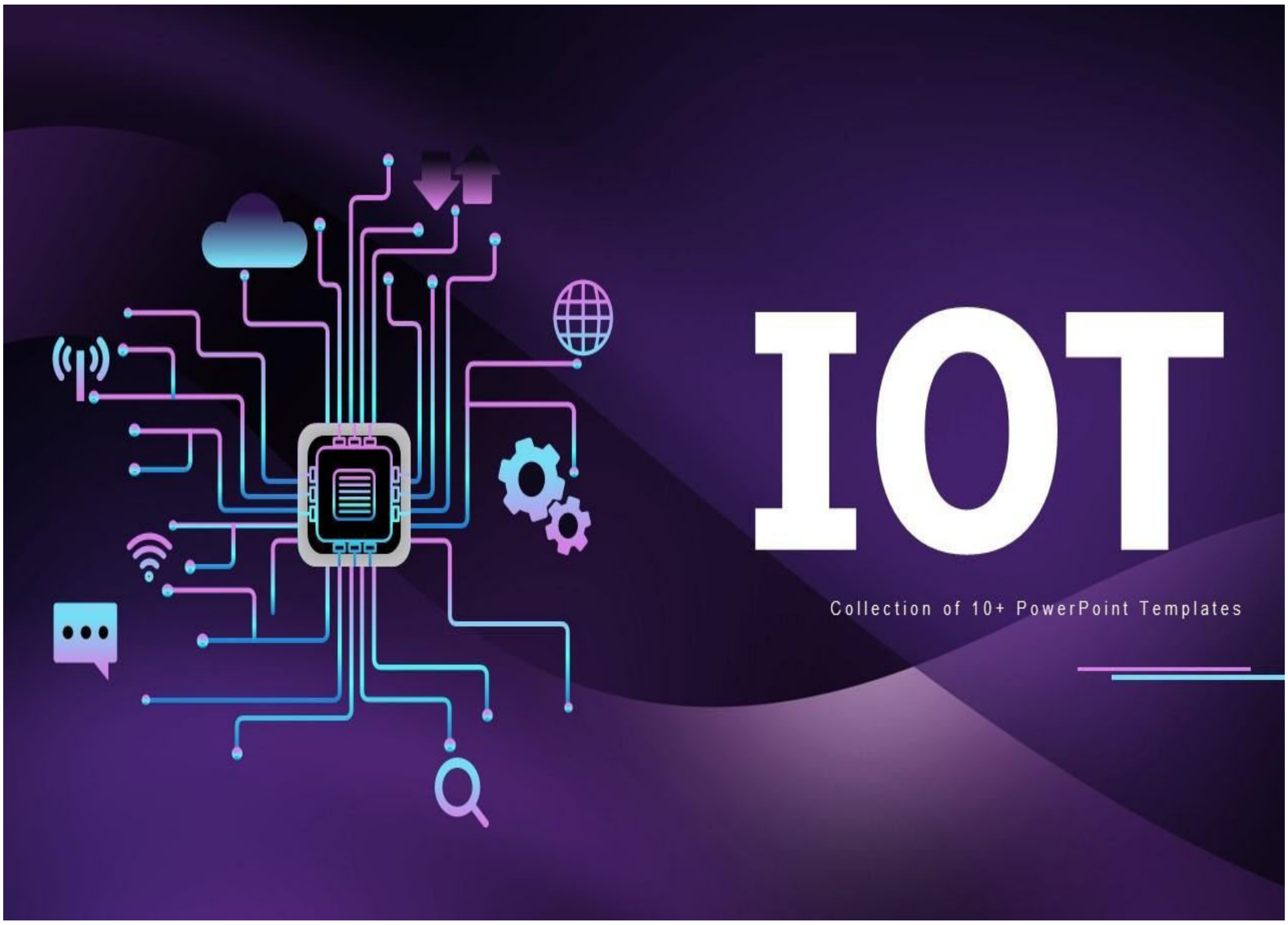
INTERFACING RUGGED BOARD WITH TWO COLOR SENSOR



Presented by
K. Prem kumar
SBCS indian pvt ltd

Table of Contents

1.0 Introduction

The KY-011 two-colour sensor is a compact electronic module designed for colour detection and recognition in a variety of electronic projects. Comprising two key components—a red-sensitive photodiode and a green-sensitive photodiode—the sensor operates on the fundamental principle of colour differentiation based on light absorption and reflection.

At its core, the KY-011 utilises the unique properties of materials to selectively absorb and reflect light at specific wavelengths. The red photodiode primarily responds to light in the red spectrum, typically ranging from 600 to 750 nanometers, while the green photodiode is sensitive to light in the green spectrum, generally between 500 and 600 nanometers.

The sensor's working mechanism involves exposing the object of interest to light, and the photodiodes generate analog voltage signals corresponding to the intensity of red and green light detected. These signals are then processed by a connected microcontroller, often an Arduino, which interprets the colour information.

One noteworthy feature of the KY-011 sensor is its potential for sensitivity adjustment. Some versions of the module include a potentiometer, allowing users to fine-tune the sensor's responsiveness to different lighting conditions.

The analog output from the KY-011 sensor lends itself to straightforward integration with microcontrollers, enabling the implementation of colour recognition algorithms. These algorithms can define specific thresholds for red and green values, facilitating the identification of distinct colours based on the sensor's readings.While the KY-011 sensor provides a cost-effective solution for basic colour detection, it is important to consider its limitations. The sensor is optimised for red and green detection and may not cover the entire visible spectrum. Additionally, ambient light conditions can impact its accuracy, necessitating thoughtful consideration during application design.

2.0 Hardware Components

In this project we are mainly used two components those are:

1. KY-011 Two color sensor

2. Rugged board

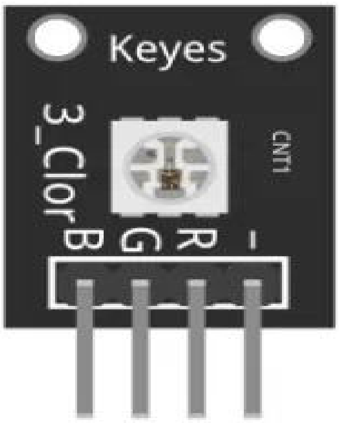
2.1 KY-011 Two color Sensor

This module consist of a common cathode 3mm red/green LED, a 0Ω resistor, and 3 male header pins. Since the operating voltage is between 2.0v and 2.5v, you’ll have to use limiting resistors to prevent burnout when connecting to the Arduino/STM32.

Operating Voltage	2.0v ~ 2.5v
Working Current	10mA
Color	Red + Blue
Beam Angle	150

Connection Diagram

Connect the Red pin (R) on the board to Pin 10 on the Arduino, connect the Blue pin (B) to pin 11. Lastly, connect the ground pin (Y) to GND.We’ll use a couple of resistors between the board and the STM32 to prevent burning the LED.



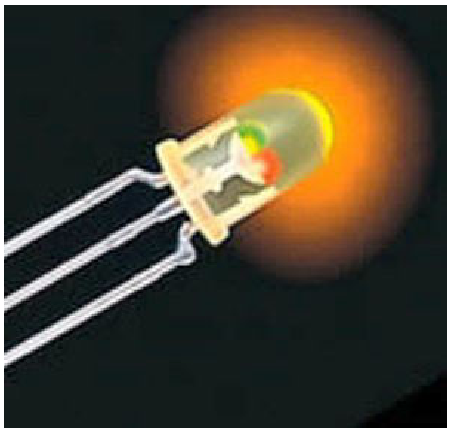


Fig: Two color sensor **Fig:** Internal led in the sensor

2.2 Rugged Board

RuggedBoard is an Open source Industrial single board computer powered by ARM Cortex-A5 SOC @500 MHz, implemented with the finest platform for rapid prototyping. The usage of System On Module over a System On Chip is the most rapid way to achieve time to market,curtail development risks for product quantities ranging from a few hundred to thousands.

RuggedBoard- A5D2x consists of Multiple Interfaces such as Ethernet,RS232, CAN, RS485, Digital Input and Digital Output with optically isolated, Standard MikroBus header for Add-On Sensors, Actuators and Multiple Wireless Modules such as ZigBee, LoRa,Bluetooth etc. mPCIeconnector with USB interface used for Cloud Connectivity modules 3G,4G, NB-IoT, WiFi. Expansion header with GPIO, UART, I2C, SPI,PWR etc.

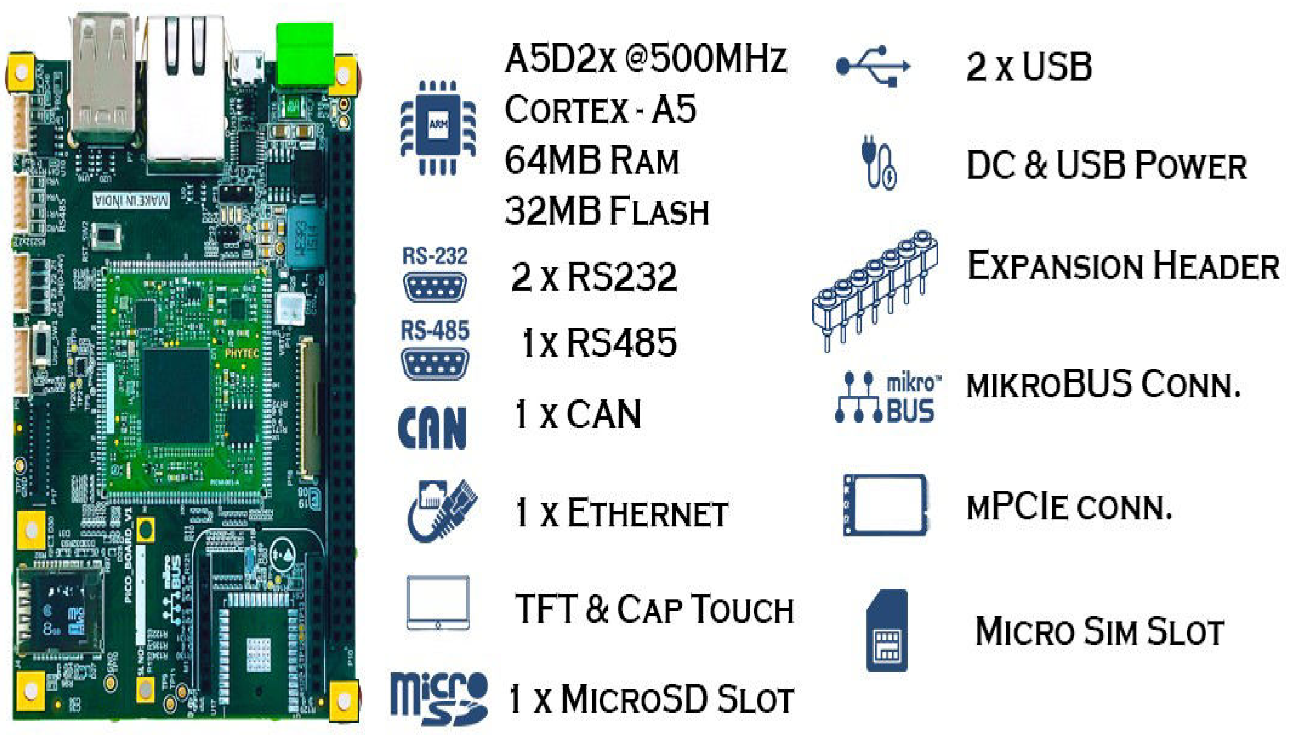


Fig: RuggedBoard-A5D2x

3.0 SOFTWARE COMPONENTS

3.1 MINICOM

Minicom is a text-based serial communication program that is commonly used to connect to and communicate with devices over a serial port. It is often used 15 for debugging and configuring devices, especially in embedded systems and projects involving microcontrollers or other hardware components. Below is an explanation of how minicom can be used in a project.

1. Installation:

• Before using minicom, you need to install it on your system. You can typically install it using your system's package manager.

• bash

• sudo apt-get install minicom

2. Connecting to a Serial Port:

• Minicom is primarily used for serial communication, so you need to be connect it to the serial port of the device you want to communicate with.Use the following command to open minicom:

• bash

• minicom -D /dev/ttyUSB0

• Here, /dev/ttyUSB0 is the path to the serial port. The actual port may vary depending on your system and the connected device.

3. Configuration:

• Once minicom is open, you may need to configure the serial port settings such as baud rate, data bits, stop bits, and parity. This is often necessary to match the settings of the device you are communicating with. You can access the configuration menu by pressing Ctrl-A followed by Z.

4. Interacting with the Device:

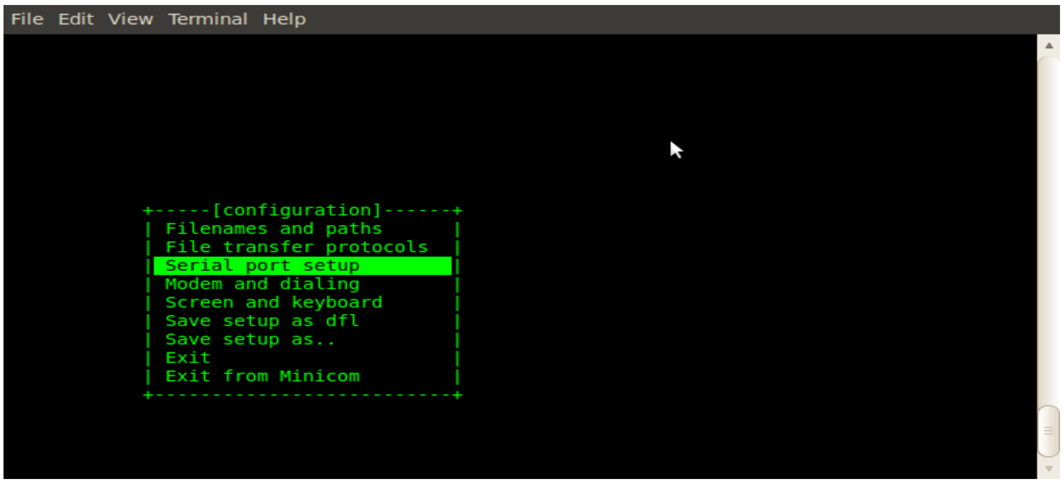
• After configuring the serial port, you can interact with the device. Minicom allows you to send commands and receive responses. This is particularly useful for debugging purposes and for configuring devices that have a serial console.

5. Exiting Minicom:

• To exit minicom, you can use the Ctrl-A followed by X shortcut.

6. File Transfer:

• Minicom also supports file transfer using protocols like Xmodem or Ymodem. This can be useful for updating firmware or transferring files between your computer and the connected device.



```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

Fig: Minicom Window

4.0 Source code

Hear mentioning two types of code for ky-011 sensor

4.1 Sys code :-

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <unistd.h>

// GPIO export paths
char export_path[100] = "/sys/class/gpio/export";
char direction_path[100] = "/sys/class/gpio/";
char value_path[100] = "/sys/class/gpio/";

// Function to set up GPIO for a specific LED
void setup_gpio(int pin_offset, const char *group) {
    int fd;

    // Export the GPIO pin
    fd = open(export_path, O_WRONLY);
    perror("open");
    char gpio_pin[4];
    sprintf(gpio_pin, "%d", pin_offset);
    write(fd, gpio_pin, strlen(gpio_pin));
    perror("write");
    close(fd);

    // Construct the direction and value paths
        sprintf(direction_path + 16, "%s%d%s", group,
pin_offset, "/direction");
     sprintf(value_path + 16, "%s%d%s", group, pin_offset,
"/value");

    // Open the direction file and set it to "out"
    fd = open(direction_path, O_WRONLY);
    perror("open");
    write(fd, "out", 4);
    perror("write");
    close(fd);
}
```

```
int main() {
    int red_pin = 31;  // GPIO pin for the red LED
    int blue_pin = 13; // GPIO pin for the blue LED

    // Set up GPIO for the red LED
    setup_gpio(red_pin, "PA");

    // Set up GPIO for the blue LED
    setup_gpio(blue_pin, "PA");

    int red_fd, blue_fd;

    while (1) {
        // Open the value file for the red LED and toggle it
        red_fd = open(value_path, O_WRONLY);
        perror("open");
        write(red_fd, "1", 2);
        perror("write");
        sleep(1);
        write(red_fd, "0", 2);
        perror("write");        close(red_fd);

        sleep(1); // Introduce a delay between the colors

        // Open the value file for the blue LED and toggle it
        blue_fd = open(value_path, O_WRONLY);
        perror("open");
        write(blue_fd, "1", 2);
        perror("write");
        sleep(1);
        write(blue_fd, "0", 2);
        perror("write");
        close(blue_fd);
        sleep(1); // Introduce a delay between the colors
    }
    return 0;
}
```

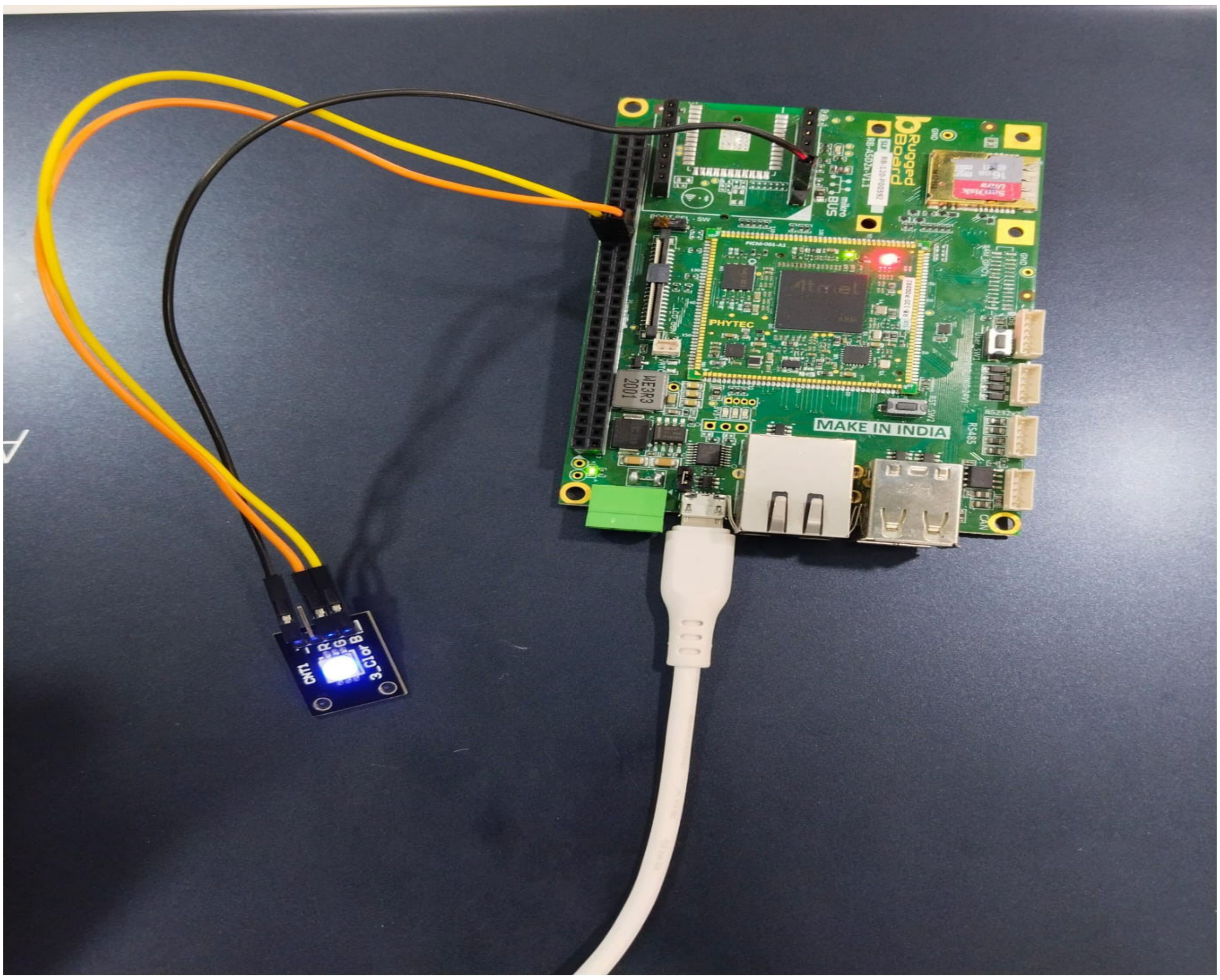
4.2. MRAA code :-

```
#include <stdio.h>
#include <unistd.h>
#include <mraa.h>

// Function to set up GPIO for a specific LED
void setup_gpio(int pin_offset, mraa_gpio_context gpio) {
   // Export the GPIO pin
   mraa_gpio_dir(gpio, MRAA_GPIO_OUT);
}
int main() {
   int red_pin = 31;  // GPIO pin for the red LED
   int blue_pin = 13; // GPIO pin for the blue LED

   mraa_init();  // Initialize MRAA

   // Set up GPIO for the red LED
   mraa_gpio_context red_gpio = mraa_gpio_init(red_pin);
   if (red_gpio == NULL) {
```

```
            fprintf(stderr, "Failed to initialize GPIO for the red
LED\n");
        return 1;
    }
    setup_gpio(red_pin, red_gpio);

    // Set up GPIO for the blue LED
                        mraa_gpio_context      blue_gpio      =
mraa_gpio_init(blue_pin);
    if (blue_gpio == NULL) {
           fprintf(stderr, "Failed to initialize GPIO for the blue
LED\n");
        return 1;
    }
    setup_gpio(blue_pin, blue_gpio);
    while (1) {
        // Toggle the red LED
        mraa_gpio_write(red_gpio, 1);
        sleep(1);
        mraa_gpio_write(red_gpio, 0);
        sleep(1);

        // Introduce a delay between the colors
        sleep(1);

        // Toggle the blue LED
        mraa_gpio_write(blue_gpio, 1);
        sleep(1);
        mraa_gpio_write(blue_gpio, 0);
        sleep(1);
        // Introduce a delay between the colors
        sleep(1);
    }

    // Clean up
    mraa_gpio_close(red_gpio);
    mraa_gpio_close(blue_gpio);
    return 0;
}
```

5.0 Expected output

The expected output for the ky-011 two color sensor by using mraa code and sys code is shown below :



6.0 Real time applications

The KY-011 two-color sensor, also known as the RGB module, is a simple module that consists of a red and green LED. While it might not be as sophisticated as more advanced color sensors, Here are some potential real-time applications for the KY-011 two-color sensor:

1. **Color Sorting Systems:**

Implementing a small-scale color sorting system for objects based on their color. The sensor can detect the color of the objects and trigger sorting mechanisms accordingly.

2. **Traffic Light Simulation:**

Simulating a basic traffic light system for educational purposes. The red and green LEDs on the module can represent the different states of a traffic light.

3. **Ambient Light Adjustment:**

Adjusting the intensity or color temperature of ambient lighting in a room based on the detected color. For example, creating mood lighting that changes based on the environment.

4. Plant Growth Monitoring:

Monitoring the light conditions for plants in a garden or indoor setting. The sensor can be used to adjust artificial lighting for optimal plant growth.

5. Interactive Art Installations:

Incorporating the sensor into interactive art installations where the color or brightness of the artwork changes based on environmental conditions or user interaction.

6. Color-based Alarm System:

Creating a simple alarm system that triggers different responses based on the color detected. For instance, red might indicate a critical condition, while green could signify normal operation.

7. Color Identification for Educational Purposes:

Building educational projects for teaching color identification to children. The sensor can be part of a hands-on learning experience.

8. Feedback for Accessibility Devices:

Integrating the sensor into accessibility devices to provide feedback to users with visual impairments. For example, different colors could represent different states or conditions.

9. Game Development:

Incorporating the sensor into interactive games where the detected color influences the gameplay or visual effects.

10. Quality Control in Manufacturing:

Implementing a basic quality control system on a production line where the color of items is checked, and appropriate actions are taken based on the results.

7.0 REFFERENCES

1. https://arduinomodules.info/ky-009-rgb-full-color-led-smd-module/

2. https://forum.arduino.cc/t/multiple-color-recognition-sensors-on-one-arduino/279572

3. https://arduinomodules.info/ky-011-two-color-led-module-3mm/

8.0 CONCLUSION

In conclusion, the KY-011 two-color sensor, with its dual LED configuration and photodiode for color detection, offers a simple and cost-effective solution for basic color sensing applications. While its features are limited compared to more advanced color sensors, it finds practical use in educational settings, prototyping, and DIY projects. The sensor's ability to detect and differentiate between red and green light, coupled with its ease of integration, makes it a popular choice for beginners and hobbyists exploring fundamental concepts of color mixing and light intensity measurement.