

```

1  #include <Adafruit_DotStar.h>
2  #include <SPI.h>
3
4  #define SERIAL_BAUD_RATE 9600
5
6
7  #define DATAPIN    4    //data signal input pin
8  #define CLOCKPIN  5    //clock signal input pin
9  #define NUMPIXELS 255  // Number of LEDs in strip
10
11 #define Delta_X    10    // mm
12
13 #define blue_pin   8    //blue LED PIN
14
15 float    NAobj = 0.25;  //numerical aperture
16 float    d = 150;      //Distance from the sample to Adafruit DotStar LEDs
17
18 boolean Beginflag = true;
19 boolean flag = true;
20 char terminator = '/';
21 String mode;
22 uint32_t color = 0xFF0000;
23
24 //declare a DotStar object
25 Adafruit_DotStar strip(NUMPIXELS, DATAPIN, CLOCKPIN, DOTSTAR_BRG);
26
27 void setup() {
28   #if defined(__AVR_ATtiny85__) && (F_CPU == 16000000L)
29     clock_prescale_set(clock_div_1); // Enable 16 MHz on Trinket
30   #endif
31
32   // Initialize serial interface
33   Serial.begin(SERIAL_BAUD_RATE);
34   pinMode(blue_pin, OUTPUT);
35
36   Serial.println("<Arduino is ready>");
37   Serial.println("-----");
38   Serial.println("Command List:");
39   Serial.println("-----");
40   Serial.println("COMMAND: Fluorescence excitation");
41   Serial.println("SYNTAX:  fe/color_on/off/");
42   Serial.println("EXAMPLE: fe/blue_on/");
43   Serial.println("-----");
44   Serial.println("COMMAND: Bright Field");
45   Serial.println("SYNTAX:  bf/(r,g,b)/");
46   Serial.println("EXAMPLE: bf/(000,064,254)/");
47   Serial.println("-----");
48   Serial.println("COMMAND: Dark Field");
49   Serial.println("SYNTAX:  df/(r,g,b)/");
50   Serial.println("EXAMPLE: df/(000,064,254)/");
51   Serial.println("-----");
52   Serial.println("COMMAND: Phase Contrast");
53   Serial.println("SYNTAX:  pc/(r,g,b)/field_direction(bt/bb/bl/br/dt/db/dl/dr)");
54   Serial.println("EXAMPLE: pc/(000,064,254)/bl/");
55   Serial.println("-----");
56   Serial.println("COMMAND: Multiple Points");
57   Serial.println("SYNTAX:  mp/number(1~255)/(r,g,b)num1/(r,g,b)num2/.../");
58   Serial.println("EXAMPLE: mp/3/(000,064,254)254/.../");
59   Serial.println("-----");
60   Serial.println("COMMAND: Set Parameters");
61   Serial.println("SYNTAX:  set/dist/num/");
62   Serial.println("          set/na/num/");
63   Serial.println("EXAMPLE: set/dist/90/");
64   Serial.println("          set/na/0.5/");
65   Serial.println("-----");
66   Serial.println("COMMAND: Turn Off All");
67   Serial.println("SYNTAX:  x/");
68   Serial.println("-----");
69   Serial.println("-----");
70
71   strip.begin(); // Initialize pins for output
72   strip.show();  // Turn all LEDs off ASAP
73 }

```

```

74
75 void loop() {
76     if (Serial.available()) {
77         if (Beginflag) {
78             mode = Serial.readStringUntil(terminator); //set mode
79             Serial.print("MODE: ");
80             Serial.println(mode);
81             Beginflag = false; //maintain mode until next command
82         }
83         if (flag) {
84             char val = mode.charAt(0); // get the first character of mode
85             switch (val) {
86                 case 'f': //fe
87                     Fluorescence_excitation(); // Fluorescence excitation function
88                     break;
89
90                 case 'b': //bf
91                     bright_field(); // bright field function
92                     break;
93
94                 case 'd': //df
95                     dark_field(); // dark field function
96                     break;
97
98                 case 'p': //pc
99                     phase_contrast(); // phase contrast function
100                     break;
101
102                 case 'm': //mp
103                     multiple_points(); //multiple points function
104                     break;
105
106                 case 's': //set
107                     set(); // set parameters function
108                     break;
109
110                 case 'x': //x
111                     strip.clear();
112                     strip.show();
113                     Serial.println("Turn Off All");
114                     Serial.println("-----");
115                     break;
116
117                 default:
118                     Serial.println("Wrong Command.");
119                     break;
120             }
121             flag = false;
122         }
123         if (Serial.available()) {
124             char tem = Serial.read(); //gets one byte from serial buffer
125             if (tem == '\n') {
126                 Beginflag = true;
127                 flag = true; //maintain lights mode until next command( avoid
128                             flashing)
129             }
130         }
131     }
132
133     /* Fluorescence excitation function */
134     void Fluorescence_excitation() {
135         // read and print light color and status
136         String Status = Serial.readStringUntil(terminator);
137         Serial.print("STATUS: ");
138         Serial.println(Status);
139         Serial.println("-----");
140
141         if (Status == "blue_on")
142         {
143             digitalWrite(blue_pin, 0);
144         }
145         if (Status == "blue_off")

```

```

146     {
147         digitalWrite(blue_pin, 1);
148     }
149 }
150
151
152 /* bright field function */
153 void bright_field() {
154
155     // calculate optical parameters
156     float theta = asin(NAobj);
157     double x = d * tan(theta);
158     int radius = int(x / Delta_X) + 1;          //bright field diameter
159
160     // read and print light color
161     String Col = Serial.readStringUntil(terminator);
162     int red = (Col.substring(1, 4)).toInt();
163     int green = (Col.substring(5, 8)).toInt();
164     int blue = (Col.substring(9, 12)).toInt();
165     Serial.print("RED: ");
166     Serial.print(red);
167     Serial.print(" GREEN: ");
168     Serial.print(green);
169     Serial.print(" BLUE: ");
170     Serial.println(blue);
171     Serial.println("-----");
172
173     if (red == 0 && green == 0 && blue == 0) {
174         strip.clear();
175         strip.show();
176         Serial.println("Turn Off All");
177     }
178     else {
179         strip.clear();
180         strip.show();
181
182         int first = 0;          //first led index
183         int count = 0;          //number of lights turned on
184         switch (radius) {
185             case 1:
186                 first = 254;
187                 count = 1;
188                 break;
189             case 2:
190                 first = 248;
191                 count = 7;
192                 break;
193             case 3:
194                 first = 236;
195                 count = 19;
196                 break;
197             case 4:
198                 first = 216;
199                 count = 39;
200                 break;
201             case 5:
202                 first = 192;
203                 count = 63;
204                 break;
205             case 6:
206                 first = 164;
207                 count = 91;
208                 break;
209             case 7:
210                 first = 132;
211                 count = 123;
212                 break;
213             case 8:
214                 first = 92;
215                 count = 163;
216                 break;
217             case 9:
218                 first = 48;

```

```

219         count = 207;
220         break;
221     case 10:
222         first = 0;
223         count = 255;
224         break;
225     default:
226         first = 0;
227         count = 255;
228         Serial.println("Num is Out of Range");
229         break;
230 }
231 color = strip.Color(green, red, blue);
232 strip.fill(color, first, count);
233 strip.show(); // Refresh strip
234 }
235 }
236
237 /* dark field function */
238 void dark_field() {
239
240     // calculate optical parameters
241     float theta = asin(NAobj);
242     double x = d * tan(theta);
243     int radius = int(x / Delta_X) + 1; //bright field diameter
244
245     // read and print light color
246     String Col = Serial.readStringUntil(terminator);
247     int red = (Col.substring(1, 4)).toInt();
248     int green = (Col.substring(5, 8)).toInt();
249     int blue = (Col.substring(9, 12)).toInt();
250     Serial.print("RED: ");
251     Serial.print(red);
252     Serial.print(" GREEN: ");
253     Serial.print(green);
254     Serial.print(" BLUE: ");
255     Serial.println(blue);
256     Serial.println("-----");
257
258     if (red == 0 && green == 0 && blue == 0) {
259         strip.clear();
260         strip.show();
261         Serial.println("Turn Off All");
262     }
263     else {
264         strip.clear();
265         strip.show();
266
267         int first = 0; //first led index
268         int count = 0; //number of lights turned on
269         switch (radius) {
270             case 1:
271                 first = 0;
272                 count = 254;
273                 break;
274             case 2:
275                 first = 0;
276                 count = 248;
277                 break;
278             case 3:
279                 first = 0;
280                 count = 236;
281                 break;
282             case 4:
283                 first = 0;
284                 count = 216;
285                 break;
286             case 5:
287                 first = 0;
288                 count = 192;
289                 break;
290             case 6:
291                 first = 0;

```

```

292         count = 164;
293         break;
294     case 7:
295         first = 0;
296         count = 132;
297         break;
298     case 8:
299         first = 0;
300         count = 92;
301         break;
302     case 9:
303         first = 0;
304         count = 48;
305         break;
306     case 10:
307         first = 0;
308         count = 1;
309         break;
310     default:
311         first = 0;
312         count = 1;
313         Serial.println("Num is Out of Range");
314         break;
315     }
316     color = strip.Color(green, red, blue);
317     strip.fill(color, first, count);
318     strip.show(); // Refresh strip
319 }
320 }
321
322 /*single half-ring function for phase contrast function*/
323 void single_half_ring(int num, int red, int green, int blue, char t) {
324     int first = 0; //first led index
325     int count = 0; //number of lights turned on
326     color = strip.Color(green, red, blue);
327     switch (t) {
328         case 't':
329             switch (num) { //ring number
330                 case 1:
331                     first = 254;
332                     count = 1;
333                     break;
334                 case 2:
335                     first = 248;
336                     count = 4;
337                     break;
338                 case 3:
339                     first = 236;
340                     count = 7;
341                     break;
342                 case 4:
343                     first = 216;
344                     count = 11;
345                     break;
346                 case 5:
347                     first = 192;
348                     count = 13;
349                     break;
350                 case 6:
351                     first = 164;
352                     count = 15;
353                     break;
354                 case 7:
355                     first = 132;
356                     count = 17;
357                     break;
358                 case 8:
359                     first = 92;
360                     count = 21;
361                     break;
362                 case 9:
363                     first = 48;
364                     count = 23;

```

```

365         break;
366     case 10:
367         first = 0;
368         count = 25;
369         break;
370     default:
371         Serial.println("Num is Out of Range");
372         break;
373 }
374 strip.fill(color, first, count);
375 strip.show(); // Refresh strip
376 break;
377 case 'b':
378     switch (num) {
379         case 1:
380             first = 254;
381             count = 1;
382             break;
383         case 2:
384             strip.setPixelColor(248, color);
385             first = 251;
386             count = 3;
387             break;
388         case 3:
389             strip.setPixelColor(236, color);
390             first = 242;
391             count = 6;
392             break;
393         case 4:
394             strip.setPixelColor(216, color);
395             first = 226;
396             count = 10;
397             break;
398         case 5:
399             strip.setPixelColor(192, color);
400             first = 204;
401             count = 12;
402             break;
403         case 6:
404             strip.setPixelColor(164, color);
405             first = 178;
406             count = 14;
407             break;
408         case 7:
409             strip.setPixelColor(132, color);
410             first = 148;
411             count = 16;
412             break;
413         case 8:
414             strip.setPixelColor(92, color);
415             first = 112;
416             count = 20;
417             break;
418         case 9:
419             strip.setPixelColor(48, color);
420             first = 70;
421             count = 22;
422             break;
423         case 10:
424             strip.setPixelColor(0, color);
425             first = 24;
426             count = 24;
427             break;
428         default:
429             Serial.println("Num is Out of Range");
430             break;
431     }
432     strip.fill(color, first, count);
433     strip.show(); // Refresh strip
434     break;
435 case 'l':
436     switch (num) {
437         case 1:

```

```
438         first = 254;
439         count = 1;
440         strip.fill(color, first, count);
441         break;
442     case 2:
443         first = 248;
444         count = 2;
445         strip.fill(color, first, count);
446         first = 253;
447         count = 1;
448         strip.fill(color, first, count);
449         break;
450     case 3:
451         first = 236;
452         count = 4;
453         strip.fill(color, first, count);
454         first = 245;
455         count = 3;
456         strip.fill(color, first, count);
457         break;
458     case 4:
459         first = 216;
460         count = 6;
461         strip.fill(color, first, count);
462         first = 231;
463         count = 5;
464         strip.fill(color, first, count);
465         break;
466     case 5:
467         first = 192;
468         count = 7;
469         strip.fill(color, first, count);
470         first = 210;
471         count = 6;
472         strip.fill(color, first, count);
473         break;
474     case 6:
475         first = 164;
476         count = 8;
477         strip.fill(color, first, count);
478         first = 185;
479         count = 7;
480         strip.fill(color, first, count);
481         break;
482     case 7:
483         first = 132;
484         count = 9;
485         strip.fill(color, first, count);
486         first = 156;
487         count = 8;
488         strip.fill(color, first, count);
489         break;
490     case 8:
491         first = 92;
492         count = 11;
493         strip.fill(color, first, count);
494         first = 122;
495         count = 10;
496         strip.fill(color, first, count);
497         break;
498     case 9:
499         first = 48;
500         count = 12;
501         strip.fill(color, first, count);
502         first = 81;
503         count = 11;
504         strip.fill(color, first, count);
505         break;
506     case 10:
507         first = 0;
508         count = 13;
509         strip.fill(color, first, count);
510         first = 36;
```

```

511         count = 12;
512         strip.fill(color, first, count);
513         break;
514     default:
515         Serial.println("Num is Out of Range");
516         break;
517     }
518     strip.show(); // Refresh strip
519     break;
520     case 'r':
521         switch (num) {
522             case 1:
523                 first = 254;
524                 count = 1;
525                 break;
526             case 2:
527                 first = 250;
528                 count = 3;
529                 break;
530             case 3:
531                 first = 239;
532                 count = 7;
533                 break;
534             case 4:
535                 first = 221;
536                 count = 11;
537                 break;
538             case 5:
539                 first = 198;
540                 count = 13;
541                 break;
542             case 6:
543                 first = 171;
544                 count = 15;
545                 break;
546             case 7:
547                 first = 140;
548                 count = 17;
549                 break;
550             case 8:
551                 first = 102;
552                 count = 21;
553                 break;
554             case 9:
555                 first = 59;
556                 count = 23;
557                 break;
558             case 10:
559                 first = 12;
560                 count = 25;
561                 break;
562             default:
563                 Serial.println("Num is Out of Range");
564                 break;
565         }
566         strip.fill(color, first, count);
567         strip.show(); // Refresh strip
568         break;
569     }
570 }
571
572 /* phase contrast function */
573 void phase_contrast() {
574     strip.clear();
575     strip.show();
576
577     // calculate optical parameters
578     float theta = asin(NAobj);
579     double x = d * tan(theta);
580     int radius = int(x / Delta_X) + 1; //bright field diameter
581
582     // read and print light color
583     String Col = Serial.readStringUntil(terminator);

```



```

584 int red = (Col.substring(1, 4)).toInt();
585 int green = (Col.substring(5, 8)).toInt();
586 int blue = (Col.substring(9, 12)).toInt();
587 Serial.print("RED: ");
588 Serial.print(red);
589 Serial.print(" GREEN: ");
590 Serial.print(green);
591 Serial.print(" BLUE: ");
592 Serial.println(blue);
593
594 // read and print phase direction
595 String Dir = Serial.readStringUntil(terminator);
596 Serial.print("DIRECTION: ");
597 Serial.println(Dir);
598 Serial.println("-----");
599
600 int val = 0;
601 if (Dir == "bt") val = 1; //bright field top
602 else if (Dir == "bb") val = 2; //bright field bottom
603 else if (Dir == "bl") val = 3; //bright field left
604 else if (Dir == "br") val = 4; //bright field right
605 else if (Dir == "dt") val = 5; //dark field top
606 else if (Dir == "db") val = 6; //dark field bottom
607 else if (Dir == "dl") val = 7; //dark field left
608 else if (Dir == "dr") val = 8; //dark field right
609 else Serial.println("Direction is Wrong");
610
611 int k;
612 switch (val) {
613     case 1:
614         for (k = 1 ; k <= radius; k++ ) {
615             single_half_ring(k, red, green, blue, 't');
616         }
617         strip.show(); // Refresh strip
618         break;
619     case 2:
620         for (k = 1 ; k <= radius; k++ ) {
621             single_half_ring(k, red, green, blue, 'b');
622         }
623         strip.show(); // Refresh strip
624         break;
625     case 3:
626         for (k = 1 ; k <= radius; k++ ) {
627             single_half_ring(k, red, green, blue, 'l');
628         }
629         strip.show(); // Refresh strip
630         break;
631     case 4:
632         for (k = 1 ; k <= radius; k++ ) {
633             single_half_ring(k, red, green, blue, 'r');
634         }
635         strip.show(); // Refresh strip
636         break;
637     case 5:
638         for (k = radius + 1 ; k <= 10; k++ ) {
639             single_half_ring(k, red, green, blue, 't');
640         }
641         strip.show(); // Refresh strip
642         break;
643     case 6:
644         for (k = radius + 1 ; k <= 10; k++ ) {
645             single_half_ring(k, red, green, blue, 'b');
646         }
647         break;
648     case 7:
649         for (k = radius + 1 ; k <= 10; k++ ) {
650             single_half_ring(k, red, green, blue, 'l');
651         }
652         break;
653     case 8:
654         for (k = radius + 1 ; k <= 10; k++ ) {
655             single_half_ring(k, red, green, blue, 'r');
656         }

```

```

657         break;
658     }
659 }
660
661 /* multiple points function */
662 void multiple_points() {
663     strip.clear();
664     strip.show();
665
666     // read and print number of points
667     int num = Serial.readStringUntil(terminator).toInt();
668     Serial.print("Points_Number: ");
669     Serial.println(num);
670
671     /* turn on the points */
672     for (int i = 1; i <= num; i++) {
673
674         // read and print light color and index
675         String string = Serial.readStringUntil(terminator);
676         int red = (string.substring(1, 4)).toInt();
677         int green = (string.substring(5, 8)).toInt();
678         int blue = (string.substring(9, 12)).toInt();
679         int index = (string.substring(13)).toInt();
680         Serial.print("Index: ");
681         Serial.print(index);
682         Serial.print(" RED: ");
683         Serial.print(red);
684         Serial.print(" GREEN: ");
685         Serial.print(green);
686         Serial.print(" BLUE: ");
687         Serial.println(blue);
688
689         color = strip.Color(green, red, blue);
690         strip.setPixelColor(index, color); //turn on one point
691     }
692     strip.show();
693     Serial.println("-----");
694 }
695
696 /* set parameters function */
697 void set() {
698     String string = Serial.readStringUntil(terminator);
699
700     if (string == "na" ) { // change numerical aperture
701         String number = Serial.readStringUntil(terminator);
702         float para = number.toFloat();
703         NAobj = para;
704         Serial.print("new NAobj: ");
705         Serial.println(NAobj);
706         Serial.println("-----");
707     }
708     else if (string == "dist") { //change distance from the sample
709         to Adafruit DotStar LEDs
710         String number = Serial.readStringUntil(terminator);
711         int para = number.toInt();
712         d = para;
713         Serial.print("new distance: ");
714         Serial.println(d);
715         Serial.println("-----");
716     }
717     else {
718         Serial.println("No command");
719     }
720 }
721

```