

システム実験 実験第13回レポート

6119019056 山口力也

2019/07/19 日提出

1 目的

本実験では以下の項目を目標とした.

- カラーセンシング技術を理解し, 応用できる.
- カラーセンサのキャリブレーションを理解し, 応用できる.
- I^2C による通信方法を理解し, 状況に合わせて通信プログラムの設定を変更できる.
- 色空間の変換方法を理解し, 色情報を Processing で可視化できる.

2 演習

2.1 演習 8.1.1

カラーセンサの色計測を行うために以下図 1 に示す回路を構成した.

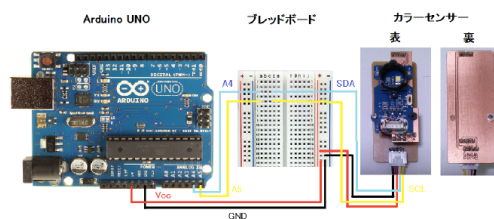


図 1: 演習 8.1.1 の回路

2.2 演習 8.1.2

Arduino で、 $I^2 - C$ による通信を実行するための Wire 関数の使い方について理解した。

以下ソースコード 1 に作成したプログラムのソースコードを示す。

ソースコード 1: 演習 8.1.2(Arduino)

```
1 #include <Wire.h>
2 #define COLOR_SENSOR_ADDR 0x39
3 #define REG_BLOCK_READ 0xCF
4 unsigned int readingdata[20];
5 unsigned int i, green, red, blue;
6 //double X, Y, Z, x, y, z; //演習 8.1.6 で使用
7
8 void setup(){
9   Serial.begin(9600); //シリアル通信の初期化
10  Wire.begin(); //I2C バスに接続
11  Wire.beginTransmission(COLOR_SENSOR_ADDR); //カラーセンサの
    AD 変換開始
12  Wire.write(0x80); //REG_CTL
13  Wire.write(0x03); //CTL_DAT_INITIATE
14  Wire.endTransmission();
15 }
16
17 void loop(){
18   readRGB();
19   //calculateCoordinate(); //演習 8.1.6 で使用
20   delay(500);
21 }
22
23 void readRGB(){
24   Wire.beginTransmission(COLOR_SENSOR_ADDR);
25   Wire.write(REG_BLOCK_READ);
26   Wire.endTransmission(); //送信完了
27   Wire.beginTransmission(COLOR_SENSOR_ADDR); //送受信開始
28   Wire.requestFrom(COLOR_SENSOR_ADDR, 8); //カラーセンサにデ
    ータ要求
29   delay(500);
30   if(Wire.available() >= 8){ //8byte 以上受信したと き
31     for(i =0; i < 8; i++){
32       readingdata[i]=Wire.read(); //データの受信
33     }
34   }
35   green = readingdata[1]*256 + readingdata[0]; //受信データ
    の配列と色情報の順番固定
```

```

36   red = readingdata[3]*256 + readingdata[2]; //2byte の RGB
      値にデータ復元
37   blue = readingdata[5]*256 + readingdata[4];
38   //Serial.println("RGB values = ");
39   Serial.print(red, DEC); Serial.print(","); //DEC: ascii
      code で 10 進表記
40   Serial.print(green, DEC); Serial.print(",");
41   Serial.println(blue, DEC);
42 }

```

2.3 演習 8.1.3

演習 8.1.2 で取得したカラーセンサの値を Processing で可視化した。
以下ソースコード 2 に作成したプログラムのソースコードを示す。

ソースコード 2: 演習 8.1.3(Processing)

```

1  import processing.serial.*;
2  Serial myPort;
3  float Red, Green, Blue;
4  void setup() {
5      size(640, 480);
6      myPort = new Serial(this, "/dev/ttyACM1", 9600);
7      //myPort = new Serial(this, "COM10", 9600);
8      myPort.bufferUntil('\n'); //改行までメッセージ受信
9  }
10 void draw() { //受信した値で描画
11     background(0);
12     fill(Red,0,0); rect(10,470,100,-Red);
13     fill(0,Green,0); rect(120,470,100,-Green);
14     fill(0,0,Blue); rect(230,470,100,-Blue);
15     fill(Red, Green, Blue); rect(360,110,260,260);
16     print("R="+Red+", "); print("G="+Green+", "); println("B
      "+Blue);
17     fill(255);
18     text("Red=",10,10); text(int(Red),40,10);
19     text("Green=",120,10); text(int(Green),160,10);
20     text("Blue=",230,10); text(int(Blue),265,10);
21 }
22 void serialEvent(Serial myPort) {
23     String myString = myPort.readStringUntil('\n'); //シリアル
      バッファ一読込み
24     if (myString != null){
25         myString = trim(myString); //空白文字など消去
26         float data[] = float(split(myString, ',')); //カンマ区切
      りで複数の情報を読み込む

```

```

27     if (data.length >1){
28
29         // キャリブレーションなし
30         /*
31         Red = data[0];
32         Green = data[1]; //ここに Green の処理を入れる
33         Blue = data[2]; //ここに Blue の処理を入れる
34         */
35         // キャリブレーションあり
36         Red = map(data[0], 7424, 36096, 0, 255); //下線部をシ
           リアルモニタの最大値に変更
37         Green = map(data[1], 16392, 60168, 0, 255); //ここに
           Green の処理を入れる
38         Blue = map(data[2], 11520, 42240, 0, 255); //ここに
           Blue の処理を入れる
39         //min 7424,16392,11520
40         //max 36096,60168,42240
41
42         myPort.clear();
43     }
44 }
45 }

```

2.4 演習 8.1.4

フルカラー LED により基本色である RGB を再現した。以下図 2 にブレッ
ドボード上の構成図を示す。

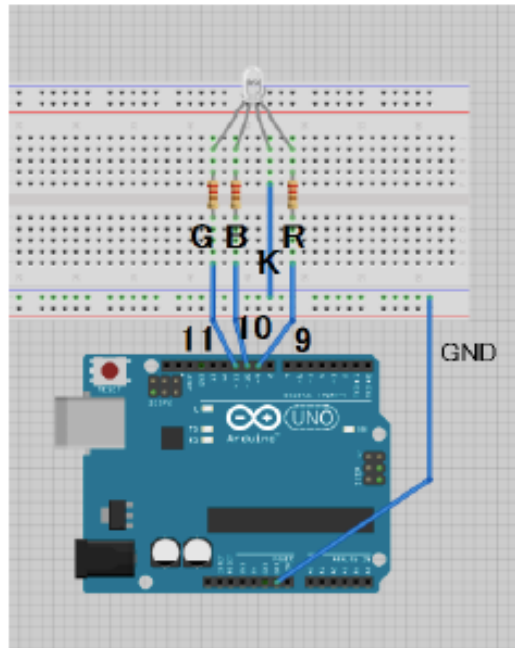


図 2: 演習 8.1.4 の配線図

2.5 演習 8.1.5

カラーセンサによりフルカラー LED の値を読み出しシリアルモニタに表示するプログラムを作成した。ただし同期タイミングを合わせるため、タイマ割り込みを用いた。以下ソースコード 3 に作成したプログラムのソースコードを示す。

ソースコード 3: 演習 8.1.5(Arduino)

```

1 #include <Wire.h>
2
3 #define COLOR_SENSOR_ADDR 0x39
4 #define REG_BLOCK_READ 0xCF
5 #define ledred 9
6 #define ledgreen 11
7 #define ledblue 10
8 unsigned int readingdata[20];
9 unsigned int i, green, red, blue;
10 unsigned long Timenow, Timeprev;
11 int change;
12 int period; //間隔
13 double X, Y, Z, x, y, z; //演習 8.1.6 で使用

```

```

14 void setup(){
15     Serial.begin(9600); //シリアル通信の初期化
16     Wire.begin(); //I2C バスに接続
17     Wire.beginTransmission(COLOR_SENSOR_ADDR); //カラーセンサの
        AD 変換開始
18     Wire.write(0x80); //REG_CTL
19     Wire.write(0x03); //CTL_DAT_INITIATE
20     Wire.endTransmission();
21
22     pinMode(ledred, OUTPUT) ; // RED 9 番ピン
23     pinMode(ledgreen, OUTPUT) ; // GREEN 10 番ピン
24     pinMode(ledblue, OUTPUT) ; // BLUE 9 番ピン
25     analogWrite(ledred, 0); // PWM duty 比
26     analogWrite(ledgreen, 0);
27     analogWrite(ledblue, 0);
28     change = 0;
29     Timeprev = millis();
30 }
31 void lightled() {
32     if(change == 0){
33         analogWrite(ledred, 255); // PWM duty 比
34         analogWrite(ledgreen, 0);
35         analogWrite(ledblue, 0);
36         change++;
37     }
38     else if(change == 1){
39         analogWrite(ledred, 0); // PWM duty 比
40         analogWrite(ledgreen, 255);
41         analogWrite(ledblue, 0);
42         change++;
43     }
44     else if(change == 2){
45         analogWrite(ledred, 0); // PWM duty 比
46         analogWrite(ledgreen, 0);
47         analogWrite(ledblue, 255);
48         change = 0;
49     }
50 }
51 void loop(){
52     readRGB();
53     Timenow =millis();
54     if (Timenow - Timeprev >= period){
55         lightled();
56     }
57     calculateCoordinate(); //演習 8.1.6 で使用
58     Serial.println(change);

```

```

59   delay(500);
60 }
61
62 void readRGB(){
63   Wire.beginTransaction(COLOR_SENSOR_ADDR);
64   Wire.write(REG_BLOCK_READ);
65   Wire.endTransmission(); //送信完了
66   Wire.beginTransaction(COLOR_SENSOR_ADDR); //送受信開始
67   Wire.requestFrom(COLOR_SENSOR_ADDR, 8); //カラーセンサにデ
      ータ要求
68   delay(500);
69   if(Wire.available() >= 8){ //8byte 以上受信したと き
70     for(i =0; i < 8; i++){
71       readingdata[i]=Wire.read(); //データの受信
72     }
73   }
74   green = readingdata[1]*256 + readingdata[0]; //受信データ
      の配列と色情報の順番固定
75   red = readingdata[3]*256 + readingdata[2]; //2byte の RGB
      値にデータ復元
76   blue = readingdata[5]*256 + readingdata[4];
77   Serial.println("RGB values = ");
78   Serial.print(red, DEC); Serial.print(","); //DEC: ascii
      code で 10 進表記
79   Serial.print(green, DEC); Serial.print(",");
80   Serial.println(blue, DEC);
81 }
82 void calculateCoordinate() {
83   X=(-0.142)*red+(1.549)*green+(-0.956)*blue;
84   Y=(-0.324)*red+(1.578)*green+(-0.731)*blue;
85   Z=(-0.682)*red+(0.770)*green+(0.563)*blue;
86   x=X/(X+Y+Z); y=Y/(X+Y+Z);
87   if( (X>0) && (Y>0) && (Z>0) ){
88     Serial.println("x, y = ");
89     Serial.print(x, 2); Serial.print(","); Serial.println(y,
      2);
90   }
91   else Serial.println("Error!");
92 }

```

2.6 演習 8.1.6

RGB 値から xy 値に変換するためのプログラムを作成した. ここで変換行列に

$$A = \begin{pmatrix} -0.142 & 1.549 & -0.956 \\ -0.334 & 1.578 & -0.731 \\ -0.682 & 0.770 & 0.563 \end{pmatrix} \quad (1)$$

を用いた. 以下ソースコード 4 に作成したプログラムのソースコードを示す.

ソースコード 4: 演習 8.1.6(Arduino)

```
1 #include <Wire.h>
2
3 #define COLOR_SENSOR_ADDR 0x39
4 #define REG_BLOCK_READ 0xCF
5 #define ledred 9
6 #define ledgreen 11
7 #define ledblue 10
8 unsigned int readingdata[20];
9 unsigned int i, green, red, blue;
10 unsigned long Timenow, Timeprev;
11 int change;
12 int period; //間隔
13 double X, Y, Z, x, y, z; //演習 8.1.6 で使用
14 void setup(){
15   Serial.begin(9600); //シリアル通信の初期化
16   Wire.begin(); //I2C バスに接続
17   Wire.beginTransmission(COLOR_SENSOR_ADDR); //カラーセンサの
      AD 変換開始
18   Wire.write(0x80); //REG_CTL
19   Wire.write(0x03); //CTL_DAT_INITIATE
20   Wire.endTransmission();
21
22   pinMode(ledred, OUTPUT) ; // RED 9 番ピン
23   pinMode(ledgreen, OUTPUT) ; // GREEN 10 番ピン
24   pinMode(ledblue, OUTPUT) ; // BLUE 9 番ピン
25   analogWrite(ledred, 0); // PWM duty 比
26   analogWrite(ledgreen, 0);
27   analogWrite(ledblue, 0);
28   change = 0;
29   Timeprev = millis();
30 }
31 void lightled() {
32   if(change == 0){
33     analogWrite(ledred, 255); // PWM duty 比
34     analogWrite(ledgreen, 0);
```



```

35     analogWrite(ledblue, 0);
36     change++;
37 }
38 else if(change == 1){
39     analogWrite(ledred, 0); // PWM duty 比
40     analogWrite(ledgreen, 255);
41     analogWrite(ledblue, 0);
42     change++;
43 }
44 else if(change == 2){
45     analogWrite(ledred, 0); // PWM duty 比
46     analogWrite(ledgreen, 0);
47     analogWrite(ledblue, 255);
48     change = 0;
49 }
50 }
51 void loop(){
52     readRGB();
53     Timenow =millis();
54     if (Timenow - Timeprev >= period){
55         lightled();
56     }
57     calculateCoordinate(); //演習 8.1.6 で使用
58     Serial.println(change);
59     delay(500);
60 }
61
62 void readRGB(){
63     Wire.beginTransaction(COLOR_SENSOR_ADDR);
64     Wire.write(REG_BLOCK_READ);
65     Wire.endTransmission(); //送信完了
66     Wire.beginTransaction(COLOR_SENSOR_ADDR); //送受信開始
67     Wire.requestFrom(COLOR_SENSOR_ADDR, 8); //カラーセンサにデ
        ータ要求
68     delay(500);
69     if(Wire.available() >= 8){ //8byte 以上受信したと き
70         for(i =0; i < 8; i++){
71             readingdata[i]=Wire.read(); //データの受信
72         }
73     }
74     green = readingdata[1]*256 + readingdata[0]; //受信データ
        の配列と色情報の順番固定
75     red = readingdata[3]*256 + readingdata[2]; //2byte の RGB
        値にデータ復元
76     blue = readingdata[5]*256 + readingdata[4];
77     Serial.println("RGB values = ");

```

```

78   Serial.print(red, DEC); Serial.print(","); //DEC: ascii
      code で 10 進表記
79   Serial.print(green, DEC); Serial.print(",");
80   Serial.println(blue, DEC);
81 }
82 void calculateCoordinate() {
83   X=(-0.142)*red+(1.549)*green+(-0.956)*blue;
84   Y=(-0.324)*red+(1.578)*green+(-0.731)*blue;
85   Z=(-0.682)*red+(0.770)*green+(0.563)*blue;
86   x=X/(X+Y+Z); y=Y/(X+Y+Z);
87   if( (X>0) && (Y>0) && (Z>0) ){
88     Serial.println("x, y = ");
89     Serial.print(x, 2); Serial.print(","); Serial.println(y,
      2);
90   }
91   else Serial.println("Error!");
92 }

```

3 課題

3.1 課題 8.1.1

演習 8.1.3 で作成したプログラムを元に, Clear 値についても可視化した.
また, カラーセンサの値を読み取る通信速度についても検討した. これについて
はタイマー割り込みを用いて同期させることで改善した.

以下ソースコード 5,6 にそれぞれ作成したプログラムのソースコードを示す.

ソースコード 5: 課題 8.1.1(Arduino)

```

1  #include <Wire.h>
2  #define COLOR_SENSOR_ADDR 0x39
3  #define REG_BLOCK_READ 0xCF
4  unsigned int readingdata[20];
5  unsigned int i, green, red, blue, clr;
6  //double X, Y, Z, x, y, z; //演習 8.1.6 で使用
7
8  void setup(){
9    Serial.begin(9600); //シリアル通信の初期化
10   Wire.begin(); //I2C バスに接続
11   Wire.beginTransmission(COLOR_SENSOR_ADDR); //カラーセンサの
      AD 変換開始
12   Wire.write(0x80); //REG_CTL
13   Wire.write(0x03); //CTL_DAT_INITIATE
14   Wire.endTransmission();
15 }

```

```

16
17 void loop(){
18   readRGB();
19   //calculateCoordinate(); //演習 8.1.6 で使用
20   delay(500);
21 }
22
23 void readRGB(){
24   Wire.beginTransmission(COLOR_SENSOR_ADDR);
25   Wire.write(REG_BLOCK_READ);
26   Wire.endTransmission(); //送信完了
27   Wire.beginTransmission(COLOR_SENSOR_ADDR); //送受信開始
28   Wire.requestFrom(COLOR_SENSOR_ADDR, 8); //カラーセンサにデータ要求
29   delay(500);
30   if(Wire.available() >= 8){ //8byte 以上受信したとき
31     for(i =0; i < 8; i++){
32       readingdata[i]=Wire.read(); //データの受信
33     }
34   }
35   green = readingdata[1]*256 + readingdata[0]; //受信データの配列と色情報の順番固定
36   red = readingdata[3]*256 + readingdata[2]; //2byte の RGB 値にデータ復元
37   blue = readingdata[5]*256 + readingdata[4];
38   clr = readingdata[7]*256 + readingdata[6];
39   Serial.println("RGB values = ");
40   Serial.print(red, DEC); Serial.print(","); //DEC: ascii code で 10 進表記
41   Serial.print(green, DEC); Serial.print(",");
42   Serial.print(blue, DEC); Serial.print(",");
43   Serial.println(clr, DEC);
44 }

```

ソースコード 6: 課題 8.1.1(Processing)

```

1 import processing.serial.*;
2 Serial myPort;
3 float Red, Green, Blue,Clr;
4 void setup() {
5   size(640, 480);
6   myPort = new Serial(this, "/dev/ttyACM0", 9600);
7   //myPort = new Serial(this, "COM10", 9600);
8   myPort.bufferUntil('\n'); //改行までメッセージ受信
9 }
10 void draw() { //受信した値で描画

```

```

11  background(0);
12  fill(Red,0,0); rect(10,470,100,-Red);
13  fill(0,Green,0); rect(120,470,100,-Green);
14  fill(0,0,Blue); rect(230,470,100,-Blue);
15  fill(Red, Green, Blue); rect(360,110,260,260);
16  fill(Clr); rect(360,50,Clr,30);
17  print("R="+Red+", "); print("G="+Green+", "); print("B="+
    Blue+",");println("C="+Clr);
18  fill(255);
19  text("Red=",10,10); text(int(Red),40,10);
20  text("Green=",120,10); text(int(Green),160,10);
21  text("Blue=",230,10); text(int(Blue),265,10);
22  text("Clr=",360,10); text(int(Clr),390,10);
23  }
24  void serialEvent(Serial myPort) {
25      String myString = myPort.readStringUntil('\n'); //シリアル
        バッファー読み込み
26      if (myString != null){
27          myString = trim(myString); //空白文字など消去
28          float data[] = float(split(myString, ',')); //カンマ区切
            りで複数の情報を読み込む
29          if (data.length >1){
30
31              // キャリブレーションなし
32              /*
33              Red = data[0];
34              Green = data[1]; //ここに Green の処理を入れる
35              Blue = data[2]; //ここに Blue の処理を入れる
36              */
37              // キャリブレーションあり
38              Red = map(data[0], 256, 2048, 0, 255); //下線部をシリ
                アルモニタの最大値に変更
39              Green = map(data[1], 264, 2056, 0, 255); //ここに
                Green の処理を入れる
40              Blue = map(data[2], 0, 1280, 0, 255); //ここに Blue
                の処理を入れる
41              Clr = map(data[3],768, 5376, 0, 255); //Clr の処理
42              //min 256,264,0,768
43
44              //max 2048,2056,1280,5376
45
46
47
48              myPort.clear();
49          }
50      }

```

51 }

3.2 課題 8.1.2

演習 8.1.5 のスケッチを元に,Green,Blue を加えまた,シアン, マゼンタ, イエローを表現するプログラムを完成させた.

以下ソースコード 7 に作成したプログラムのソースコードを示す.

ソースコード 7: 課題 8.1.2(Arduino)

```
1 #include <Wire.h>
2
3 #define COLOR_SENSOR_ADDR 0x39
4 #define REG_BLOCK_READ 0xCF
5 #define ledred 9
6 #define ledgreen 11
7 #define ledblue 10
8 unsigned int readingdata[20];
9 unsigned int i, green, red, blue;
10 unsigned long Timenow,Timeprev;
11 int change;
12 int period; //間隔
13 //double X, Y, Z, x, y, z; //演習 8.1.6 で使用
14 void setup(){
15     Serial.begin(9600); //シリアル通信の初期化
16     Wire.begin(); //I2C バスに接続
17     Wire.beginTransmission(COLOR_SENSOR_ADDR); //カラーセンサの
        AD 変換開始
18     Wire.write(0x80); //REG_CTL
19     Wire.write(0x03); //CTL_DAT_INITIATE
20     Wire.endTransmission();
21
22     pinMode(ledred, OUTPUT) ; // RED 9 番ピン
23     pinMode(ledgreen, OUTPUT) ; // GREEN 10 番ピン
24     pinMode(ledblue, OUTPUT) ; // BLUE 9 番ピン
25     analogWrite(ledred, 0); // PWM duty 比
26     analogWrite(ledgreen,0);
27     analogWrite(ledblue,0);
28     change = 0;
29     Timeprev = millis();
30 }
31 void lightled() {
32     //red
33     if(change == 0){
34         analogWrite(ledred, 255); // PWM duty 比
```

```

35     analogWrite(ledgreen, 0);
36     analogWrite(ledblue, 0);
37     change++;
38 }
39 //green
40 else if(change == 1){
41     analogWrite(ledred, 0); // PWM duty 比
42     analogWrite(ledgreen, 255);
43     analogWrite(ledblue, 0);
44     change++;
45 }
46 //blue
47 else if(change == 2){
48     analogWrite(ledred, 0); // PWM duty 比
49     analogWrite(ledgreen, 0);
50     analogWrite(ledblue, 255);
51     change++;
52 }
53 //cyan
54 else if(change == 3){
55     analogWrite(ledred, 0); // PWM duty 比
56     analogWrite(ledgreen, 255);
57     analogWrite(ledblue, 255);
58     change++;
59 }
60 //Magenta
61 else if(change == 4){
62     analogWrite(ledred, 255); // PWM duty 比
63     analogWrite(ledgreen, 0);
64     analogWrite(ledblue, 255);
65     change++;
66 }
67 //Yellow
68 else if(change == 5){
69     analogWrite(ledred, 255); // PWM duty 比
70     analogWrite(ledgreen, 255);
71     analogWrite(ledblue, 0);
72     change = 0;
73 }
74 }
75 void loop(){
76     readRGB();
77     Timenow =millis();
78     if (Timenow - Timeprev >= period){
79         lightled();
80     }

```

```

81 //calculateCoordinate(); //演習 8.1.6 で使用
82 Serial.println(change);
83 delay(500);
84 }
85
86 void readRGB(){
87   Wire.beginTransmission(COLOR_SENSOR_ADDR);
88   Wire.write(REG_BLOCK_READ);
89   Wire.endTransmission(); //送信完了
90   Wire.beginTransmission(COLOR_SENSOR_ADDR); //送受信開始
91   Wire.requestFrom(COLOR_SENSOR_ADDR, 8); //カラーセンサにデ
      ータ要求
92   delay(500);
93   if(Wire.available() >= 8){ //8byte 以上受信したと き
94     for(i =0; i < 8; i++){
95       readingdata[i]=Wire.read(); //データの受信
96     }
97   }
98   green = readingdata[1]*256 + readingdata[0]; //受信データ
      の配列と色情報の順番固定
99   red = readingdata[3]*256 + readingdata[2]; //2byte の RGB
      値にデータ復元
100   blue = readingdata[5]*256 + readingdata[4];
101   Serial.println("RGB values = ");
102   Serial.print(red, DEC); Serial.print(","); //DEC: ascii
      code で 10 進表記
103   Serial.print(green, DEC); Serial.print(",");
104   Serial.println(blue, DEC);
105 }

```

結果としては, 青と赤でマゼンタ, 青と緑でシアン, 赤と緑でイエローがそれぞれフルカラー LED の出力として得られた.

3.3 課題 8.1.3

取得したカラーセンサの RGB 値を XYZ 値に変換後,xy 色度図上に投影した. ただしオートキャリブレーションを行い, あらかじめキャリブレーションはするものとした.

以下ソースコード 8,9 にそれぞれ作成したプログラムのソースコードを示す.

ソースコード 8: 課題 8.1.3(Arduino)

```

1 #include <Wire.h>
2 #define COLOR_SENSOR_ADDR 0x39
3 #define REG_BLOCK_READ 0xCF
4 unsigned int readingdata[20];

```

```

5 unsigned int i, green, red, blue,clr;
6 double X, Y, Z, x, y, z; //演習 8.1.6 で使用
7 float Red, Green, Blue;
8 int dataR, dataG, dataB;
9 int dataR_min, dataG_min, dataB_min;
10 int dataR_max, dataG_max, dataB_max;
11 unsigned long timenow;
12
13 void setup(){
14     pinMode(9, OUTPUT) ; // RED 9 番ピン
15     analogWrite(9, 0); // PWM duty 比
16
17     Serial.begin(9600); //シリアル通信の初期化
18     Wire.begin(); //I2C バスに接続
19     Wire.beginTransmission(COLOR_SENSOR_ADDR); //カラーセンサの
        AD 変換開始
20     Wire.write(0x80); //REG_CTL
21     Wire.write(0x03); //CTL_DAT_INITIATE
22     Wire.endTransmission();
23     delay(1000);
24     //Autocalibration
25     CalibrationColorSensor();
26 }
27 void loop(){
28     readRGB();
29
30     Serial.print(dataR_min);Serial.print("--> ");Serial.
        println(dataR_max);
31     Serial.print(dataG_min);Serial.print("--> ");Serial.
        println(dataG_max);
32     Serial.print(dataB_min);Serial.print("--> ");Serial.
        println(dataB_max);
33
34     //Auto キャリブレーションあり
35     red = map(red, dataR_min, dataR_max, 0, 255); //下線部をシ
        リアルモニタの最大値に変更
36     green = map(green, dataG_min, dataG_max, 0, 255);; //ここ
        に Green の処理を入れる
37     blue = map(blue, dataB_min, dataB_max, 0, 255);; //ここに
        Blue の処理を入れる
38
39     //manual calibration
40     //red = map(red, 0, 1792, 0, 255); //下線部をシリアルモニ
        タの最大値に変更
41     //green = map(green, 0, 2568, 0, 255);; //ここに Green の
        処理を入れる

```



```

42 //blue = map(blue, 0, 1792, 0, 255);; //ここに Blue の処
    理を入れる
43 /*
44 Serial.print(red); Serial.print(","); //DEC: ascii code で
    10 進表記
45 Serial.print(green); Serial.print(",");
46 Serial.println(blue);
47 */
48 calculateCoordinate(); //演習 8.1.6 で使用
49
50 analogWrite(9, 255); // PWM duty 比
51 delay(200); //点滅周期は各自で調整
52 analogWrite(9, 0);
53 delay(200); //点滅周期は各自で調整
54
55 }
56 void readRGB(){
57   Wire.beginTransaction(COLOR_SENSOR_ADDR);
58   Wire.write(REG_BLOCK_READ);
59   Wire.endTransmission(); //送信完了
60   Wire.beginTransaction(COLOR_SENSOR_ADDR); //送受信開始
61   Wire.requestFrom(COLOR_SENSOR_ADDR, 8); //カラーセンサにデ
    ータ要求
62   delay(200);
63   if(Wire.available() >= 8){ //8byte 以上受信したと き
64     for(i =0; i < 8; i++){
65       readingdata[i]=Wire.read(); //データの受信
66     }
67   }
68   green = readingdata[1]*256 + readingdata[0]; //受信データ
    の配列と色情報の順番固定
69   red = readingdata[3]*256 + readingdata[2]; //2byte の RGB
    値にデータ復元
70   blue = readingdata[5]*256 + readingdata[4];
71   clr = readingdata[7]*256 + readingdata[6];
72   /*
73   Serial.println("RGB values = ");
74   Serial.print(red, DEC); Serial.print(","); //DEC: ascii
    code で 10 進表記
75   Serial.print(green, DEC); Serial.print(",");
76   Serial.println(blue, DEC); //Serial.print(",");
77   Serial.println(clr, DEC);
78   */
79 }
80 void calculateCoordinate(){
81   X=(-0.142)*red+(1.549)*green+(-0.956)*blue;

```

```

82   Y=(-0.324)*red+(1.578)*green+(-0.731)*blue;
83   Z=(-0.682)*red+(0.770)*green+(0.563)*blue;
84
85   x=X/(X+Y+Z);
86   y=Y/(X+Y+Z);
87
88   if( (X>0) && (Y>0) && (Z>0) ){
89       //Serial.println("x, y = ");
90       Serial.print(x, 2); Serial.print(","); Serial.println(y,
91           2);
92   }
93   else Serial.println("Error!");
94 }
95 void CalibrationColorSensor()
96 {
97     unsigned long timeInit;
98
99     dataR_min = 30000;
100    dataG_min = 30000;
101    dataB_min = 30000;
102    dataR_max = 0;
103    dataG_max = 0;
104    dataB_max = 0;
105
106    timeInit = millis();
107    while ( 1 ) {
108        readRGB();
109        if ( red < dataR_min ) dataR_min = red;
110        if ( green < dataG_min ) dataG_min = green;
111        if ( blue < dataB_min ) dataB_min = blue;
112        if ( red > dataR_max ) dataR_max = red;
113        if ( green > dataG_max ) dataG_max = green;
114        if ( blue > dataB_max ) dataB_max = blue;
115
116        if ( millis() - timeInit > 5000 )
117            break;
118    }
119
120 }

```

ソースコード 9: 課題 8.1.3(Processing)

```

1 PImage img; // Declare variable "a" of type PImage
2 import processing.serial.*;
3 Serial myPort;

```

```

4 float x,y,xx,yy;
5
6
7 void setup() {
8   size(525, 575);
9   // The image file must be in the data folder of the
      current sketch
10  // to load successfully
11  img = loadImage("xymapF.png"); // Load the image into the
      program
12  myPort = new Serial(this,"/dev/ttyACM0",9600);
13  myPort.bufferUntil('\n');
14 }
15 void draw() { //受信した値で描画
16
17   image(img, 0, 0);
18   //Figure drawn in rect TL =(25,55), BR =(470,560)
19
20   println("x="+x);
21   println("y="+y);
22
23   xx = map(x, 0, 0.8, 25, 470);
24   yy = map(y, 0, 0.9, 560, 55);
25   println("xx="+xx);
26   println("yy="+yy);
27
28   fill(255);
29   ellipse(xx,yy,15,15);
30   //red point
31   ellipse(405,358,10,10);
32   //green point
33   ellipse(203,193,10,10);
34   //blue point
35   ellipse(114,523,10,10);
36
37   strokeWeight(5);
38   line(405,358,203,193); //R-G
39   line(405,358,114,523); //R-G
40   line(203,193,114,523); //G-B
41 }
42
43 void serialEvent(Serial myPort) {
44   String myString = myPort.readStringUntil('\n'); //シリアル
      バッファー読み込み
45   if (myString != null){
46     myString = trim(myString); //空白文字など消去

```

```

47     float data[] = float(split(myString, ',')); //カンマ区切
        りで複数の情報を読み込む
48     if (data.length > 1) {
49         x = data[0];
50         y = data[1];
51         myPort.clear();
52     }
53 }
54 }

```

また, 以下図 3 にプログラムの実行結果の図を示す.

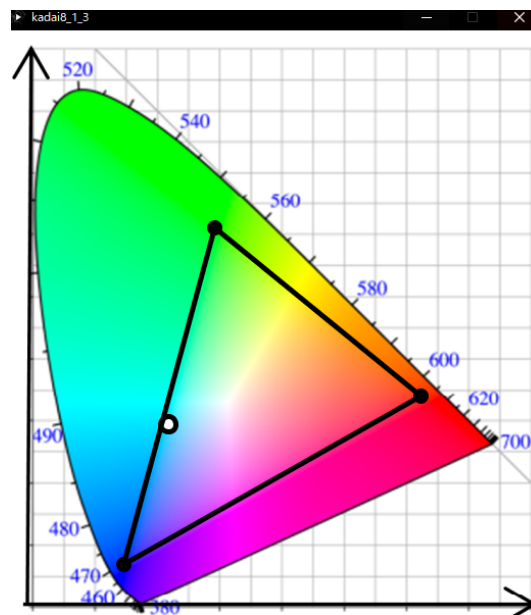


図 3: 課題 8.1.3 の実行結果

3.4 課題 8.1.4

取得したカラーセンサの RGB 値を XYZ 値に変換後, xy 色度図上に投影した. ただしオートキャリブレーションを行わず, あらかじめカラーセンサの値を用いてマニュアルキャリブレーションするものとした.

以下ソースコード 10,11 にそれぞれ作成したプログラムのソースコードを示す.

ソースコード 10: 課題 8.1.4(Arduino)

```

1 #include <Wire.h>
2

```

```

3  #define COLOR_SENSOR_ADDR 0x39
4  #define REG_BLOCK_READ 0xCF
5  #define ledred 9
6  #define ledgreen 11
7  #define ledblue 10
8  unsigned int readingdata[20];
9  unsigned int i, green, red, blue;
10 unsigned long Timenow,Timeprev;
11 int change;
12 int period; //間隔
13 double X, Y, Z, x, y, z; //演習 8.1.6 で使用
14 void setup(){
15     Serial.begin(9600); //シリアル通信の初期化
16     Wire.begin(); //I2C バスに接続
17     Wire.beginTransmission(COLOR_SENSOR_ADDR); //カラーセンサの
        AD 変換開始
18     Wire.write(0x80); //REG_CTL
19     Wire.write(0x03); //CTL_DAT_INITIATE
20     Wire.endTransmission();
21
22     pinMode(ledred, OUTPUT) ; // RED 9 番ピン
23     pinMode(ledgreen, OUTPUT) ; // GREEN 10 番ピン
24     pinMode(ledblue, OUTPUT) ; // BLUE 9 番ピン
25     analogWrite(ledred, 0); // PWM duty 比
26     analogWrite(ledgreen,0);
27     analogWrite(ledblue,0);
28     change = 0;
29     Timeprev = millis();
30 }
31 void lightled() {
32     if(change == 0){
33         analogWrite(ledred, 255); // PWM duty 比
34         analogWrite(ledgreen, 0);
35         analogWrite(ledblue, 0);
36         change++;
37     }
38     else if(change == 1){
39         analogWrite(ledred, 0); // PWM duty 比
40         analogWrite(ledgreen, 255);
41         analogWrite(ledblue, 0);
42         change++;
43     }
44     else if(change == 2){
45         analogWrite(ledred, 0); // PWM duty 比
46         analogWrite(ledgreen, 0);
47         analogWrite(ledblue, 255);

```

```

48     change = 0;
49 }
50 }
51 void loop(){
52     readRGB();
53     Timenow =millis();
54     if (Timenow - Timeprev >= period){
55         lightled();
56     }
57     calculateCoordinate(); //演習 8.1.6 で使用
58     //Serial.println(change);
59     delay(500);
60 }
61
62 void readRGB(){
63     Wire.beginTransaction(COLOR_SENSOR_ADDR);
64     Wire.write(REG_BLOCK_READ);
65     Wire.endTransmission(); //送信完了
66     Wire.beginTransaction(COLOR_SENSOR_ADDR); //送受信開始
67     Wire.requestFrom(COLOR_SENSOR_ADDR, 8); //カラーセンサにデ
        ータ要求
68     delay(500);
69     if(Wire.available() >= 8){ //8byte 以上受信したと き
70         for(i =0; i < 8; i++){
71             readingdata[i]=Wire.read(); //データの受信
72         }
73     }
74     green = readingdata[1]*256 + readingdata[0]; //受信データ
        の配列と色情報の順番固定
75     red = readingdata[3]*256 + readingdata[2]; //2byte の RGB
        値にデータ復元
76     blue = readingdata[5]*256 + readingdata[4];
77     /*
78     Serial.println("RGB values = ");
79     Serial.print(red, DEC); Serial.print(","); //DEC: ascii
        code で 10 進表記
80     Serial.print(green, DEC); Serial.print(",");
81     Serial.println(blue, DEC);
82     */
83 }
84 void calculateCoordinate() {
85     X=(-0.142)*red+(1.549)*green+(-0.956)*blue;
86     Y=(-0.324)*red+(1.578)*green+(-0.731)*blue;
87     Z=(-0.682)*red+(0.770)*green+(0.563)*blue;
88     x=X/(X+Y+Z); y=Y/(X+Y+Z);
89     if( (X>0) && (Y>0) && (Z>0) ){

```

```

90 //Serial.println("x, y = ");
91   Serial.print(x, 2); Serial.print(","); Serial.println(y,
      2);
92 }
93 else Serial.println("Error!");
94 }

```

ソースコード 11: 課題 8.1.4(Processing)

```

1 PImage img; // Declare variable "a" of type PImage
2 import processing.serial.*;
3 Serial myPort;
4 float x,y,xx,yy;
5
6
7 void setup() {
8   size(525, 575);
9   // The image file must be in the data folder of the
      current sketch
10  // to load successfully
11  img = loadImage("xymapF.png"); // Load the image into the
      program
12  myPort = new Serial(this,"/dev/ttyACM0",9600);
13  myPort.bufferUntil('\n');
14 }
15 void draw() { //受信した値で描画
16
17   image(img, 0, 0);
18   //Figure drawn in rect TL =(25,55), BR =(470,560)
19
20   println("x="+x);
21   println("y="+y);
22
23   xx = map(x, 0, 0.8, 25, 470);
24   yy = map(y, 0, 0.9, 560, 55);
25   println("xx="+xx);
26   println("yy="+yy);
27
28   fill(255);
29   ellipse(xx,yy,15,15);
30   //red point
31   ellipse(405,358,10,10);
32   //green point
33   ellipse(203,193,10,10);
34   //blue point
35   ellipse(114,523,10,10);

```

```

36
37  strokeWeight(5);
38  line(405,358,203,193); //R-G
39  line(405,358,114,523); //R-G
40  line(203,193,114,523); //G-B
41 }
42
43 void serialEvent(Serial myPort) {
44   String myString = myPort.readStringUntil('\n'); //シリアル
    パッファー読み込み
45   if (myString != null){
46     myString = trim(myString); //空白文字など消去
47     float data[] = float(split(myString, ',')); //カンマ区切
        りで複数の情報を読み込む
48     if (data.length > 1) {
49       x = data[0];
50       y = data[1];
51       myPort.clear();
52     }
53   }
54 }

```

また、以下図 4 にプログラムの実行結果の図を示す。

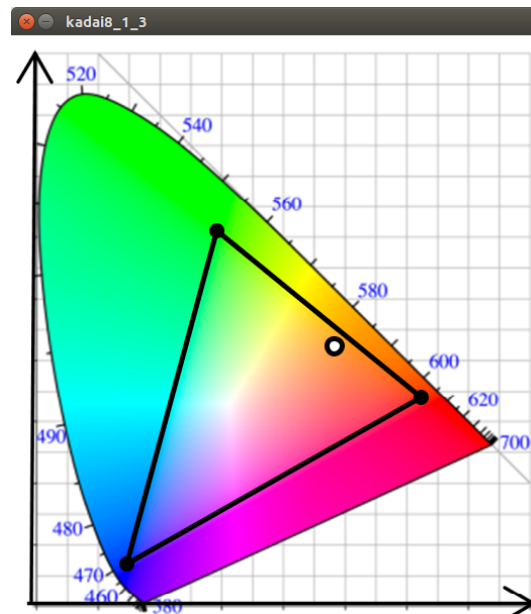


図 4: 課題 8.1.4 の実行結果

正直課題 8.1.3 と課題 8.1.4 であまり違いがでなかった。キャリブレーション

ンがうまくいっていないのか青や赤をセンシングしてもそれに近いところまでいかず中間の値で止まっていた。

3.5 課題 8.1.5

色をセンシングしてから表示するまでにかかる内部処理の時間を millis 関数を用いて計測した。以下ソースコード 12 に作成したプログラムのソースコードを示す。

ソースコード 12: 課題 8.1.5(Arduino)

```
1 #include <Wire.h>
2 #define COLOR_SENSOR_ADDR 0x39
3 #define REG_BLOCK_READ 0xCF
4 unsigned int readingdata[20];
5 unsigned int i, green, red, blue, clr;
6 double X, Y, Z, x, y, z; //演習 8.1.6 で使用
7 float Red, Green, Blue;
8 int dataR, dataG, dataB;
9 int dataR_min, dataG_min, dataB_min;
10 int dataR_max, dataG_max, dataB_max;
11 unsigned long timenow, timeprev;
12
13 void setup(){
14   pinMode(9, OUTPUT) ; // RED 9 番ピン
15   analogWrite(9, 0); // PWM duty 比
16   timeprev = millis();
17   Serial.begin(9600); //シリアル通信の初期化
18   Wire.begin(); //I2C バスに接続
19   Wire.beginTransmission(COLOR_SENSOR_ADDR); //カラーセンサの
      AD 変換開始
20   Wire.write(0x80); //REG_CTL
21   Wire.write(0x03); //CTL_DAT_INITIATE
22   Wire.endTransmission();
23   delay(1000);
24   //Autocalibration
25   CalibrationColorSensor();
26 }
27 void loop(){
28   timeprev = millis();
29   readRGB();
30
31   Serial.print(dataR_min);Serial.print("--> ");Serial.
      println(dataR_max);
32   Serial.print(dataG_min);Serial.print("--> ");Serial.
      println(dataG_max);
```

```

33 Serial.print(dataB_min);Serial.print("--> ");Serial.
    println(dataB_max);
34
35 //Auto キャリブレーションあり
36 red = map(red, dataR_min, dataR_max, 0, 255); //下線部をシ
    リアルモニタの最大値に変更
37 green = map(green, dataG_min, dataG_max, 0, 255);; //ここ
    に Green の処理を入れる
38 blue = map(blue, dataB_min, dataB_max, 0, 255);; //ここに
    Blue の処理を入れる
39
40 //manual calibration
41 //red = map(red, 0, 1792, 0, 255); //下線部をシリアルモニ
    タの最大値に変更
42 //green = map(green, 0, 2568, 0, 255);; //ここに Green の
    処理を入れる
43 //blue = map(blue, 0, 1792, 0, 255);; //ここに Blue の処
    理を入れる
44
45 Serial.print(red); Serial.print(","); //DEC: ascii code で
    10 進表記
46 Serial.print(green); Serial.print(",");
47 Serial.println(blue);
48 timenow = millis();
49 Serial.println(timenow - timeprev);
50
51
52 calculateCoordinate(); //演習 8.1.6 で使用
53
54 analogWrite(9, 255); // PWM duty 比
55 delay(200); //点滅周期は各自で調整
56 analogWrite(9, 0);
57 delay(200); //点滅周期は各自で調整
58
59 }
60 void readRGB(){
61   Wire.beginTransmission(COLOR_SENSOR_ADDR);
62   Wire.write(REG_BLOCK_READ);
63   Wire.endTransmission(); //送信完了
64   Wire.beginTransmission(COLOR_SENSOR_ADDR); //送受信開始
65   Wire.requestFrom(COLOR_SENSOR_ADDR, 8); //カラーセンサにデ
    ータ要求
66   delay(200);
67   if(Wire.available() >= 8){ //8byte 以上受信したと き
68     for(i =0; i < 8; i++){
69       readingdata[i]=Wire.read(); //データの受信

```

```

70     }
71 }
72 green = readingdata[1]*256 + readingdata[0]; //受信データ
      の配列と色情報の順番固定
73 red = readingdata[3]*256 + readingdata[2]; //2byte の RGB
      値にデータ復元
74 blue = readingdata[5]*256 + readingdata[4];
75 clr = readingdata[7]*256 + readingdata[6];
76 /*
77 Serial.println("RGB values = ");
78 Serial.print(red, DEC); Serial.print(","); //DEC: ascii
      code で 10 進表記
79 Serial.print(green, DEC); Serial.print(",");
80 Serial.println(blue, DEC); //Serial.print(",");
81 Serial.println(clr, DEC);
82 */
83 }
84 void calculateCoordinate(){
85     X=(-0.142)*red+(1.549)*green+(-0.956)*blue;
86     Y=(-0.324)*red+(1.578)*green+(-0.731)*blue;
87     Z=(-0.682)*red+(0.770)*green+(0.563)*blue;
88
89     x=X/(X+Y+Z);
90     y=Y/(X+Y+Z);
91
92     if( (X>0) && (Y>0) && (Z>0) ){
93         //Serial.println("x, y = ");
94         Serial.print(x, 2); Serial.print(","); Serial.println(y,
          2);
95     }
96     else Serial.println("Error!");
97 }
98
99 void CalibrationColorSensor()
100 {
101     unsigned long timeInit;
102
103     dataR_min = 30000;
104     dataG_min = 30000;
105     dataB_min = 30000;
106     dataR_max = 0;
107     dataG_max = 0;
108     dataB_max = 0;
109
110     timeInit = millis();
111     while ( 1 ) {

```

```
112     readRGB();
113     if ( red < dataR_min ) dataR_min = red;
114     if ( green < dataG_min ) dataG_min = green;
115     if ( blue < dataB_min ) dataB_min = blue;
116     if ( red > dataR_max ) dataR_max = red;
117     if ( green > dataG_max ) dataG_max = green;
118     if ( blue > dataB_max ) dataB_max = blue;
119
120     if ( millis() - timeInit > 5000 )
121         break;
122 }
123
124 }
```

計測すると大体 200~203ms だった. これより, 計測に 200ms かかると考えると, センシングのタイミング (サンプリング周期) は 400ms 以上にすべきだと考えられる.