MCTA 3203 SYSTEM INTEGRATION LAB

WEEK 9:
Image/Video input interfacing with microcontroller and computer based system: Software and hardware.

SECTION : 1
GROUP NUMBER : 7 (G)

LECTURER’S NAME : DR ZULKIFLI BIN ZAINAL ABIDIN

GROUP MEMBERS :

NO :		COURSE	MATRIC
1.	MUHAMMAD NAQIB AIMAN BIN YUSNI	BMCT	2117765
2.	MUHAMAD AZRI BIN ABDUL AZIZ	BMCT	2113537
3.	MUHAMMAD HAFIDZUDDIN HANIF DANIAL BIN NORIZAL	BMCT	2123651
4.	SHAREEN ARAWIE BIN HISHAM	BMCT	2116943

TABLE OF CONTENTS

Abstract

The objective of this research is to combine colour sensors and computer vision technologies to learn how to recognize and react to different colours utilising microcontrollers. It looks into the design, implementation, and use of an interface that enables simple interaction between microcontrollers, computers, and input devices for images and videos. The study consists of two main parts. In the first, an RGB LED is controlled properly through recognizing the colours red, green, and blue using an Arduino board and a colour sensor. Next, using a Pixy Camera to identify these colours, the RGB LED is modified based on the identified colours. The study investigates how hardware components and software algorithms interact to rapidly recognize colours.

Introduction

For the objective of this experiment, we are aware that colour processing and detection are vital elements of numerous applications, such as robotics and industrial automation. This project aims to investigate possible interconnections between colour sensors and computer vision techniques in a microcontroller-based system. Using a TCS230 colour sensor, the experiment's beginning stage detects the primary colours red, green, and blue. After processing the sensor data, the Arduino board turns on an RGB LED to show the colour that was observed. In addition to the Pixy Camera, a computer vision tool has been released that can recognize colour signatures. With the help of a camera, this complex setup dynamically changes the RGB LED to improve colour detecting abilities. The experiment shows how varied colour detection techniques may be, as well as how useful they could be in fields like automation and robotics.

Material and Equipment

- Arduino board
- Colour sensor (TCS3200 or TCS34725)
- Jumper wires
- Breadboard
- RGB LED
- Computer with Arduino IDE and Python installed

- USB cable for Arduino

Circuit & Experimental Setup

Experimental Setup

1. Hardware Setup:

   •Connect the colour sensor to the Arduino using jumper wires. Refer to the sensor's datasheet and Arduino's pinout diagrams for guidance1.

   •If using an RGB LED, connect it to the Arduino for colour display.

2. Arduino Programming:

   •Write an Arduino sketch1 to interface with the colour sensor.

   •Read RGB colour data from the sensor and convert it to a format that can be sent to the computer.

   •Calibrate the sensor for accurate colour readings.

3. Testing and Data Collection:

   •Test the system with different coloured objects.

   •Collect data on the detected colours, their accuracy, and how the system performs in various lighting conditions.

   •Analyse the response time of the system when detecting colours.

   5. Analysis:

   •Evaluate the accuracy of colour detection by comparing detected colours with actual colours.

   •Analyse how the system performs in different lighting conditions.

   •Calculate the average response time for colour detection.

Results

In this experiment, we can see that the colour sensor has detected several colours and has shown different numeral values that will indicate the colour that it senses. From the experiment that we have done, the colour sensor has detected three colours red, green and blue depending on the colour we provided.

Discussion

Using a colour sensor module, the Colour Detection using Color Sensor experiment investigates the fundamentals of colour sensing technology. The objective is to comprehend how various colours can be identified and differentiated using colour sensors. The colour sensor is interfaced using Arduino, which then implements logic to understand the colour data it receives. The colour sensor needs to be calibrated in order for it to identify a baseline set of colours. The sensor offers different coloured things to the sensor and watches the output once it is safe to proceed. The limits of the colour sensor, including its sensitivity to ambient light and possible difficulties in differentiating similar vibrant colours with accuracy. The Color Detection with Color Sensor lab experiment provides hands-on experience in working with colour sensing technology.

By comparing the colours the colour sensor picked up with the real colours of the objects it was displayed, one may determine how accurate the colour detecting system is. We can measure the precision of the colour detection system and learn more about the operation of the colour sensor in various circumstances by methodically comparing the colours that are recognized with the actual colours. Understanding the system's limitations and fine-tuning it are dependent on the evaluation process.Secondly, since ambient light variations can have a significant effect on colour sensing accuracy, it's important to evaluate how well a colour detection system performs under various lighting circumstances. For an accurate result, it is important to carry out this experiment under ideal lighting conditions. Ideal Lighting Conditions: The colour sensor is most likely to function accurately in ideal lighting conditions, which are defined by consistent and evenly distributed illumination. This provides a starting point for comparison. We are unable to conduct this experiment in the presence of glare or shadows in addition to ideal lighting. Challenges can arise from objects' shadows and shiny surfaces' glare. The accuracy of identification may be affected by the colour sensor's interpretation of glare or shadows as colour shifts.

PART B: Colour Detection with Pixy Camera

PROCEDURE

Materials And Equipment

- Arduino
- Pixy camera
- Mini B USB Cable
- Power for arduino and pixy camera (laptop)

Experimental Setup

1. The IDC Pixy Cable is connected into Arduino's ICSP pins.
2. Mini B USB Cable is connected to Pixy.
3. Both Arduino and Pixy cables are connected to 2.0 USB Ports directly on the computer so that they are powered by USB's force.

Procedure For Pixy Camera

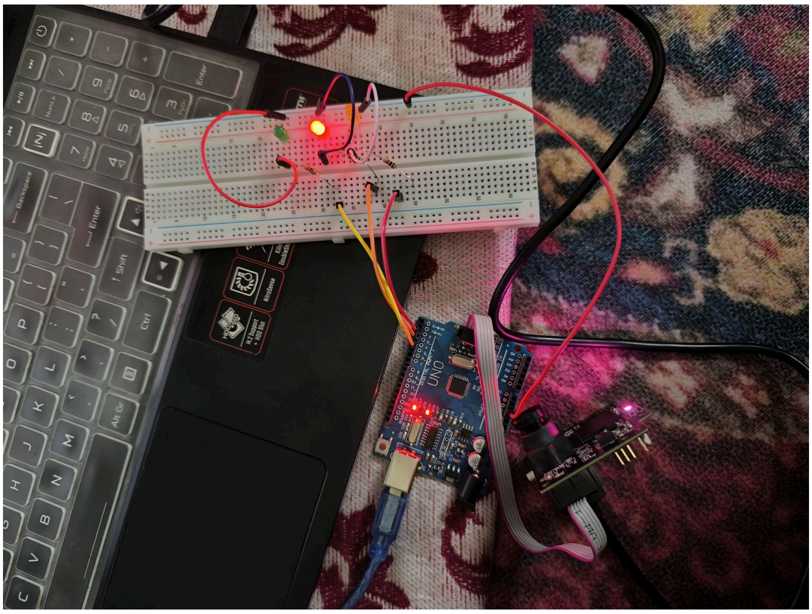
1. PixyMon is installed.
2. Pixy is configured in PixyMon :

- Icon Config > Pixy Parameters > Signature Labels and write down these three colors: green, blue, red > click ok.
- Config > Pixy Parameters tab > choose SPI Interface (Data out port: Arduino ICSP SPI; I2C address: 0x54; UART baud rate: 19200).

3. Teaching Pixy: Three objects are grabbed: one red, another yellow and another orange and they are positioned in front of the camera; make sure to get the best focus and light.
4. Go to Menu Action > Set Signature #1 > a region is selected from the green figure > the right mouse button is released making a rectangular shape inside it (now green must be shown on the screen)
5. The previous procedures are repeated for the blue and red (choose Set Signature #2 for blue & Set Signature #3 for red)
6. Finally go to Config > PixyMon Parameters (saved on computer) > General tab > Cooked render mode > boxes only; (to get rid of the pixels and emphasise the edges of the captured figure)
7. Go to Config > PixyMon Parameters (saved on Pixy) > Signature Tuning (adjust the all the ranges accordingly); (to make the edges of the captured figure clearer)

Procedure For Arduino Ide

1. Open Arduino IDE, Go to Sketch > Include Library > Add .ZIP library of pixy.
2. Arduino IDE is loaded, File > Examples > Pixy > hello_world (example) and maintain the baud rate to 9600 as the default value.
3. Arduino hello_world sketch with PixyMon is run together, Arduino's serial monitor is fired side-by-side with PixyMon. (Display Cooked/Processed/Video) button must be unable)



Coding

Arduino Coding:

```
#include <SPI.h>
#include <Pixy.h>

Pixy pixy;

const int blueLedPin = 2;    // Pin for red LED
const int redLedPin = 3;  // Pin for green LED
const int greenLedPin = 4;   // Pin for blue LED

void setup() {
  Serial.begin(9600);
  pixy.init();

  pinMode(blueLedPin, OUTPUT);
  pinMode(redLedPin, OUTPUT);
  pinMode(greenLedPin, OUTPUT);
}

void loop() {
  static int i = 0;
  int blocks = pixy.getBlocks();  // Use getBlocks method from ccc (Color Connected Components) object

  if (blocks) {
    // Turn off all LEDs initially
    digitalWrite(blueLedPin, LOW);
    digitalWrite(redLedPin, LOW);
    digitalWrite(greenLedPin, LOW);

    for (i = 0; i < blocks; i++) {
      Serial.print("Block ");
      Serial.print(i);
      Serial.print(": ");
      Serial.print("Signature: ");
      Serial.print(pixy.blocks[i].signature);  // Access signature using m_signature
      Serial.print(" X: ");
      Serial.print(pixy.blocks[i].x);  // Access x using m_x
      Serial.print(" Y: ");
      Serial.println(pixy.blocks[i].y);  // Access y using m_y
delay(100);
      // Add logic to identify and react based on the color signature
```

```
    if (pixy.blocks[i].signature == 1) {
     // Object with signature 1 detected (Color 1: Red)
     Serial.println("Color 1 (Blue) detected");
     digitalWrite(blueLedPin, HIGH);  // Turn on red LED
     // Add more actions for color 1 if needed
    } else if (pixy.blocks[i].signature == 2) {
     // Object with signature 2 detected (Color 2: Green)
     Serial.println("Color 2 (Red) detected");
     digitalWrite(redLedPin, HIGH);  // Turn on green LED
     // Add more actions for color 2 if needed
    } else if (pixy.blocks[i].signature == 3) {
     // Object with signature 3 detected (Color 3: Blue)
     Serial.println("Color 3 (Green) detected");
     digitalWrite(greenLedPin, HIGH);  // Turn on blue LED
     // Add more actions for color 3 if needed

    }
   }
  }
}
```

Result

After doing this lab, we can confirm that a pixy camera is a great hardware to detect colour. The pixy camera can detect all the colour that we show precisely as it should. When we show colour to the pixy camera we can see the numerical value change in the python interface that indicate the colour that we have shown.

Firstly, in the setup function, serial communication is initialised at a baud rate of 9600, and the Pixy camera is initialised. The pin modes for the LEDs are set to output to allow current to flow to the respective LEDs.

Next in the main loop, the program continuously checks for colour blocks detected by the Pixy camera. Initially, all blocks are detected. The program then processes each detected block, printing the information about the colour block including its signature and coordinates, to the serial monitor.

Based on the signature of the detected colour block, the corresponding LED is turned on: the blue LED for signature 1, the red LED for signature 2, and the green LED for signature 3. The code includes a small delay of about 100 ms to ensure the LEDs remain on long enough to be visually noticeable and to ease us to observe the changes of both the serial monitor and the LED lights. Hence, this setup allows the Arduino to react to different coloured objects detected by the Pixy camera by illuminating the appropriate LED.

Discussion

In this experiment, the pixy camera properly detects and recognizes various colours. Green, blue, and red are the three colours that are visible. Our observations support the accuracy of the findings. Depending on the colour that the pixy camera detects, the appropriate output is shown on the serial monitor of the Arduino IDE. Nearly immediate colour detection and output display on the serial monitor are achieved.

To ensure accurate recognition, we adjusted the range of colour patterns in the PixyMon software setup. The range of red and green is extended in this instance, but the range of blue is diminished. This is because the pixy camera's ambient environment has a blue colour.

The following outcomes were observed during testing under various illumination conditions: In bright light, normal lighting, and lowered lighting, the colours green, red, and blue are visible. However, the pixy camera was unable to identify any colour in a dark room. This demonstrates the Pixy camera's quality and the system's adaptability, which make it appropriate for a variety of uses.

Conclusion

In conclusion, this experiment was a success in demonstrating the function of a colour sensor in the pixy camera.This shows by the fact that the pixy camera can detect three different colours correctly.After all the wiring is correct the system has shown a great data output. These experiments demonstrate the practical application of Python in interfacing with hardware components for control purposes. Both experiments highlight the usefulness of a pixy camera.Overall, the pixy camera and colour sensor have shown that it is a good hardware to use for sensing colour.

Student's Declaration

Declaration:
We certify that this project/assignment is entirely our own work, except where we have given fully documented references to the work of others, and that the material in this assignment has not previously been submitted for assessment in any formal course of study.

azri

____

Name: Muhamad Azri bin Abdul Aziz

hafidzuddin

____

Name: Muhammad Hafidzuddin Hanif Danial bin Norizal

arawie

____

Name: Shareen Arawie bin Hisham

naqib

____

Name: Muhammad Naqib Aiman bin Yusni

Date: 21/5/2024