



Lab 5: Color Sensing

Tuesday 2/10 2:15pm

Jp Marhefka, Aved Gorji, & Gwyn Anawalt

Part 1

white:

Color Temp: 3743 K - Lux: 303 - R: 455 G: 416 B: 281 C: 1174

Color Temp: 3735 K - Lux: 309 - R: 468 G: 426 B: 288 C: 1203

Color Temp: 3728 K - Lux: 315 - R: 476 G: 433 B: 292 C: 1222

Color Temp: 3737 K - Lux: 309 - R: 466 G: 425 B: 287 C: 1201

Color Temp: 3744 K - Lux: 308 - R: 463 G: 423 B: 286 C: 1193

Color Temp: 3728 K - Lux: 312 - R: 471 G: 429 B: 289 C: 1210

Color Temp: 3731 K - Lux: 313 - R: 472 G: 430 B: 290 C: 1215

Color Temp: 3734 K - Lux: 313 - R: 473 G: 431 B: 291 C: 1217

Color Temp: 3731 K - Lux: 313 - R: 472 G: 430 B: 290 C: 1214

Color Temp: 3736 K - Lux: 311 - R: 471 G: 429 B: 290 C: 1212

Average White Paper:

White Paper Noise Levels

R Avg.:

G Avg:

B Avg:

Black:

Color Temp: 3758 K - Lux: 61 - R: 103 G: 90 B: 64 C: 258

Color Temp: 3871 K - Lux: 68 - R: 106 G: 97 B: 69 C: 273

Color Temp: 3897 K - Lux: 75 - R: 114 G: 106 B: 75 C: 296

Color Temp: 3870 K - Lux: 82 - R: 126 G: 116 B: 82 C: 326

Color Temp: 3879 K - Lux: 81 - R: 124 G: 115 B: 81 C: 322

Color Temp: 3858 K - Lux: 80 - R: 122 G: 113 B: 79 C: 316

Color Temp: 3869 K - Lux: 79 - R: 123 G: 113 B: 80 C: 318

Color Temp: 3899 K - Lux: 78 - R: 120 G: 111 B: 79 C: 312

Color Temp: 3878 K - Lux: 79 - R: 121 G: 112 B: 79 C: 314

Color Temp: 3889 K - Lux: 80 - R: 122 G: 113 B: 80 C: 317

Average Black Paper:

C Avg:

Part 2

Blue Data:

0	4	42	86		
1	6	43	87		
2	8	44	87		
3	13	45	87		
4	16	46	87		
5	19	47	88		
6	25	48	89		
7	29	49	88		
8	32	50	89		
9	35	51	89		
10	40	52	89		
11	44	53	89		
12	45	54	89		
13	48	55	90		
14	50	56	90		
15	55	57	90		
16	57	58	90		
17	60	59	90		
18	60	60	90		
19	62	61	91		
20	65	62	90		
21	66	63	90		
22	68	64	91		
23	70	65	91		
24	72	66	91		
25	73	67	91	84	91
26	74	68	91	85	92
27	77	69	91	86	91
28	78	70	91	87	92
29	78	71	91	88	92
30	79	72	92	89	91
31	80	73	91	90	92
32	80	74	91	91	91
33	81	75	91	92	92
34	81	76	91	93	92
35	82	77	91	94	92
36	83	78	91	95	92
37	84	79	91	96	92
38	85	80	91	97	92
39	85	81	91	98	92
40	85	82	92	99	92
41	86	83	92		

Max value = 92

90% of 92 = 82.8

99% of 92 = 91.08

Blue delay 90%: $((0.9 * (\text{maxvalue})) [i]) * 10 = 360$

Blue delay 99%: $((0.99 * (\text{maxvalue})) [i]) * 10 = 640$

Red Data:

0	4	42	79		
1	5	43	80		
2	9	44	80		
3	12	45	80		
4	14	46	80		
5	17	47	81		
6	22	48	81		
7	26	49	81		
8	28	50	81		
9	31	51	82		
10	36	52	82		
11	38	53	82		
12	41	54	82		
13	43	55	82		
14	46	56	82		
15	48	57	82		
16	53	58	83		
17	54	59	83		
18	56	60	83		
19	58	61	83		
20	59	62	83		
21	60	63	83		
22	61	64	84		
23	64	65	83		
24	65	66	83	84	84
25	66	67	83	85	84
26	68	68	83	86	84
27	69	69	83	87	84
28	71	70	84	88	85
29	71	71	84	89	84
30	72	72	84	90	84
31	72	73	84	91	84
32	74	74	84	92	84
33	74	75	84	93	84
34	75	76	84	94	84
35	75	77	84	95	84
36	76	78	84	96	84
37	77	79	84	97	84
38	77	80	84	98	84
39	78	81	84	99	84
40	78	82	84		
41	79	83	84		

Max value = 84

90% of 84 = 75.6

99% of 84 = 83.16

Red delay 90%: $((0.9 * (\text{maxvalue})) [i]) * 10 = 350$

Red delay 99%: $((0.99 * (\text{maxvalue})) [i]) * 10 = 670$

Green Data:

0	38	41	329		
1	47	42	329		
2	67	43	330		
3	95	44	331		
4	113	45	332		
5	126	46	332		
6	141	47	333		
7	153	48	334		
8	174	49	334		
9	184	50	334		
10	195	51	335		
11	205	52	335		
12	222	53	336		
13	228	54	336		
14	235	55	336		
15	248	56	337		
16	259	57	337		
17	260	58	337		
18	266	59	337		
19	269	60	338		
20	278	61	337		
21	280	62	338		
22	286	63	338	82	339
23	290	64	338	83	339
24	295	65	338	84	339
25	298	66	338	85	339
26	301	67	339	86	339
27	303	68	339	87	339
28	310	69	338	88	339
29	311	70	339	89	339
30	313	71	339	90	339
31	315	72	339	91	339
32	317	73	339	92	339
33	319	74	339	93	340
34	320	75	339	94	339
35	321	76	339	95	340
36	323	77	339	96	340
37	324	78	339	97	339
38	326	79	339	98	339
39	327	80	339	99	340
40	328	81	339		

Max value = 340

90% of 340 = 306

99% of 340 = 334

Green delay 90%: $((0.9 * (\text{maxvalue})) [i]) * 10 = 275$

Green delay 99%: $((0.99 * (\text{maxvalue})) [i]) * 10 = 490$

100ms program:

```
#define RED 44    // 90% delay 350 us
#define BLUE 46   // 90% delay 360 us
#define GREEN 42  // 90% delay 275 us
#define PHOTOIN A0

int i;
int r_reading, b_reading, g_reading;

void setup() {

    Serial.begin(9600);
    pinMode(RED, OUTPUT);
    pinMode(BLUE, OUTPUT);
    pinMode(GREEN, OUTPUT);
    digitalWrite(RED, LOW);
    digitalWrite(BLUE, LOW);
    digitalWrite(GREEN, LOW);
    analogReference(DEFAULT);
    pinMode(PHOTOIN, INPUT);

}

void loop() {

    digitalWrite(RED, HIGH);
    delayMicroseconds(350);
    for (i = 0; i < 4; i++) {
        r_reading += analogRead(PHOTOIN);
    }
    digitalWrite(RED, LOW);
    delay(100);

    digitalWrite(BLUE, HIGH);
    delayMicroseconds(360);
    for (i = 0; i < 4; i++) {
        b_reading += analogRead(PHOTOIN);
    }
}
```

```
digitalWrite(BLUE, LOW);
delay(100);

digitalWrite(GREEN, HIGH);
delayMicroseconds(275);
for (i = 0; i < 4; i++) {
    g_reading += analogRead(PHOTOIN);
}
digitalWrite(GREEN, LOW);
delay(100);

Serial.print("r = ");
Serial.println(r_reading);
Serial.print("b = ");
Serial.println(b_reading);
Serial.print("g = ");
Serial.println(g_reading);

r_reading = b_reading = g_reading = 0;
}
```

Red:

Green:

Blue:

White:

Black:

Discrimination function:

```
#define RED 44      // 90% delay 350 us
#define BLUE 46     // 90% delay 360 us
#define GREEN 42    // 90% delay 275 us
#define WHITE // update pins when we wire
#define PHOTOIN A0
#define BLK_THRESH 50 // update with better numbers from testing
#define RED_THRESH 82
#define BLUE_THRESH 250
#define GRN_THRESH 334

int i;
int reading;

void setup() {

    Serial.begin(9600);
    pinMode(RED, OUTPUT);
    pinMode(BLUE, OUTPUT);
    pinMode(GREEN, OUTPUT);
    pinMode(WHITE, OUTPUT);
    digitalWrite(RED, LOW);
    digitalWrite(BLUE, LOW);
    digitalWrite(GREEN, LOW);
    digitalWrite(WHITE, LOW);
    analogReference(DEFAULT);
    pinMode(PHOTOIN, INPUT);

}

void loop() {

    delayMicroseconds(400);
    for (i = 0; i < 4; i++) {
        reading += analogRead(PHOTOIN);
    }
    reading = reading/4;
    delay(100);
```

```
if (reading < BLK_THRESH) {
    Serial.print("black");
    digitalWrite(RED, HIGH);
    digitalWrite(BLUE, LOW);
    digitalWrite(GREEN, LOW);
    digitalWrite(WHITE, LOW);
} if else (reading < RED_THRESH) {
    Serial.print("red");
    digitalWrite(RED, HIGH);
    digitalWrite(BLUE, LOW);
    digitalWrite(GREEN, LOW);
    digitalWrite(WHITE, LOW);
} if else (reading < BLUE_THRESH) {
    Serial.print("blue");
    digitalWrite(RED, LOW);
    digitalWrite(BLUE, HIGH);
    digitalWrite(GREEN, LOW);
    digitalWrite(WHITE, LOW);
} if else (reading < GRN_THRESH) {
    Serial.print("green");
    digitalWrite(RED, LOW);
    digitalWrite(BLUE, LOW);
    digitalWrite(GREEN, HIGH);
    digitalWrite(WHITE, LOW);
} else (reading >= GRN_THRESH) {
    Serial.print("white");
    digitalWrite(RED, LOW);
    digitalWrite(BLUE, LOW);
    digitalWrite(GREEN, LOW);
    digitalWrite(WHITE, HIGH);
}

reading = 0;

}
```

Problems We Encountered:

Update this after we demo

There was a logical error writing the code for step 5 in experiment 2, the 100ms program. In the beginning we thought they would all run at the same time, but then we realized that the red and blue and green operations happen in sequence. This was easily fixed by not implementing everything in one for loop and entering the 90% values we measured previously.