Adafruit Flora Color Sensor Library Documentation

EE2361 Final Project

University Of Minnesota ECE Department

April 29, 2024

By: Kirnesh Kaushik, Sam O’Connor, Emily Schaefer

Group 25

Introduction

This library includes functions that allow use of the Adafruit Flora color sensor with the Microchip PIC24FJ64GA002 microcontroller.

First the sensor is initialized with the color_init() function and the device ID is read from the Flora color sensor using the read_device_id() function. The read_color_[color]() function takes the data from the internal registers on the Flora color sensor which correspond to clear-red-green-blue color values. The values from the registers are then put into a circular buffer with the put[color]() functions. The values are then averaged to get the clear-red-green-blue hues detected by the sensor using avg[color]() and then get_color_avg(). Finally the get_color_avg() values are turned into R/B and R/G averages which are then compared to the predetermined ratios for the corresponding colors in the identify_color() function which returns the sensed color.

Due to the sensitivity of the sensor, the ranges for colors in the identify_color() function will need to be calibrated to your specifications. The values in this library were set for a color sensor that is placed about 11/16 of an inch away from the object to identify.

Hardware Description

Microchip - PIC24FJ64GA002 - https://www.microchip.com/en-us/product/pic24fj64ga002#
Flora Color Sensor - TCS34725 - https://www.adafruit.com/product/1356
LCD - AQM0802A-RN-GBW - https://akizukidenshi.com/catalog/g/g106669/
MPLAB SNAP- PG164100 - https://www.microchip.com/en-us/development-tool/pg164100
Colorful 12mm Square Tactile Button Switch - B3F - https://www.adafruit.com/product/1010

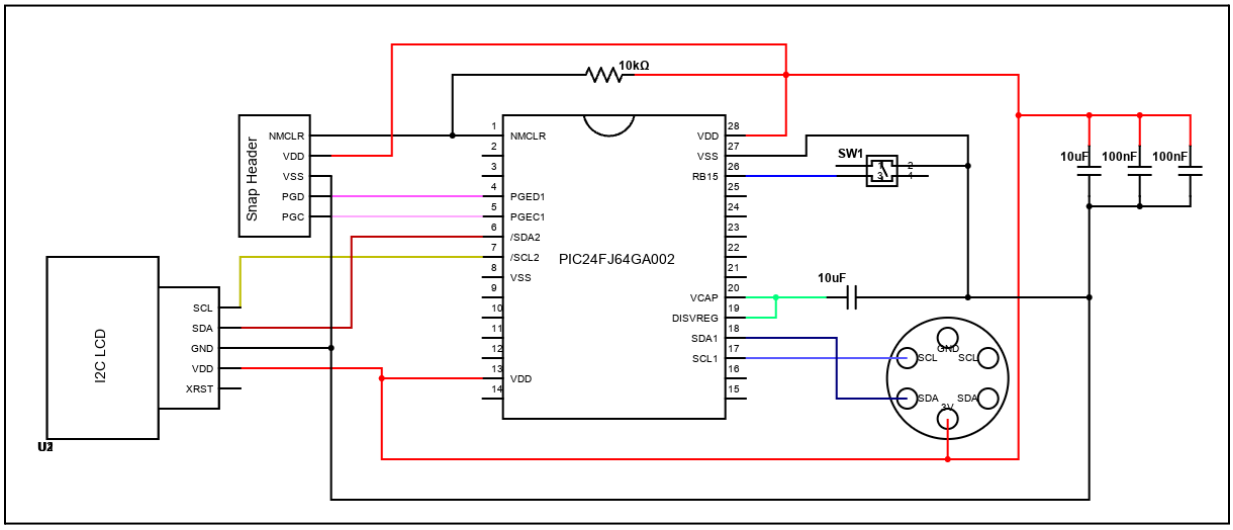

Figure 1: Mastermind game circuit schematic with Flora Color Sensor, I2C LED, button, SNAP, and PIC24FJ64GA002

Full Documentation

Color Sensor Functions

`void read_color_clear(void):`
Function Desc.: Reads 2 bytes of clear value from two data registers inside the sensor using an I2C combined read format and a command code.
Arguments: N/A
Peripherals: Flora color sensor

`void read_color_red(void):`
Function Desc.: Reads 2 bytes of red value from two data registers inside the sensor using an I2C combined read format and a command code.
Arguments: N/A
Peripherals: Flora color sensor

`void read_color_green(void):`
Function Desc.: Reads 2 bytes of green value from two data registers inside the sensor using an I2C combined read format and a command code.
Arguments: N/A
Peripherals: Flora color sensor

`void read_color_blue(void):`
Function Desc.: Reads 2 bytes of blue value from two data registers inside the sensor using an I2C combined read format and a command code.
Arguments: N/A
Peripherals: Flora color sensor

`void i2c_write_byte(char device_address, char register_address, char data);`
Function Desc.: Writes a byte to the flora color sensor using an I2C write protocol.
Arguments: “device_address” gives the address of the slave along with a write bit. “Register_address” gives the address of the register to be written to, and “data” gives the data to be written to the register.
Peripherals: Flora color sensor

`char read_device_id(void):`
Function Desc.: Reads the 1 byte device ID from the ID register inside the sensor using an I2C combined read format and a command code.
Arguments: N/A
Peripherals: Flora color sensor

`void color_init():`
Function Desc.: Initializes the flora color sensor to enable on start-up, have a baud rate of 100 kHz, connect to SDA and SCL 1 on PIC24, have maximum sensitivity, and a gain of 1.
Arguments: N/A
Peripherals: Flora color sensor

Color Averaging Functions

`void initBuffer():`
Function Desc.: Initializes the clear, red, blue, and green buffers for the moving average by setting them all to 0.
Arguments: N/A
Peripherals: N/A

`void putClear(int clear):`
Function Desc.: Puts the 16 bit clear value into the clear value circular buffer
Arguments: 16 bit clear value taken from sensor
Peripherals: N/A

`void putRed(int red):`
Function Desc.: Puts the 16 bit red value into the red value circular buffer
Arguments: 16 bit red value taken from sensor
Peripherals: N/A

`void putGreen(int green):`
Function Desc.: Puts the 16 bit green value into the green value circular buffer
Arguments: 16 bit green value taken from sensor
Peripherals: N/A

`void putBlue(int blue):`
Function Desc.: Puts the 16 bit blue value into the blue value circular buffer
Arguments: 16 bit blue value taken from sensor
Peripherals: N/A

`unsigned int clearAvg():`
Function Desc.: Loops through the entire clear value circular buffer and averages all the values
Arguments: N/A
Peripherals: N/A

`unsigned int redAvg():`
Function Desc.: Loops through the entire red value circular buffer and averages all the values
Arguments: N/A
Peripherals: N/A

`unsigned int greenAvg():`
Function Desc.: Loops through the entire green value circular buffer and averages all the values
Arguments: N/A
Peripherals: N/A

`unsigned int blueAvg():`
Function Desc.: Loops through the entire blue value circular buffer and averages all the values
Arguments: N/A
Peripherals: N/A

`void get_color_avg(void):`
Function Desc.: Reads all colors , turns them into 16-bit numbers, and puts them into the buffer. Do this 5 times before calling the color average functions for the average of each color.
Arguments: N/A
Peripherals: Flora color sensor

`char identify_color(void)`
Function Desc.: Takes in all average color values, divides the red with green and red with blue to get R/G and R/B ratios. Uses R/G and R/B ratios to accurately identify the paint chips.
Arguments: N/A
Peripherals: N/A

Basic Usage Example

To check if the system is working, place a color chip near the color sensor. These functions should activate and output values to the corresponding register.

color_init(); // initializes the color sensor through the I2C connection
read_color_clear(); // Reads the clear color, saved to the variable data_clear[2] in the format of [high byte, low byte]
read_color_red(); // Reads the red color, saved to the variable data_red[2] in the format of [high byte, low byte]
read_color_green(); // Reads the green color, saved to the variable data_green[2] in the format of [high byte, low byte]

read_color_blue(); // Reads the blue color, saved to the variable data_blue[2] in the format of [high byte, low byte]

Advanced Usage Example

Should return values corresponding to the color ratio averages in the identify_color() function. This function needs to be calibrated to your system usage.

```
    color_init(); // initializes the color sensor through the I2C connection
    pic24_init(); // initialize the pic24
    initializeButton();
    masterCodeGen();   // generate random color code
    clearLCD();

    //get device id
    int device_id_main;
    device_id_main = read_device_id(); // read the device id from the I2C connection
    delay_ms(100);

    get_color_avg(); // get all 4 colors from the pic24 5 times, convert the 2 8-bit numbers for each color into 16-bits and find the average of those for clear, red, green and blue
/*
 * get_color_avg() uses all of the following functions to get an average of all 4 colors:
 * read_color_clear, read_color_red, read_color_green, read_color_blue, putClear, putRed,
 * putGreen, putBlue, clearAvg, redAvg, greenAvg and blueAvg.
*/

     identify_color(); // Divides red_avg with green_avg and red_avg with blue_avg to get red-green and red-blue ratios. Compares the newly calculated ratios with the ratios that correspond to each paint chip to identify the paint chip under the sensor. The ratios for all the paint chips were measured and calculated beforehand at a distance of ~0.7 in.
```