MECHATRONICS SYSTEM INTEGRATION

EXPERIMENT WEEK 9 & 9a: IMAGE/VIDEO INTERFACING WITH MICROCONTROLLER

GROUP NUMBER: A

PROGRAMME: MECHATRONICS

GROUP MEMBERS	MATRIC NO
AISYAH SOFEA BINTI OTHMAN	2115386
SITI ALIAA BINTI IBRAHIM	2112618
MUHAMMAD ADLI FAHMI BIN TAJUL ARIS	2113095
ETHAR ABDALLA ABDELKARIM OSMAN	2111282
NURAIN AINAA AQILAH BINTI ROSLI	2114560

DATE OF EXPERIMENT:

Wednesday, 13th December 2023

DATE OF SUBMISSION:

Wednesday, 20th December 2023

Abstract

This report outlines two experiments conducted with an Arduino Uno, a colour sensor (RGB sensor), an RGB LED, and a Pixy camera. The first experiment focuses on utilising the RGB sensor to detect the predominant colour and subsequently lighting up an RGB LED accordingly. The second experiment explores the integration of the Pixy camera to detect a specific signature colour. Both parts of the experiment were successfully executed.

Table of Contents

Objectives

1. To design a colour detection system using Arduino, Python, and a colour sensor
2. To analyse the accuracy and performance of colour detection in different scenarios.
3. To build a system using an Arduino Uno and Pixy camera to detect three different coloured objects

Introduction

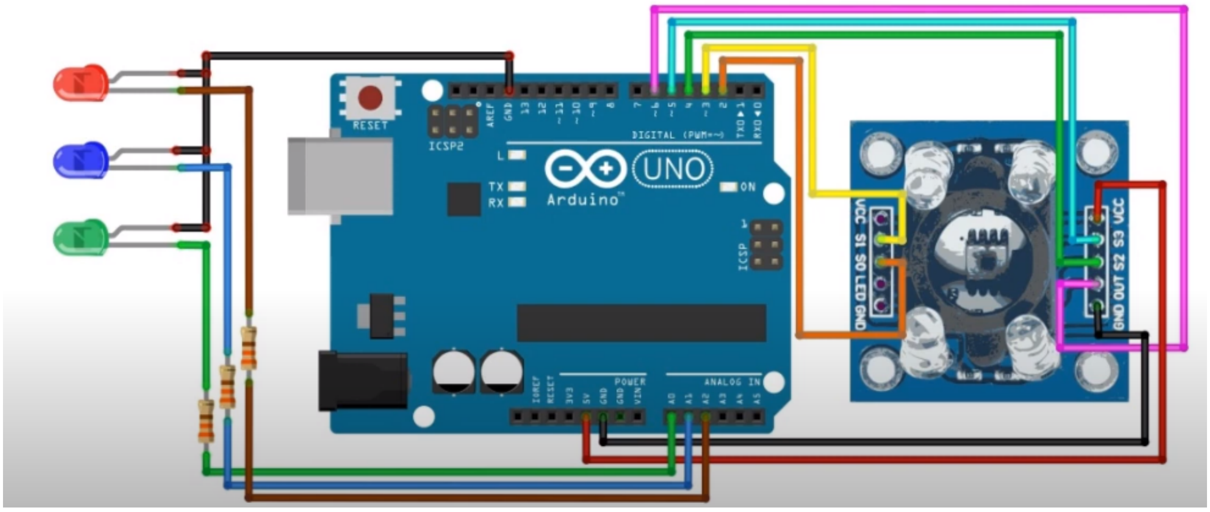
The first experiment employs an RGB sensor to capture colour information, determining the dominant colour in the sensor's field of view. Subsequently, an RGB LED is activated to emit light corresponding to the detected colour. This demonstration serves as a foundation for understanding colour-sensing applications and introduces the fundamental principles of colour-based data acquisition. The second experiment involves the Pixy camera, a vision sensor capable of recognizing specific colours or patterns. By interfacing the Pixy camera with the Arduino Uno, we demonstrate the detection of a signature colour. This experiment highlights the potential of visual recognition in real-time applications and introduces the concept of using predefined signatures for targeted colour identification. Both experiments contribute to the exploration of sensor integration with Arduino platforms, showcasing practical applications in colour sensing and visual recognition.

Material and Equipment

1. Arduino board
2. Colour sensor
3. Jumper wires
4. Breadboard
5. Red, Green, Blue LED
6. Computer with Arduino IDE and Python installed
7. USB cable for Arduino
8. Colour Paper (Green, Red, Blue)
9. Pixy camera

Experimental Setup

<u>Part 9</u>

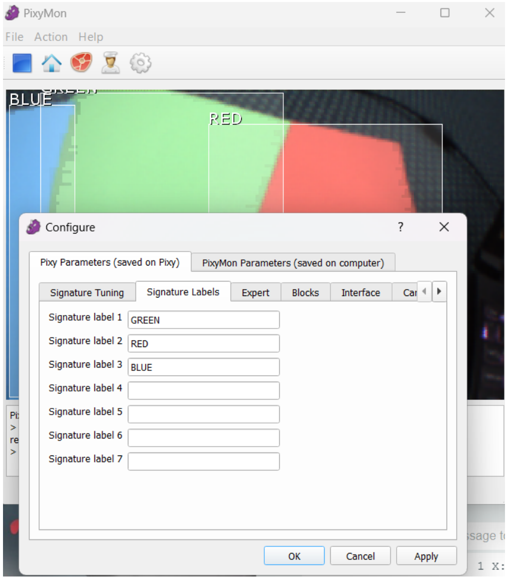


Connection for LED and Colour Sensor

In this experiment, the colour sensor detects RGB values and converts them into a specific colour. We connect the colour sensor to the Arduino by the sensor OUT pin to an input on the Arduino which is digital pin 6, VCC to 5V, GND to GND, S1 to digital pin 2, S0 to digital pin 3, S2 to digital pin 4, and S3 to digital in 5. We use RED, Yellow, and Green LEDs as output indicators based on the detected colour. We Connect the longer leg (anode) of the LED to a digital output pin on the Arduino through a current-limiting 220-ohm resistor. Connect the shorter leg (cathode) to GND.

<u>Part 9a</u>

An Arduino Uno and a Pixy camera are integrated into this experimental setup to create a colour-tracking system. The Pixy camera is connected to the Arduino Uno using I2C. The Arduino Uno is powered either through USB. Then, Color calibration is performed using the PixyMon software on a computer. The Pixy camera is connected to the computer via USB, and colour signatures for three different colour objects (red, green, blue) are configured using the PixyMon interface.



Calibrate Pixy camera using PixyMon software.

After calibration, the Pixy camera is disconnected from the computer, and an Arduino sketch is uploaded to the Arduino Uno to facilitate communication with the Pixy camera. The sketch reads colour information from Pixy and includes logic to take actions based on the recognized colours. Coloured objects are then placed within Pixy's field of view, and the experimental setup is powered up to observe how Pixy tracks and recognizes the coloured objects based on the configured colour signatures.

Methodology

Part 9: Colour Detection with Color Sensor

To conduct the colour detection experiment, connect a colour sensor (e.g., TCS3200) to an Arduino and upload the provided Arduino code. In this experiment, we used 3 LEDs which are red, green and yellow, but the yellow LED will detect the blue-coloured paper because we do not have a blue LED. Establish a serial connection between the Arduino and a computer using a USB cable. Modify the Python code to match the correct COM port and run the experiment. The Arduino continuously reads RGB values from the colour sensor and sends them to Python over the serial connection. In Python, receive and interpret the RGB data to determine the detected colour based on predefined thresholds. Print the detected colour in the Python console, experiment with different coloured objects, and observe how the detected colour changes.

Arduino Code	**Python Code**
#define S0 2	import serial
#define S1 3	import matplotlib.pyplot as plt
#define S2 4	from matplotlib.animation import
#define S3 5	FuncAnimation
#define OUT 6	
	# Open serial connection to Arduino (adjust
#define LedRed A2	COM port as needed)
#define LedGreen A0	arduino = serial.Serial('COM3', 9600,
#define LedBlue A1	timeout=1)
int R,G,B = 0;	# Initialize lists to store RGB values and
	time
void setup() {	times = []
pinMode(S0, OUTPUT);	reds = []
pinMode(S1, OUTPUT);	greens = []
pinMode(S2, OUTPUT);	blues = []
pinMode(S3, OUTPUT);	
pinMode(OUT, INPUT);	# Function to update the graph
	def update_graph(frame):
pinMode(LedRed, OUTPUT);	# Read data from Arduino
pinMode(LedGreen, OUTPUT);	arduino_data =
pinMode(LedBlue, OUTPUT);	arduino.readline().decode('utf-8').strip()
// Setting frequency-scaling to 20%	# Split RGB values
digitalWrite(S0,HIGH);	red, green, blue = map(int,
digitalWrite(S1,LOW);	arduino_data.split(','))
Serial.begin(9600);	# Append values to lists
	times.append(frame)
for (int i=0; i<=5; i++){	reds.append(red)
digitalWrite(LedRed,	greens.append(green)

!digitalRead(LedRed));
    digitalWrite(LedGreen,
!digitalRead(LedGreen));
    digitalWrite(LedBlue,
!digitalRead(LedBlue));
    delay(250);
  }
}
void loop() {
  // Setting red filtered photodiodes to be read Red frequency
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  R = pulseIn(OUT, LOW); // Reading the output Red frequency
  delay(100);
  // Setting Green filtered photodiodes to be read Green frequency
  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);
  G = pulseIn(OUT, LOW); // Reading the output Green frequency
  delay(100);
  // Setting Blue filtered photodiodes to be read Blue frequency
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  B = pulseIn(OUT, LOW); // Reading the output Blue frequency
  delay(100);
 	
//--
	blues.append(blue)
	
	    # Interpret the received data to determine the detected color
	    if red > 200 and green < 100 and blue < 100:
	        detected_color = "Red"
	    elif red < 100 and green > 200 and blue < 100:
	        detected_color = "Green"
	    elif red < 100 and green < 100 and blue > 200:
	        detected_color = "Blue"
	    else:
	        detected_color = "Unknown"
	
	    # Display the detected color
	    print(f"Detected Color: {detected_color}")
	
	    # Update the graph
	    ax.clear()
	    ax.plot(times, reds, label='Red', color='red')
	    ax.plot(times, greens, label='Green', color='green')
	    ax.plot(times, blues, label='Blue', color='blue')
	    ax.legend()
	    ax.set_title('RGB Values Over Time')
	    ax.set_xlabel('Time')
	    ax.set_ylabel('RGB Values')

------Detect colors based on sensor values
  if (R>24 && R<35 && G>70 && G<105 && B>50 && B<90){   // to detect red
    digitalWrite(LedRed, HIGH);
    Serial.print("RED ON: ");
  }
  else if (R>40 && R<100 && G>30 && G<85 && B>50 && B<95){ // to detect green
    digitalWrite(LedGreen, HIGH);
    Serial.print("GREEN ON : ");
  }
  else if (R>50 && R<100 && G>20 && G<80 && B>10 && B<60){ // to detect blue
    digitalWrite(LedBlue, HIGH);
    Serial.print("BLUE ON : ");
  }
  else{
    digitalWrite(LedRed, LOW);
    digitalWrite(LedGreen, LOW);
    digitalWrite(LedBlue, LOW);
  }
 	
//--
 	
  // Print RGB Sensor Values
  Serial.print("R= ");
  Serial.print(R);
  Serial.print(" | ");
  Serial.print("G= ");
  Serial.print(G);
	# Create a live graph
	fig, ax = plt.subplots()
	ani = FuncAnimation(fig, update_graph, interval=1000)
	
	# Show the graph
	plt.show()
	
	# Close the serial connection and plot on exit
	arduino.close()

Serial.print(" | ");
Serial.print("B= ");
Serial.print(B);
Serial.println();
delay(200);
}	

This is the code to colour sensor read the sensor out :

Arduino Code

```
#define S0  2
#define S1  3
#define S2  4
#define S3  5
#define OUT 6

#define LedRed    A2
#define LedGreen  A0
#define LedBlue   A1

int R,G,B = 0;

void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(OUT, INPUT);

  pinMode(LedRed, OUTPUT);
  pinMode(LedGreen, OUTPUT);
  pinMode(LedBlue, OUTPUT);
```

```
  // Setting frequency-scaling to 20%
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);

  Serial.begin(9600);

  // To make the color indicator LED blink, ignore it if you do not use the color indicator
LED.
  for (int i=0; i<=5; i++){
    digitalWrite(LedRed, !digitalRead(LedRed));
    digitalWrite(LedGreen, !digitalRead(LedGreen));
    digitalWrite(LedBlue, !digitalRead(LedBlue));
    delay(250);
  }
}
void loop() {
  // Setting red filtered photodiodes to be read Red frequency
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  R = pulseIn(OUT, LOW);  // Reading the output Red frequency
  delay(100);
  // Setting Green filtered photodiodes to be read Green frequency
  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);
  G = pulseIn(OUT, LOW);  // Reading the output Green frequency
  delay(100);
  // Setting Blue filtered photodiodes to be read Blue frequency
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  B = pulseIn(OUT, LOW);  // Reading the output Blue frequency
  delay(100);
  // Print RGB Sensor Values
```

```
  Serial.print("R= ");
  Serial.print(R);
  Serial.print(" | ");
  Serial.print("G= ");
  Serial.print(G);
  Serial.print(" | ");
  Serial.print("B= ");
  Serial.print(B);
  Serial.println();
  delay(200);
}
```

Part 9a

To conduct the colour detection experiment using the Pixy Camera and Arduino, start by connecting the Pixy Camera to the Arduino using jumper wires, linking GND to GND, VCC to 5V, and TX to RX. Install PixyMon software on your computer, configure colour signatures, and teach Pixy about the colours of interest. Install the Pixy library in the Arduino IDE and write code to communicate with the Pixy Camera, read colour signatures, and print detected colours via the serial connection. Upload the code to the Arduino, disconnect the Pixy Camera from the computer, and connect it to the Arduino. Power the Arduino, open the Serial Monitor, and observe the detected colour signatures. Library Inclusion: This line includes the Pixy library, allowing the Arduino sketch to use Pixy-related functions and features.

```
#include <Pixy.h> //Library Inclusion
Pixy pixy;
void setup() {  //this function is executed once when the Arduino is POwered on or reset
  Serial.begin(9600); // Initializes serial communication with a baud rate of 9600
  pixy.init(); // initializzes the Pixy camera
}
void loop() {
  int blocks = pixy.getBlocks();
```

```
 if (blocks) {
   // Loop through detected blocks
   for (int i = 0; i < blocks; i++) {
     Serial.print("Block ");
     Serial.print(": ");
     Serial.print(i);// display how many blocks detected
     Serial.print("Signature: ");
     Serial.print(pixy.blocks[i].signature);
     Serial.print(" X: "); // display X-Coordinates
     Serial.print(pixy.blocks[i].x);
     Serial.print(" Y: "); // Display Y-Coordinates
     Serial.println(pixy.blocks[i].y);

     // Add logic to identify and react based on the color signature
     if (pixy.blocks[i].signature == 1) {
       // Object with signature 1 detected (Color green)
       Serial.print("green");
     } else if (pixy.blocks[i].signature == 2) {
       // Object with signature 2 detected (Color red)
       Serial.print("red");
     } else if (pixy.blocks[i].signature == 3) {
       // Object with signature 3 detected (Color blue)
       Serial.print("blue");
     }delay(1000);
   }
 }


}
```

Data Collection

<u>Part 9 : Colour Detection with Color Sensor</u>

S0	S1	OUTPUT FREQUENCY SCALING (f_o)
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

S2	S3	PHOTODIODE TYPE
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

Table 9.0 : Color Light to Frequency Converter for GY-31 TCS3200

Coloured Paper	LED Colour Representation (in the circuit)	Range of Colour Sensor Value to Turn on the LED		Accuracy	Performance under	
					Room Lighting	Slightly High Lighting
Red	Red	R	24 to 35	High	High	High
		G	70 to 105			
		B	50 to 90			
Green	Green	R	40 to 100	Medium	High	Low
		G	30 to 85			
		B	50 to 95			
Blue	Yellow	R	50 to 100	High	High	High
		G	20 to 80			
		B	10 to 60			

Table 9.1: Data collection on detection of colour sensor given certain condition and value

**the text with colour is representing the LED colour that light up*

Timestamp	**Explanation**
17:46:07.589 -> R= 315 | G= 305 | B= 225 17:46:08.054 -> R= 272 | G= 260 | B= 255 17:46:08.583 -> R= 261 | G= 209 | B= 51 17:46:09.068 -> BLUE ON : R= 71 | G= 32 | B= 24 17:46:09.579 -> BLUE ON : R= 61 | G= 27 | B= 18 17:46:10.071 -> BLUE ON : R= 58 | G= 27 | B= 18 17:46:10.582 -> BLUE ON : R= 58 | G= 28 | B= 18 17:46:11.052 -> BLUE ON : R= 58 | G= 28 | B= 18 17:46:11.567 -> BLUE ON : R= 58 | G= 27 | B= 18 17:46:12.086 -> BLUE ON : R= 58 | G= 28 | B= 19 17:46:12.588 -> BLUE ON : R= 53 | G= 28 | B= 19 17:46:13.072 -> BLUE ON : R= 66 | G= 34 | B= 26 17:46:13.568 -> R= 226 | G= 202 | B= 145 17:46:14.084 -> GREEN ON : R= 61 | G= 59 | B= 62 17:46:14.597 -> GREEN ON : R= 60 | G= 49 | B= 59 17:46:15.078 -> GREEN ON : R= 57 | G= 56 | B= 59 17:46:15.578 -> GREEN ON : R= 58 | G= 56 | B= 59 17:46:16.086 -> GREEN ON : R= 56 | G= 56 | B= 59 17:46:16.594 -> GREEN ON : R= 58 | G= 51 | B= 60 17:46:17.106 -> GREEN ON : R= 59 | G= 59 | B= 56 17:46:17.608 -> GREEN ON : R= 59 | G= 58 | B= 61 17:46:18.093 -> GREEN ON : R= 57 | G= 57 | B= 60 17:46:18.594 -> GREEN ON : R= 56 | G= 56 | B= 59 17:46:19.113 -> GREEN ON : R= 55 | G= 55 | B= 59 17:46:19.605 -> GREEN ON : R= 56 | G= 55 | B= 59 17:46:20.096 -> GREEN ON : R= 57 | G= 56 | B= 60 17:46:20.617 -> BLUE ON : R= 52 | G= 58 | B= 27 17:46:21.088 -> GREEN ON : R= 60 | G= 60 | B= 63 17:46:21.590 -> GREEN ON : R= 59 | G= 58 | B= 61 17:46:22.113 -> GREEN ON : R= 59 | G= 57 | B= 61 17:46:22.599 -> GREEN ON : R= 58 | G= 56 | B= 60 17:46:23.123 -> GREEN ON : R= 52 | G= 57 | B= 61 17:46:23.600 -> GREEN ON : R= 59 | G= 51 | B= 60 17:46:24.117 -> GREEN ON : R= 60 | G= 59 | B= 63 17:46:24.616 -> BLUE ON : R= 61 | G= 59 | B= 13 17:46:25.113 -> GREEN ON : R= 60 | G= 58 | B= 61	1. *Blue On* -referring to when the 'blue' colour detected *2.Green On* -referring to when the 'green' colour detected *3. R= ... | G=.... | B=...* -referring to when none of the 'red', 'blue' or 'green' colour is detected and is the time when we change the colour of the paper.

Table 9.2 : Output from 'Serial Monitor' on Arduino upon detection of colour change

Part 9a

```
redBlock 2: Signature: 3 X: 176 Y: 99
blueBlock 0: Signature: 2 X: 160 Y: 127
redBlock 1: Signature: 2 X: 316 Y: 2
redBlock 0: Signature: 2 X: 182 Y: 44
redBlock 0: Signature: 1 X: 217 Y: 99
greenBlock 1: Signature: 2 X: 9 Y: 187
redBlock 2: Signature: 2 X: 44 Y: 16
redBlock 3: Signature: 3 X: 118 Y: 7
blueBlock 0: Signature: 1 X: 160 Y: 99
greenBlock 0: Signature: 1 X: 160 Y: 99
greenBlock 0: Signature: 1 X: 160 Y: 99
greenBlock 0: Signature: 1 X: 160 Y: 99
greenBlock 0: Signature: 1 X: 160 Y: 99
greenBlock 0: Signature: 1 X: 160 Y: 99
greenBlock 0: Signature: 3 X: 215 Y: 96
blue
```

The pixy camera will read and recognize the coloured objects based on the colour signatures configured colour signatures then send the information to the Arduino to be displayed. The information that is displayed starts with a number of blocks detected, signature, X-coordinate, Y-coordinate and lastly signature label or colour.

Data Analysis

Part 9 : Colour Detection with Colour Sensor

The accuracy of the colour sensor is red/blue > green. The actual colours that are used are coloured paper. The colour of the paper can be seen in the video. Only green LED light lit up when the colour sensor was put down on the non-green-coloured table.

In normal (room) lighting, all LEDs only light up to their respective detected colours by the colour sensor. However, when slightly higher lighting is used, the green LEDs sometimes light up. Hence, the system performs slightly poorly in higher lighting due to green sensors.

The calculation for average response time for colour detection:

timestamp used for calculation

```
17:46:11.567 -> BLUE ON : R= 58 | G= 27 | B= 18
17:46:12.086 -> BLUE ON : R= 58 | G= 28 | B= 19
17:46:12.588 -> BLUE ON : R= 53 | G= 28 | B= 19
17:46:13.072 -> BLUE ON : R= 66 | G= 34 | B= 26
17:46:13.568 -> R= 226 | G= 202 | B= 145
17:46:14.084 -> GREEN ON : R= 61 | G= 59 | B= 62
17:46:14.597 -> GREEN ON : R= 60 | G= 49 | B= 59
17:46:15.078 -> GREEN ON : R= 57 | G= 56 | B= 59
17:46:15.578 -> GREEN ON : R= 58 | G= 56 | B= 59
17:46:16.086 -> GREEN ON : R= 56 | G= 56 | B= 59
```

Total Responses, N = 10

Eg: t_1 = 11.567 ms

$$\text{Average Response Time} = \frac{\text{Total time taken to respond}}{\text{Total \# of responses}}$$

$$= \frac{\sum_{1}^{10} t}{N}$$

$$= \frac{138.304}{10}$$

$$= 13.830 \text{ ms}$$

<u>Part 9a</u>

"Block :" indicates the number of colour blocks that the pixy camera can identify and "0" means that it is the only visible colour on the screen. Otherwise, that is another colour block that it cannot identify or has not been signed yet on the screen. In this experiment, only 3 signature colours are set which are green, red and blue respectively. Regarding the X and Y coordinates, the location from which blocks its present is unidentified. The only difficulty we faced was that the colour depth may change depending on the power input after the calibration and signature addition, which could prevent colour blocks from being detected.

Result

<u>Part 9:</u>

The colour detection experiment produced accurate and consistent results under normal lighting conditions. The colour sensor successfully distinguished between red, green and blue based on the defined thresholds. Table 9.0 provides a summary of the LED colour representation and the corresponding colour sensor values for each colour. The accuracy of the colour detection was high, with the expected led lighting up when the corresponding colour was presented to the sensor. Under slightly higher lighting conditions, the green LED occasionally lit up, indicating a decrease in performance in presence of additional light. Despite this, the system demonstrated reliable colour detection. Table 9.1 presents the data collected during the experiment, indicating the detected colour at different times. The system responded appropriately to changes in colour, with the serial monitor output providing real-time information about the detected colours. The average response time for colour detection was calculated to be approximately 13.38 milliseconds.

<u>Part 9a:</u>

The pixy camera experiment successfully recognized and tracked coloured objects based on defined colour signatures. The serial monitor output displayed information about the number of detected blocks, their signatures and their positions. The experiment focused on three colours (green, red and blue). Table 9.2 illustrates the output from the serial monitor upon the detection of colour changes. The “block” value indicates the number of colour blocks detected, and the signature labels correspond to the defined colours. The pixy camera accurately identified the colour blocks and transmitted the information to the Arduino for display.

Discussion

<u>Part 9</u>

- Arduino UNO

The Arduino microcontroller is responsible for reading data from the colour sensor and making decisions depending on it. The colour sensor TCS3200 and LED were connected to the Arduino uno board to interface with each other. The Arduino code was written and programmed to prompt the LED to turn on when the colour was detected.

- Colour sensor TCS3200

The colour sensor TCS3200 was used to detect the colours red, green and blue. The sensor generates digital outputs based on the intensity of each colour component. The Arduino reads these outputs to determine the colour of the object. The RGB colour data collected from the sensor was read and converted to a computer-readable representation through Arduino.

- LED conduct

We used red, green and yellow LEDs to represent each colour however the paper colours were red, green and blue respectively. When we direct the coloured paper to the sensor, the represented LED will turn on. For example, if we direct the red paper to the sensor, the red LED will turn on, and if we direct the blue paper to the sensor, the yellow LED will turn on.

<u>Part 9A</u>

- Pixy camera

The Pixy camera is intended to recognise coloured objects by analysing colour signatures. Based on the colour signatures configured, the Pixy camera successfully recognised and tracked coloured objects. The Serial Monitor output revealed the number of identified blocks, as well as their signatures and positions. The demonstration demonstrated the Pixy camera's ability to recognise predetermined colours and transfer this information to the Arduino.

Conclusion

In conclusion, both experiments conducted to execute image/video interfacing with a microcontroller were successful. Specifically utilising a colour sensor in part 9 and integrating a pixy camera in part 9a. The colour detection with colour sensor experiment demonstrated the Arduino Uno’s capability to detect and differentiate between red, green and blue using the colour sensor. The pixy camera accurately identified the colours.

Recommendation

The need for improvement is always needed to make sure the accuracy, reliability, and adaptability of the experiment results. Researchers can enhance their techniques and cut down on possible error sources through continuous improvement. The findings become more reliable, yielding more exact and accurate outcomes. Below are a few recommendations that we can improve in the future.

1. **Calibration and signature stability:** It is critical to guarantee consistent colour signatures and calibration in the Pixy camera experiment (Part 9a). If the lighting condition changes or the power input fluctuates, the system's performance may be affected. In the future we should ensure the stability of colour signatures and calibration over time.

2. **Enhance the lighting handling:** The lightning condition will affect the performance of the colour sensor throughout the experiment. In the future, we should consider adding external components to enhance the colour detection of the sensor so that the sensor could adapt or not be affected by higher lighting conditions.

3. **Integration of multiple sensors:** For both part 9 and 9 A we should consider integrating multiple colour sensors and cameras to extend the experiment to improve

the sensor's colour detection. Multiple sensors that are placed at different locations can increase the colour detection coverage.

By implementing these recommendations, we can improve the system's reliability and efficiency. Hence, it will reduce the errors that might occur in the future.

References

Link:

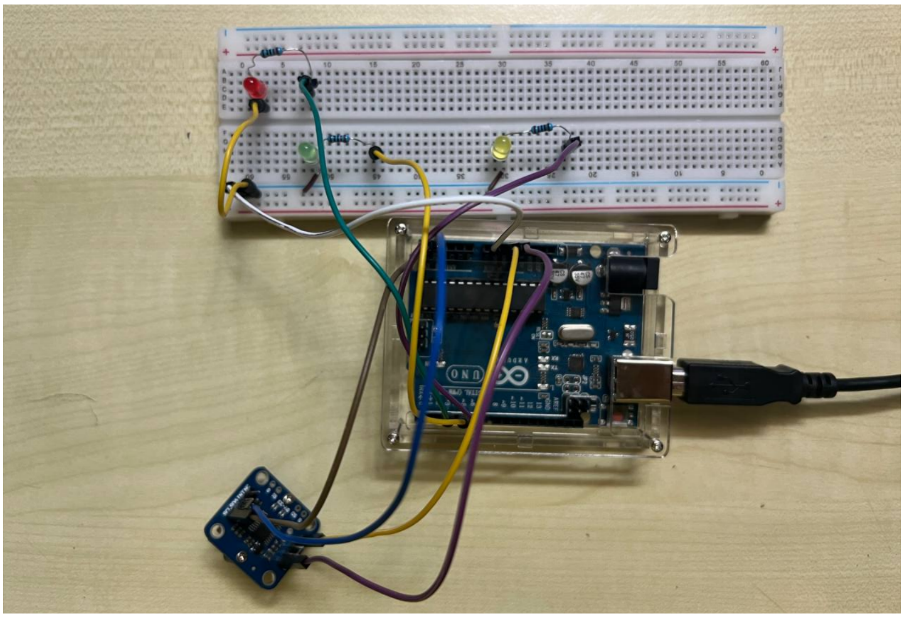
1. Color Sensor Module with Arduino (Code)
   https://www.youtube.com/watch?v=2Nbo8NRT2WE

2. Arduino Unit
   https://forum.arduino.cc/t/how-fast-does-arduino-process-each-line-of-code/6687

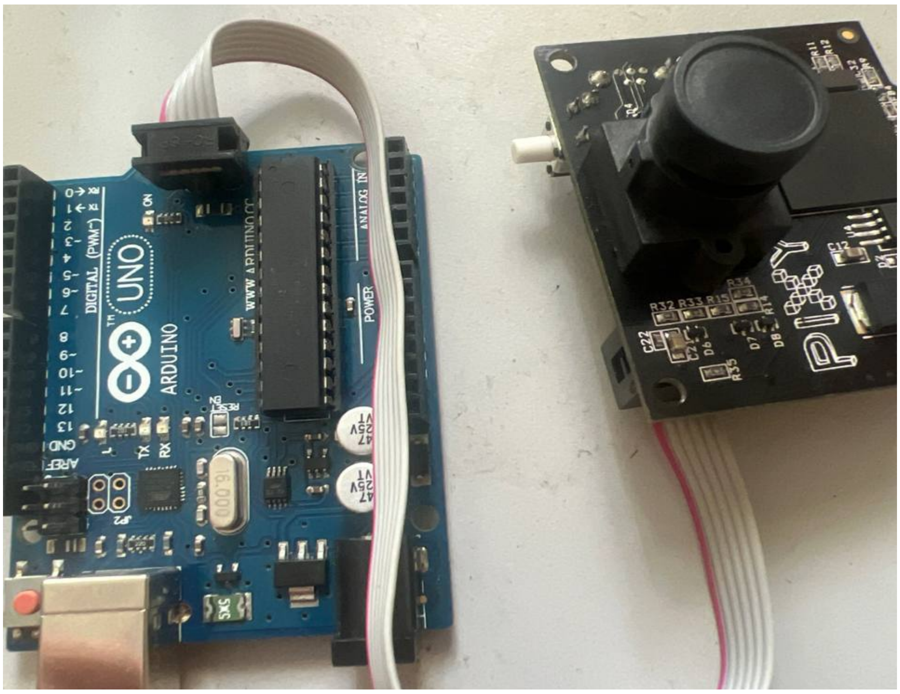
PDF:

Zulkifli. (2014). *Mechatronics Interfacing Lab Manual*, (Rev. ed.). Unpublished Class Materials

Appendices



Circuit for Color Detection with Color Sensor



Circuit for Color Detection with Pixy Camera

Acknowledgments

We would like to express our sincere gratitude to the individuals who provided invaluable assistance, guidance, and support during this experiment. First and foremost, we extend our appreciation to Dr. Wahju Sediono for their comprehensive instruction and mentorship throughout the experiment. Their insights, feedback, and enthusiasm played an important role in our understanding of Arduino programming.

Our fellow group members also deserve special acknowledgment for their collaboration and support. Our discussions, knowledge-sharing, and problem-solving sessions greatly enriched our understanding of this experiment's concepts and enhanced the overall learning experience. The collective contributions of our group members have not only enriched our learning experience but have also significantly contributed to the successful completion of this project.

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 acknowledgment, and that the original work contained herein has 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 **read** and **understand** the content of the report and no further improvement on the report is needed from any of the individual's contributions 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	**Sofea**	
Name	AISYAH SOFEA OTHMAN	**Read** ✅
Matric Number	2115386	**Understand** ✅
Contribution	- Experimental setup - Methodology - Material & Equipment	**Agree** ✅

Signature	**Aliaa**	
Name	SITI ALIAA BINTI IBRAHIM	**Read** ✅
Matric Number	2112618	**Understand** ✅
Contribution	- Recommendation - Discussion	**Agree** ✅

Signature	**Adli**	
Name	ADLI FAHMI	**Read** ✅
Matric Number	2113095	**Understand** ✅
Contribution	- Data Collection 9a - Data Analysis 9a - Abstract - Introduction	**Agree** ✅

Signature	**Ethar**	
Name	ETHAR OSMAN	**Read** ✅
Matric Number	2111282	**Understand** ✅
Contribution	- Conclusion - Result	**Agree** ✅

Signature	**ainaa**	
Name	NURAIN AINAA AQILAH BINTI ROSLI	**Read** ✅
Matric Number	2114560	**Understand** ✅
Contribution	-Data Collection Exp 9 -Data Analysis Exp 9 -Proofreading	**Agree** ✅