Experiment 01

Fruit Ripeness monitor

In this experiment, we will try to recognize the color of a fruit and then determine its ripeness. The user is notified when the fruit ripens.

Background

The APDS9960 breakout board is capable of RGB color sensing, rudimentary gesture sensing, proximity sensing, and ambient light sensing. With the provided library, a microcontroller connected to the breakout board can identify the intensity of red, blue, and green colors along with clear light. It can determine simple hand gestures like moving left-right, up-down, and vice-versa. It can also provide how close an object is from the sensor within a few centimeters range.

The APDS9960 breakout board has an inbuilt voltage regulator which enables it to be connected to both a 3.3V or a 5V supply. The sensor supports I2C communication. The 8-bit proximity resolution engine can reject ambient light and can produce accurate position values within the detectable range. The sensor employs four directional photodiodes to detect the reflected IR light from the IR LED driver. The change in the reflected light at these four positions is used to estimate gestures. The provided library enables users to program simple gestures directly, but detection of more complicated gesture combinations can be programmed. The Ambient Color Sense (ALS) and Color Sense (RGBC) engines have built-in UV and IR filters to block spectrum noise. Due to its high sensitivity, the sensor can be ideally deployed under dark glass such as mobile phone screens.

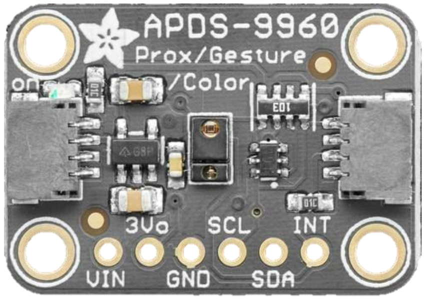
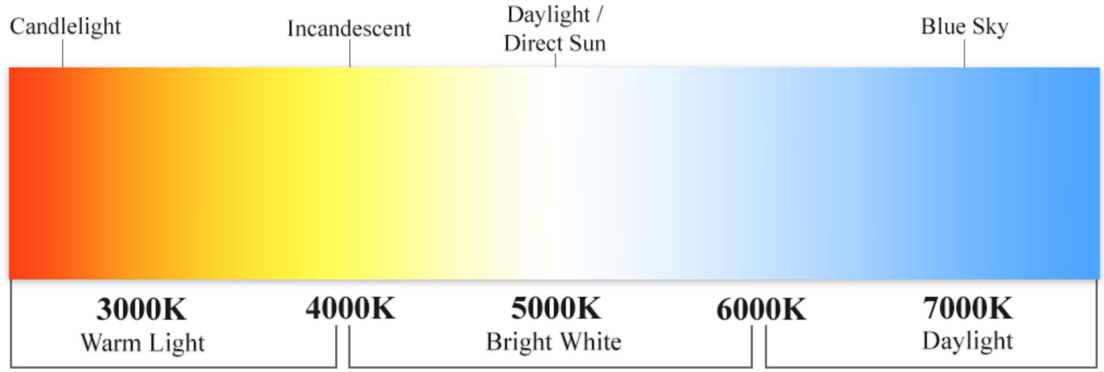


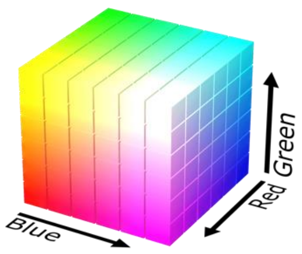
Fig 1. The RGB sensor module has an LED driver and a photodetector. After the light is emitted by the LED, the photodetector captures the reflected light bouncing off from the object. This reflected light enables the module to generate a signal estimating the color.

The sensor has user-configurable interrupt signals that can be programmed to send a signal by setting threshold values based on proximity or either of the R, G, B, and C channels. These triggers can be used to automate response generation without additional overhead on the microcontroller.

Experiment Set-up: Configuration

For the experiment, students are given sample images of fruits at different stages of the life cycle. Fruits such as Bananas show distinct color variations as they ripen. Multiple images of such fruits at various stages of the lifecycle will be given to the students. Students





are required to detect the color of the fruit and create a logic table corresponding to each color at the given age in the lifecycle of a fruit. The logic table can be used later to match the color of the fruit that needs to be detected for its life stage expectation.

The sensor outputs color temperature, and luminance values of R G and B channels. Color temperature defines how the whites of your color look. It is measured in Kelvin. Warmer temperature turns yellow and as you go cooler the whites turn blue. Daylight is roughly in between these two extremes. The sensor also outputs illuminance values in lux. Illuminance tells you the intensity of the light. With the combination of the individual values of R, G, and B we can determine the color. For example, if the RGB value says 255,192,203, it represents the color pink.

For the experiment, students can use an outlining tool linked in the references section. You can enter the RGB values into the tool and it will show the associated color with the RGB code. Once the color is obtained, match it with the table created earlier to determine the ripeness of the fruit.

Instructions

1. Download the Zip file of the library from the course materials "APDS-9960_RGB_and_Gesture_Sensor_Arduino_Library"
2. Now extract the zip file.
3. Copy the Adafruit_APDS9960 library folder to the default Arduino library folder (refer to the 3.2.3 section in the manual). Now open Arduino and click on “File” from the Menu bar. Click on “Open” to open the “color.ino” file. The “color.ino” file can be found in the course material for this module. The IDE now loads the program to determine color. Flash the code to the Node MCU. Refer to the IoT Board Manual for flashing instructions.
4. Open the Serial Monitor from the dropdown menu under “Tools” on the Menu bar. Refer to the IoT Board Manual for setting up the Serial Monitor window. In the serial monitor, the active color readings of R, G, B, and C channels are displayed. Copy the values for the reference pictures of fruits.
5. The reference values are to be noted down. Create a table with the reference values. This table will be used to compare the values that are obtained for the random fruit images provided in the test folder.
6. Based on the comparison, determine a rough estimate for the fruit and label the image accordingly for submission.

☞ Deliverables

Demonstration:

1. Record a video demonstration explaining the outcome of the experiment. Refer to the title page for a brief description of the expected outcome. Make sure you talk over all observations and the video is presentable. Also, don't forget to show the data updating in real time on the serial monitor. Describe how would you develop a comparison algorithm to estimate the fruit ripeness. Can you think on ways to integrate machine learning algorithms?

References and Further Reading

[1] https://www.adafruit.com/product/3595
[2] https://cdn.sparkfun.com/assets/learn_tutorials/3/2/1/Avago-APDS-9960-datasheet.pdf
[3] https://github.com/adafruit/Adafruit_APDS9960
[4] https://github.com/adafruit/Adafruit_CircuitPython_APDS9960
[5] https://www.w3schools.com/colors/colors_rgb.asp
[6] https://www.rapidtables.com/web/color/RGB_Color.html