Mechatronics System Integration (MCTA3203)

Week **9**: Image/Video input interfacing with microcontroller (ver. 2a).

Color Detection and Analysis

Objective:
To design a color detection system using Arduino, Python, and either a color sensor or an AI Vision camera. The experiment will involve hardware setup, programming Arduino and Python, and analyzing the accuracy and performance of color detection in different scenarios.

Part 1: Color Detection with Color Sensor

Materials Needed:

- Arduino board
- Color sensor (e.g., TCS3200)
- Jumper wires
- Breadboard
- RGB LED (optional)
- Computer with Arduino IDE and Python installed
- USB cable for Arduino

Experiment Steps:

1. Hardware Setup:

- Connect the color sensor to the Arduino using jumper wires. Refer to the sensor's datasheet and Arduino's pinout diagrams for guidance.
- If using an RGB LED, connect it to the Arduino for color display.

2. Arduino Programming:

- Write an Arduino sketch to interface with the color sensor.
- Read RGB color data from the sensor and convert it to a format that can be sent to the computer.
- Calibrate the sensor for accurate color readings.

3. Python Programming:

- Write a Python program to communicate with the Arduino over the serial connection using the *pyserial* library.
- Receive RGB color data from the Arduino.
- Interpret the received data to determine the detected color. You can modify the code shown below for your own purposes.

4. Testing and Data Collection:

- Test the system with different colored objects.

- Collect data on the detected colors, their accuracy, and how the system performs in various lighting conditions.
- Analyze the *response time* of the system when detecting colors.

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

Part 2: Color Detection Using Gravity HuskyLens AI Vision Camera

To build a system using an Arduino Uno and the HuskyLens AI Camera to detect three different colored objects, follow these step-by-step instructions:

Step 1: Hardware Setup

Connect HuskyLens to Arduino Uno:

Use UART interface for simplicity. Connect the pins as follows:

HuskyLens Pin	Arduino Uno Pin
GND	GND
VCC	5V
TX	D4 (SoftwareSerial RX)
RX	D5 (SoftwareSerial TX)

Note: You will need to use **SoftwareSerial** on the Arduino Uno because it has only one hardware serial port.

Power Supply:

• Power Arduino Uno via USB or external power.
• HuskyLens can be powered from Arduino's 5V pin.

Ensure that the Arduino Uno is powered either via USB or an external power supply.

Step 2: Color Detection Calibration

Use HuskyLens' Built-in Color Recognition:

1. Power on the HuskyLens.
2. Use the onboard buttons to switch to "Color Recognition" mode.
3. Point it at the object with the color you want to detect.
4. Long-press the learning button to teach the camera the color.
5. Repeat for 3 different colors—each will be stored as a learned ID.

The HuskyLens display will show bounding boxes and color labels once objects are recognized.

Step 3: Install Required Arduino Library

1. Open Arduino IDE.
2. Go to Library Manager (Sketch → Include Library → Manage Libraries…).
3. Search for "DFRobot_HuskyLens" and install it.

Step 4: Write Arduino Code

Here is a basic example using SoftwareSerial and the HuskyLens library:

```
#include <SoftwareSerial.h>
#include "HUSKYLENS.h"

HUSKYLENS huskylens;
SoftwareSerial mySerial(4, 5); // RX, TX

void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  while (!huskylens.begin(mySerial)) {
    Serial.println("HuskyLens not connected!");
    delay(1000);
  }
  Serial.println("HuskyLens Ready.");
}

void loop() {
  if (!huskylens.request()) {
    Serial.println("Request failed");
    return;
  }

  if (huskylens.available()) {
    HUSKYLENSResult result = huskylens.read();
    Serial.print("Detected Color ID: ");
    Serial.print(result.ID);
    Serial.print(" at (X: ");
    Serial.print(result.x);
    Serial.print(", Y: ");
    Serial.println(result.y);

    // Add actions based on detected color ID
    if (result.ID == 1) {
      // Color 1 detected
    } else if (result.ID == 2) {
      // Color 2 detected
    } else if (result.ID == 3) {
      // Color 3 detected
    }
  }
}
```

Step 5: Upload and Test

Upload Code:

- Connect Arduino Uno to your computer.
- Select Board: Arduino Uno.
- Select Port: The correct COM port.

- Click Upload.

Open Serial Monitor:

- Open the Serial Monitor to see color ID and coordinates.
- Try showing different colors and observe detection accuracy.

Step 6: Adjustments and Improvements

- Recalibrate colors under consistent lighting for best results.
- Use result.width and result.height to analyze object size if needed.
- Add LEDs or actuators controlled based on color detection results.

Task

- Summarize Key Findings:
  Evaluate the results of your color detection experiments with the color sensor and Gravity HuskyLens AI Vision Camera. Discuss which method yielded the most accurate and consistent results. Consider factors such as calibration, lighting conditions, and response time.
- Identify Challenges and Improvements:
  Reflect on any challenges encountered during the experiment, such as sensor calibration or handling varying lighting. Propose potential improvements, such as better calibration techniques, sensor fusion, or real-time processing optimizations.
- Insights for Future Iterations:
  Based on your findings, suggest improvements for future iterations. Consider implementing more advanced algorithms, exploring different sensor configurations, or using machine learning models to enhance color detection accuracy and speed.

Useful Links

[1] https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336
Gravity: HUSKYLENS AI Machine Vision Sensor - DFRobot Wiki

[2] https://www.youtube.com/watch?v=HZxFj2u9k00
HuskyLens Tutorial - Getting Started with Husky Lens - AI Vision Sensor

[3] https://www.dfrobot.com/product-1922.html
Gravity: HuskyLens K210 AI Camera (No-Code Vision Sensor for STEM, Arduino & micro:bit)

[4] https://howtomechatronics.com/tutorials/arduino/arduino-color-sensing-tutorial-tcs230-tcs3200-color-sensor/

[5] https://www.youtube.com/watch?v=CPUXxuyd9xw

[6] https://www.youtube.com/watch?v=g3i51hdfLaw