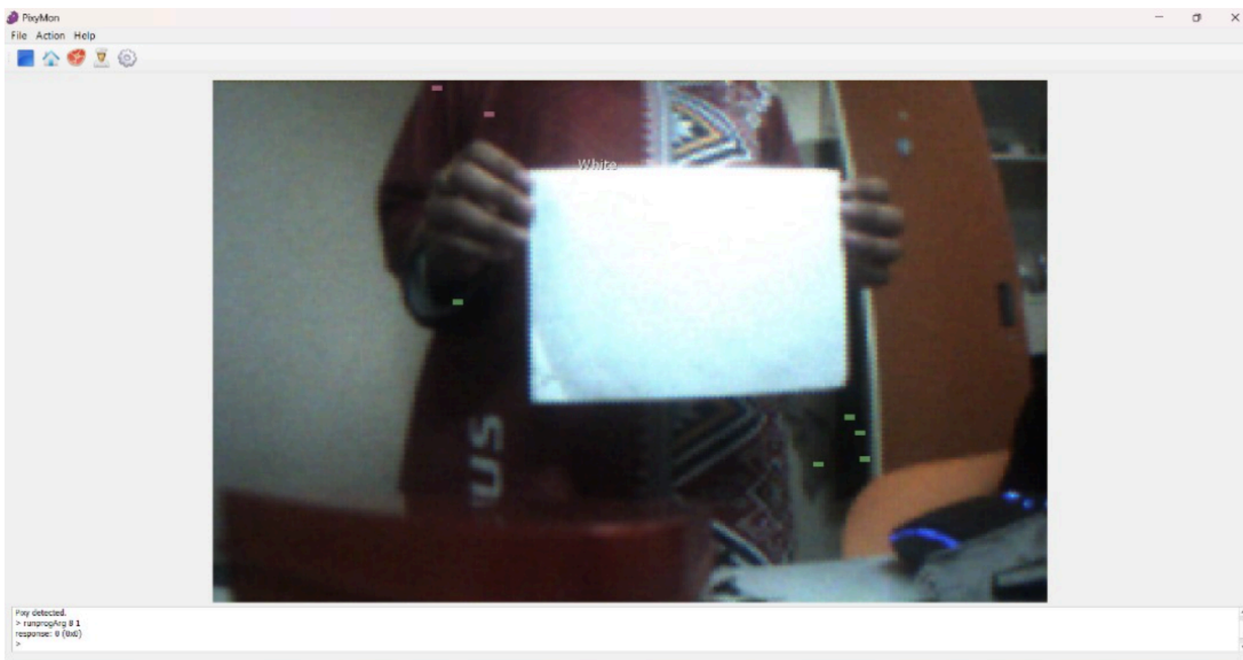
Adjust the Thresholds

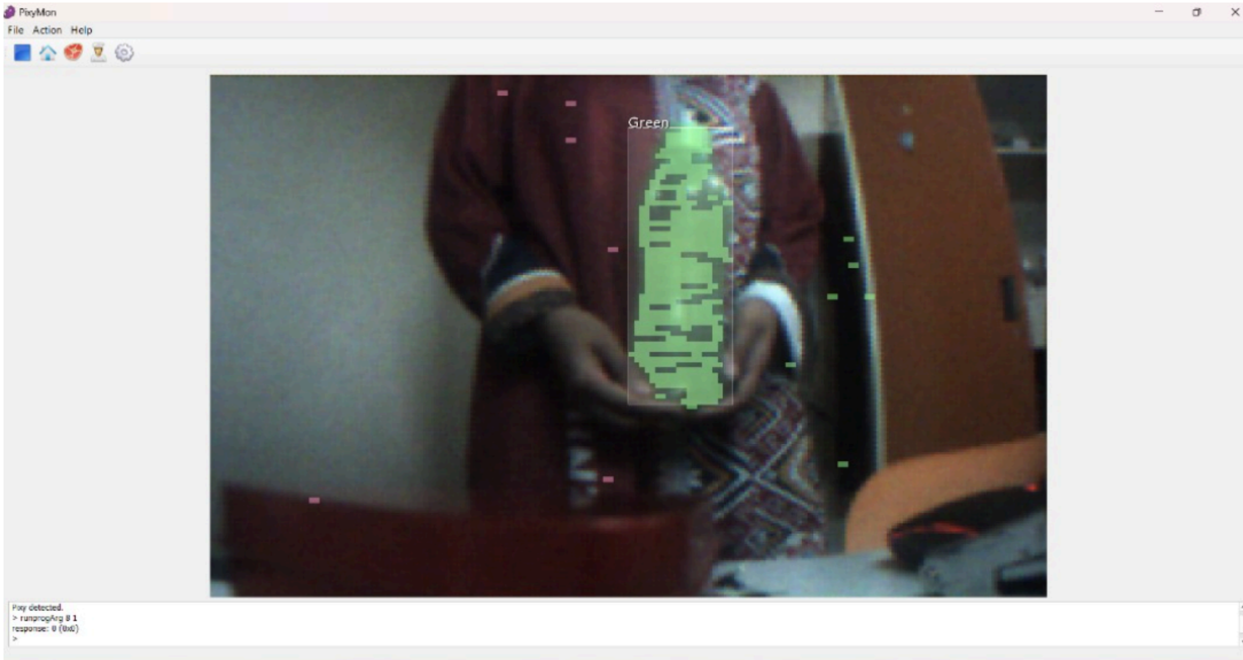
- Adjust colour signatures and threshold values in PixyMon software and Arduino code for better item recognition.

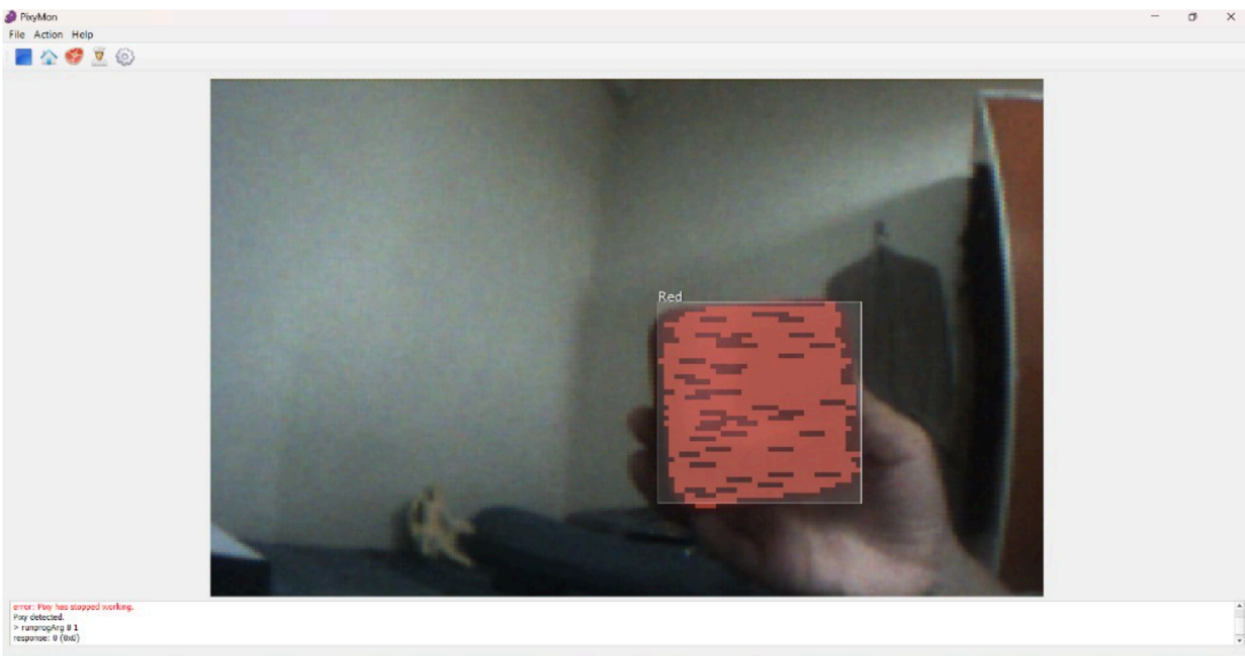
RESULTS




PixyMon
File Action Help
BLUE
Pixy detected.
> runprogArg 8 1
response: 0 (0x0)
>


PixyMon
File Action Help
White
Pixy detected.
> runprogArg 8 1
response: 0 (0x0)
>


PixyMon
File Action Help
Green
Pixy detected.
> runprogArg 8 1
response: 0 (0x0)
>


PixyMon
File Action Help
Red
error: Pixy has stopped working.
Pixy detected.
> runprogArg 8 1
response: 0 (0x0)

DISCUSSION

1. HARDWARE

The hardware configuration was a vital component of the successful object detection system employing the Pixy Camera and Arduino Uno. Proper hardware configuration allowed consistent connectivity and precise data acquisition, becoming the backbone of the detection process.

Pixy Camera

The Pixy Camera, noted for its ability to detect things based on color signatures, was picked for its applicability in real-time object tracking. Its built-in image sensor and processing skills allowed it to swiftly detect and track the colored items. Ensuring the camera was properly fixed and appropriately oriented was critical for consistent object recognition, since any movement or misalignment may result in mistakes.

Arduino Uno

The Arduino Uno functioned as the processing unit, accepting data from the Pixy Camera and running the detection algorithms. The Arduino's stable digital I/O pins permitted flawless connection with the Pixy Camera. Properly powering the Arduino and providing solid connections through adequate cabling were critical to prevent data loss and assure ongoing functioning.

Interfacing Components

The connection between the Pixy Camera and the Arduino Uno was made using conventional communication protocols. Ensuring safe and robust connections via proper cables and connectors lowered the possibility of data transmission problems. The hardware design also included concerns for power supply, ensuring that both the Pixy Camera and Arduino Uno got constant and appropriate power for maximum performance.

Stability and Environment

A steady physical setup was maintained to reduce extraneous disruptions. The components were securely fastened to prevent vibrations or shifting that may compromise detection accuracy. Additionally, the experimental setting was regulated to eliminate factors such as excessive movement or changeable lighting conditions, further contributing to the system's dependability.

Integration and Testing

Initial integration tests were undertaken to check the hardware connections and confirm that the Pixy Camera and Arduino Uno communicated successfully. These tests comprised validating the data flow from the camera to the Arduino and confirming that the Arduino could process and respond to the data appropriately.

2. SOFTWARE

The software development step was vital for the success of the object identification system employing the Pixy Camera and Arduino Uno. The software was responsible for interpreting the data from the Pixy Camera, making judgements based on color signatures, and assuring real-time object detection.

Color Calibration

Color calibration was the first crucial stage in the software process. The Pixy Camera employs color signatures to detect things, therefore the calibration included teaching the camera to recognize certain colors associated with the three target objects. This was done with the PixyMon programme, which lets users to define and fine-tune the color signatures by modifying characteristics like as hue, saturation, and brightness. Accurate calibration guaranteed that the camera could dependably discriminate between the target colors.

Data Processing

Once calibrated, the Pixy Camera communicated the color signature data to the Arduino Uno. The software on the Arduino was responsible for deciphering this data. The Arduino code, developed in the Arduino IDE, contained libraries and routines to manage the communication with the Pixy Camera over serial or I2C interfaces. The programme analysed the incoming data to determine the existence and position of the colored items.

Real-Time Detection

Real-time detection was a critical need for the system. The Arduino code was adjusted to accommodate continuous data streams from the Pixy Camera without major delays. This entailed effective use of data structures and control flow methods to guarantee that the system could respond rapidly to identified items. The software loop continually read data from the Pixy Camera, evaluated it, and triggered suitable reactions based on the observed objects' locations.

Threshold Optimization

To boost detection accuracy, the programme provided procedures for altering threshold values. These criteria helped filter out noise and extraneous data, concentrating exclusively on the significant color signatures. By fine-tuning these levels through recurrent testing, the system's accuracy in differentiating between identical hues and eliminating false positives was greatly enhanced.

Iterative Testing and Refinement

The software development method was iterative, including continual testing and modification. Initial versions of the programme were tested under various situations to uncover flaws such as false positives and missed detections. Feedback from these experiments guided revisions to the calibration settings, threshold levels, and data processing algorithms. This iterative method meant that the programme got more resilient and dependable over time.

Error Handling and Robustness

The programme was built to manage any faults and oscillations in data. This includes developing checks for out-of-range numbers, guaranteeing robust communication with the Pixy Camera, and keeping consistent performance under varying lighting circumstances. Robust error handling systems were critical for sustaining system dependability, especially in dynamic contexts.

Integration with Hardware

Finally, the software was intimately linked with the hardware arrangement. The interaction between the Pixy Camera and the Arduino Uno needed exact timing and coordination. The software catered for the hardware's limitations and capabilities, ensuring that the system as a whole ran effortlessly.

3. ELECTRICAL

The electrical setup was a critical component of the successful construction of the object detection system utilising the Pixy Camera and Arduino Uno. This step guaranteed that all components were appropriately powered and linked, allowing flawless functioning and communication between the hardware pieces.

Power Supply

A steady and appropriate power source was important for the dependable functioning of both the Pixy Camera and the Arduino Uno. The Arduino Uno was normally powered by a USB connection from a computer or an external power adapter giving 5V. The Pixy Camera requires a separate power supply, generally delivered by the Arduino’s 5V and GND pins. Ensuring that both devices received sufficient power without fluctuations was crucial to prevent resets or data loss.

Wiring and Connections

Proper wiring and secure connections were vital to the system's functionality. The connections between the Pixy Camera and Arduino Uno included numerous pins, including power (5V and GND), data transmission (I2C or UART), and control signals.

Power Connections: The 5V and GND pins of the Pixy Camera were linked to the equivalent pins on the Arduino to provide a reliable power source.

Data Communication: The Pixy Camera connects with the Arduino Uno using either I2C or UART protocol. For I2C connection, the SDA (data) and SCL (clock) lines were linked to the respective analog pins (A4 and A5) on the Arduino. For UART communication, the TX (transmit) and RX (receive) lines were linked to the digital pins (D0 and D1).

CONCLUSION

In Conclusion, the project demonstrated an organised process for building an object recognition system using a Pixy Camera and an Arduino Uno, highlighting the system's ability to distinguish between coloured items. Success is dependent on the system's ability to produce this separation consistently, which requires ongoing testing and refining. Challenges like as false positives, illumination variances, and threshold errors highlight the importance of continuing optimisation. Further refining entails altering colour signatures and threshold settings to improve detection accuracy under a variety of situations.

RECOMMENDATION

1. **Distinct Colour Selection:**

   Select unique and easily recognised colours for the items to improve colour calibration accuracy. Test different colour combinations to reduce overlap and confusion during object recognition.

2. **Threshold Value Experiment:**

   To improve colour recognition performance, experiment with different threshold settings. To increase the accuracy of object identification, adjust the sensitivity settings in both the PixyMon software and Arduino code.

3. **Controlled Environment Setup:**

To decrease variability in item identification, carry out the experiment in a controlled setting with constant illumination. Avoid intense sunshine and strong artificial lighting, which can alter colour vision.

REFERENCES

1. *Documentation for functions*. (2023, January 25). Pixycam. https://forum.pixycam.com/t/documentation-for-functions/7874
2. J3. (2019, July 25). *Arduino Meets Pixy*. Jungletronics. https://medium.com/jungletronics/arduino-meets-pixy-36138c474912
3. Workshop, D. (2018, October 19). *Pixy2 Camera – Simple Object Recognition camera*. DroneBot Workshop. https://dronebotworkshop.com/pixy2-camera/

ACKNOWLEDGEMENT

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 DECLARATION

Certificate of Originality and Authenticity

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**	/