

Optical Communication Using WDM

Jerry, Jeremy, Timothy, Alicia

Cosmos 2024
Cluster 5: Photonics: Light-based
Technologies in Everyday Life

Cosmos Life



Introduction

Have you wondered how you can send and receive texts from someone miles away from you?

- Optical communication serves as the basis for quick transmissions of data
- Nearly all forms of modern communications involve Wavelength Division Multiplexing (WDM)
 - Combines different colors of light into white light and splits it back up
 - Allows large amounts of data to be transmitted at once



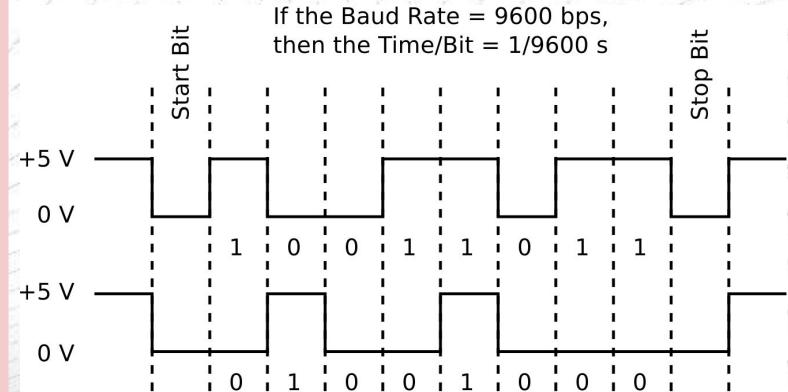
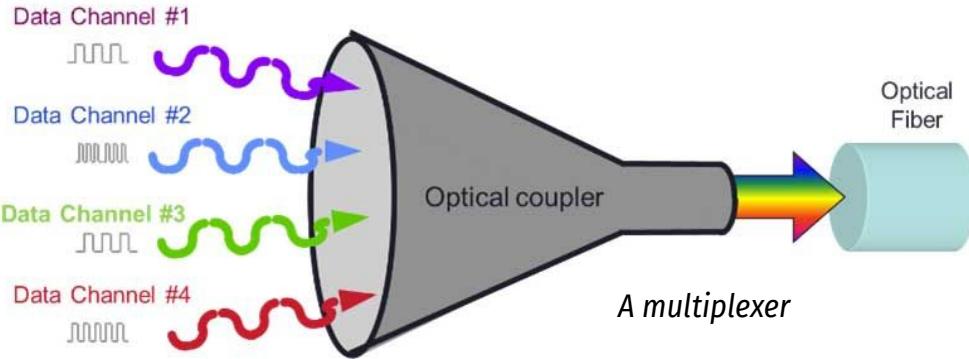
Main Goal

Test the efficiency of a free-space Wavelength Division Multiplexing system and the impact the length of a message has on the accuracy and speed of transmission.

Key Concepts

Multiplexer/Demultiplexer:

Component of WDM system that combines or separates different wavelengths of optical signals to be sent and received through a single channel, usually optical fiber

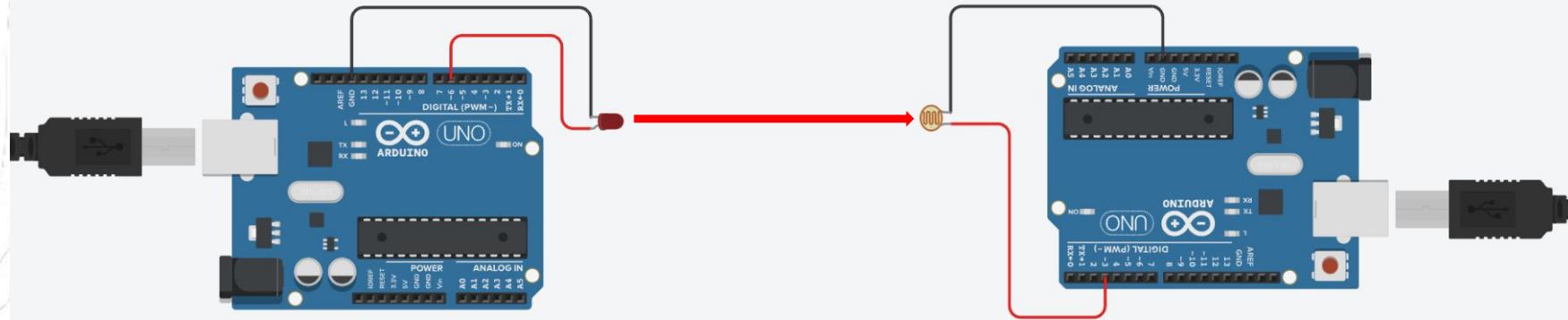


Wavelength: Determines color and energy of a light wave

Modulation: The way waves are manipulated

Photoresistor: Sensor whose resistance varies with the amount of light on its surface

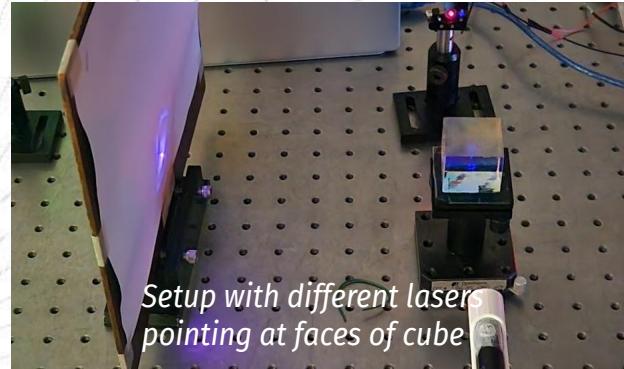
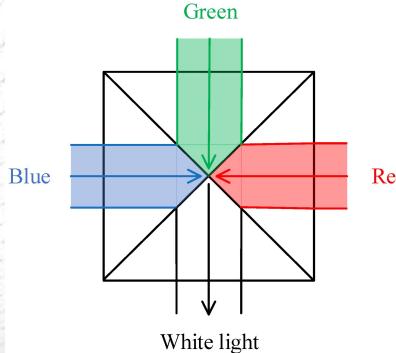
Basic Principle: One-Beam System



- Instead of a wired connection, our setup uses pulses from a laser diode
- Received by light sensor
- Challenges: alignment, signal interpretation, obstruction

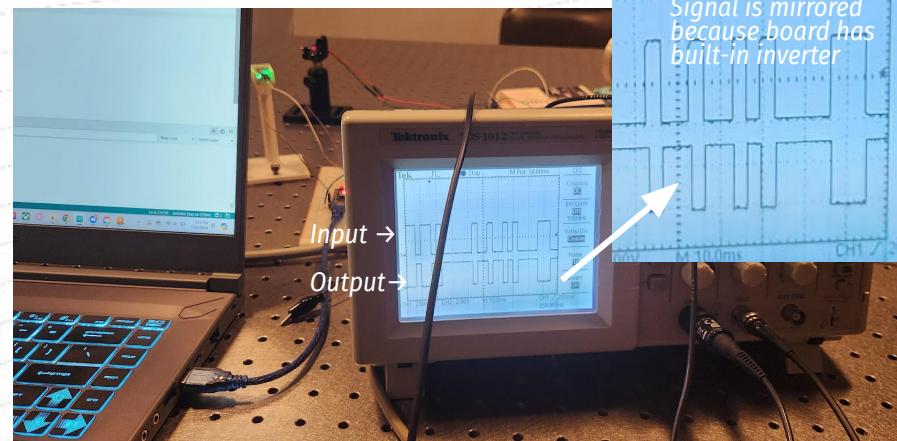
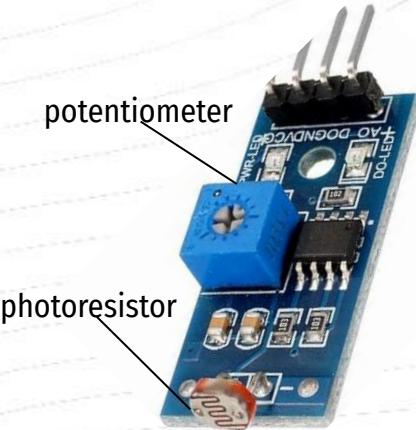
Methodology: Beam Splitting

- Used trichroic prism
 - Combines and splits beams of different colors
- Red, green, and blue combine to make white light
- Prism has coatings/materials to split beam at right angles



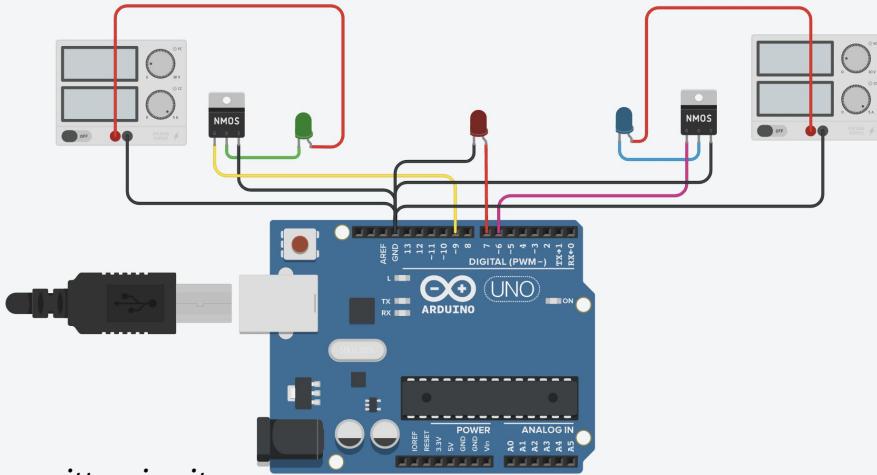
Methodology: Receiving

- Used **digital photoresistor board** to convert optical pulses to digital signal (on/off)
- Sensor contains **potentiometer** to adjust **on threshold**
 - Can tune for different wavelengths and intensities
- Designed + 3D printed stands to hold sensors steadily
- Viewed sensor output on oscilloscope

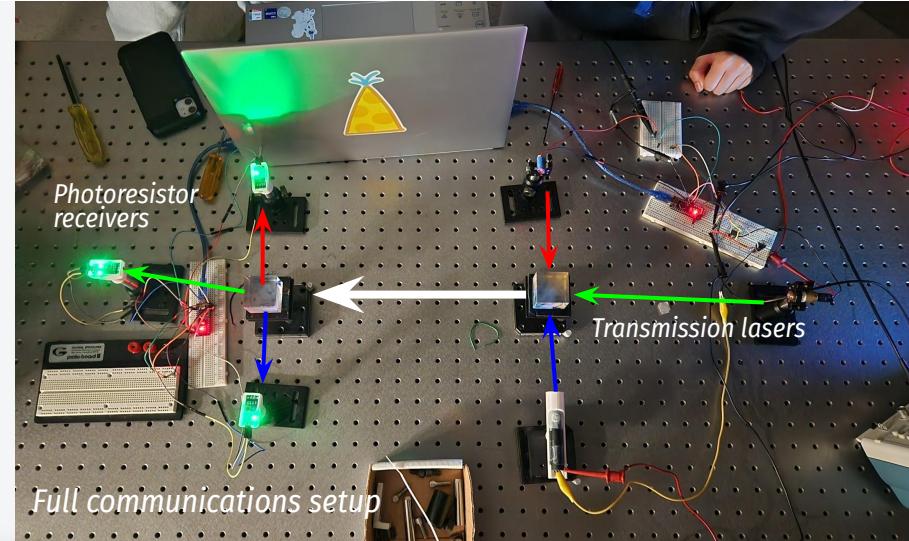


Oscilloscope reading after tuning potentiometer

Setup: Three-Beam System



Transmitter circuit



- Needed power supply and MOSFET to control green and blue lasers because of Arduino current limit
- Challenges: leveling the lasers and combining them into one beam of white light, ensuring that the blue and green laser diodes don't receive too much power

Communication Protocol

Time-Based: Each character is mapped to its ASCII code, and a pulse of a certain time is given to this character. We add a delay of 50 milliseconds for each character to ensure that even if there is a delay, it accurately reads in the time.

- Example: The character 'A' has an ASCII code of 65. Through processing: $(65 - 32 + 1) * 50 = 1700$ ms. Thus to transmit the character A, we send a laser pulse of 1700 ms

Serial-Based: We used the built in serial communication on Arduino. This data is read at a rate of $1/(\text{baud rate})$ per second on the other Arduino. In our case we use a 300 baud rate.

- Asynchronous vs Synchronous Serial Communication

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	00	NULL OR SPACES	32	20	DATA	64	40	65
1	01	START OF TEXT	33	21	!	65	41	66
2	02	END OF TEXT	34	22	?	66	42	67
3	03	END OF TRANSMISSION	35	23	DEL	67	43	68
4	04	ACKNOWLEDGE	36	24		68	44	69
5	05	ACKNOWLEDGE	37	25		69	45	70
6	06	ACKNOWLEDGE	38	26		70	46	71
7	07	ACKNOWLEDGE	39	27		71	47	72
8	08	ACKNOWLEDGE	40	28		72	48	73
9	09	ACKNOWLEDGE	41	29		73	49	74
10	0A	ACKNOWLEDGE	42	2A		74	4A	75
11	0B	ACKNOWLEDGE	43	2B		75	4B	76
12	0C	ACKNOWLEDGE	44	2C		76	4C	77
13	0D	ACKNOWLEDGE	45	2D		77	4D	78
14	0E	ACKNOWLEDGE	46	2E		78	4E	79
15	0F	ACKNOWLEDGE	47	2F		79	4F	7A
16	10	DATA LINE ESCAPE	48	30		80	50	80
17	11	DATA LINE ESCAPE	49	31		81	51	81
18	12	DATA LINE ESCAPE	50	32		82	52	82
19	13	DATA LINE ESCAPE	51	33		83	53	83
20	14	DATA LINE ESCAPE	52	34		84	54	84
21	15	DATA LINE ESCAPE	53	35		85	55	85
22	16	DATA LINE ESCAPE	54	36		86	56	86
23	17	DATA LINE ESCAPE	55	37		87	57	87
24	18	DATA LINE ESCAPE	56	38		88	58	88
25	19	END OF MESSAGE	57	39		89	59	89
26	1A	END OF MESSAGE	58	3A		90	5A	90
27	1B	END OF MESSAGE	59	3B		91	5B	91
28	1C	END OF MESSAGE	60	3C		92	5C	92
29	1D	END OF MESSAGE	61	3D		93	5D	93
30	1E	END OF MESSAGE	62	3E		94	5E	94
31	1F	END OF MESSAGE	63	3F		95	5F	95

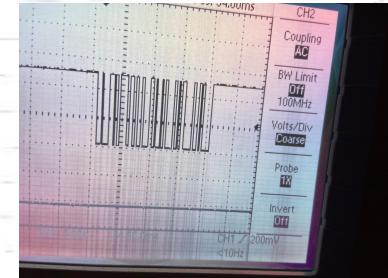
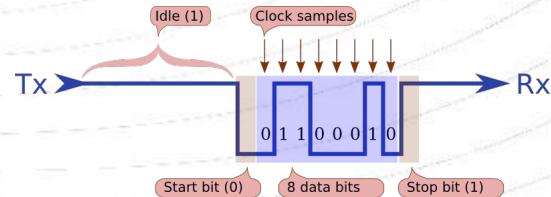
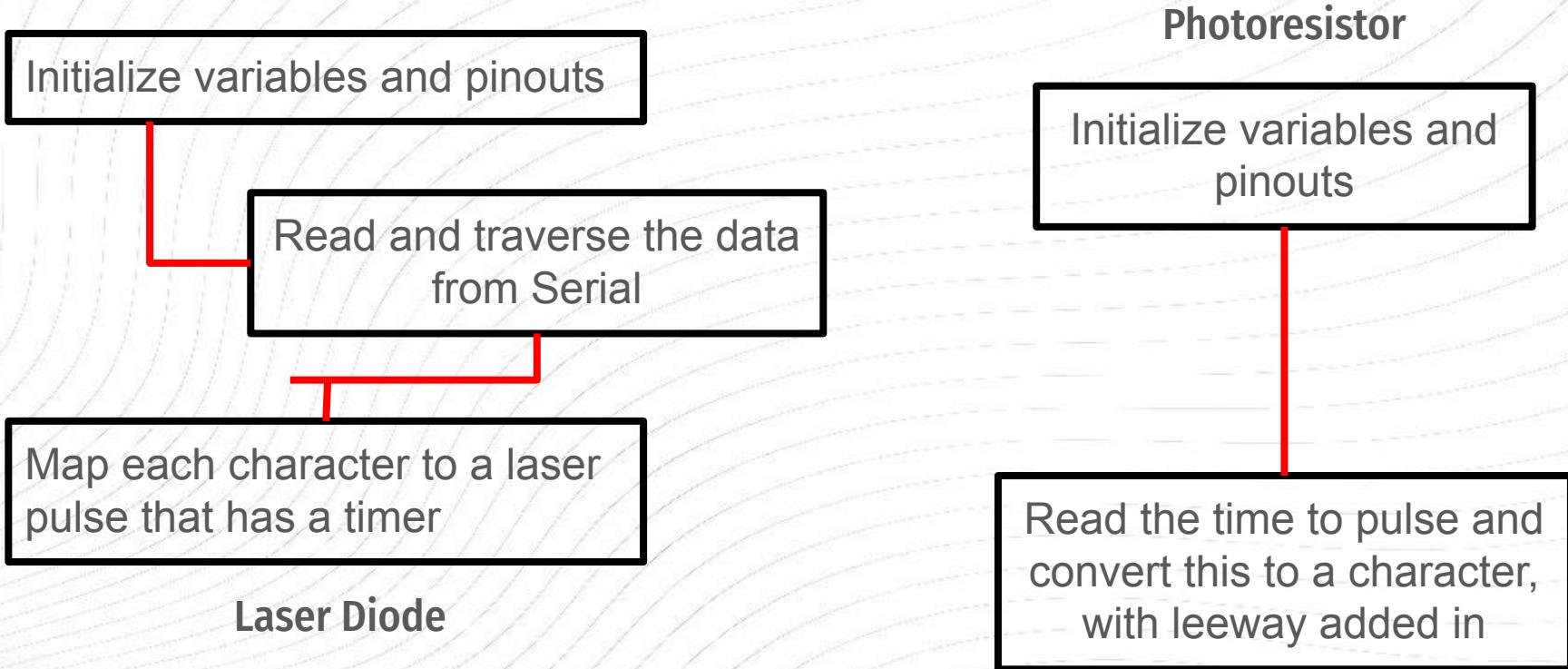


Image of Serial-Based Pulsing for "bob"

Single Beam System - Code for Time-Based Pulsing:



Single Beam System - Code for Serial-Based Pulsing:

Initialize variables and pinouts

Read and traverse the data from Serial

Use an additional programmed serial to transmit data over built in Arduino Serial

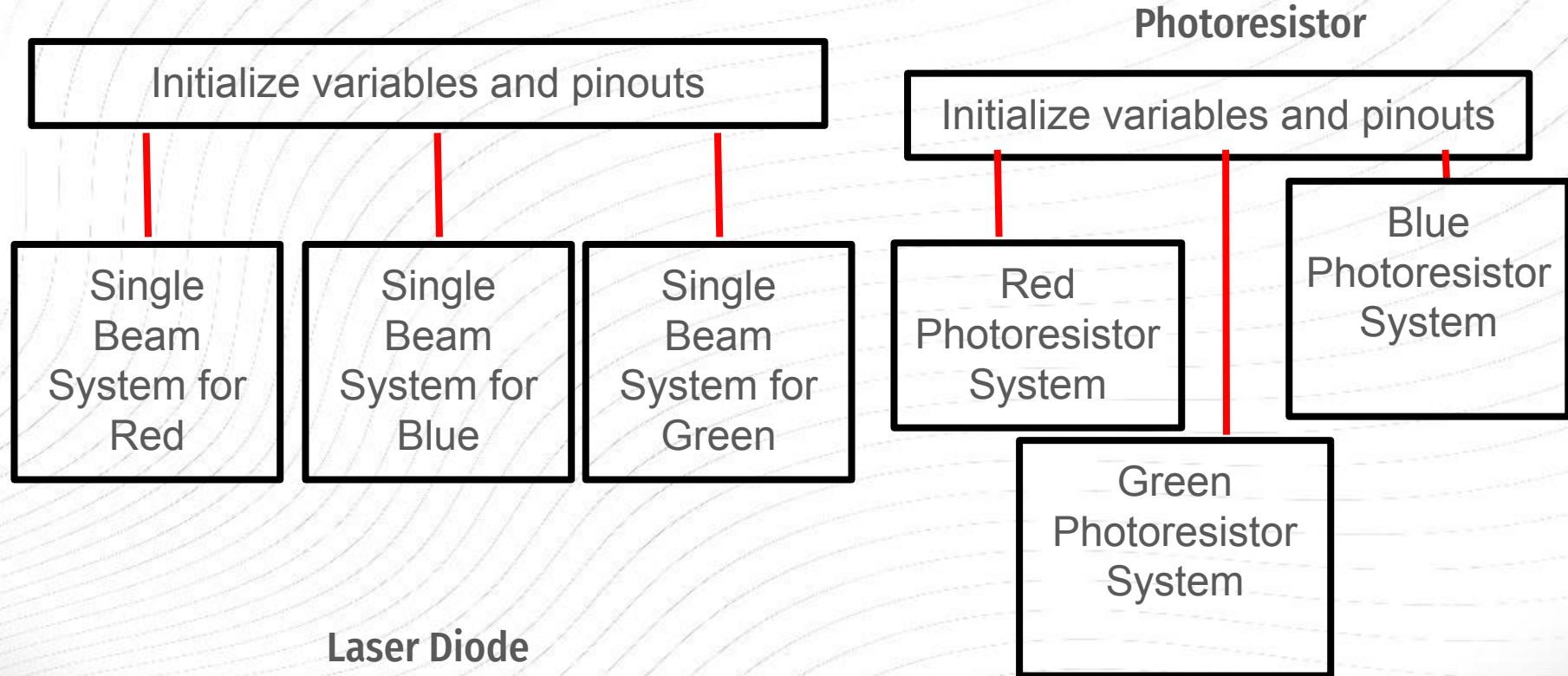
Laser Diode

Photoresistor

Initialize variables and pinouts

Each read byte transmitted is converted back to its ASCII Code value, thus a character

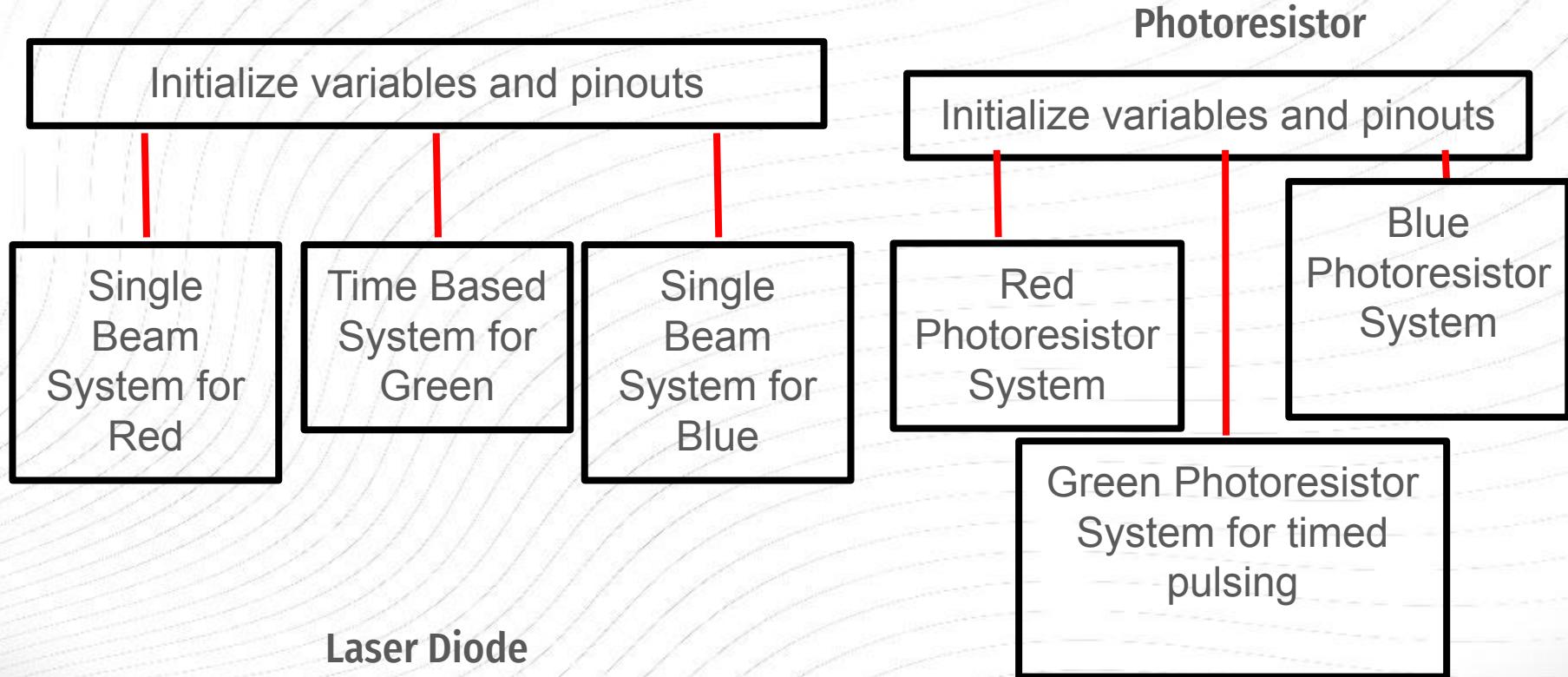
Three Beam System - Code for Serial-Based Pulsing:



Challenges

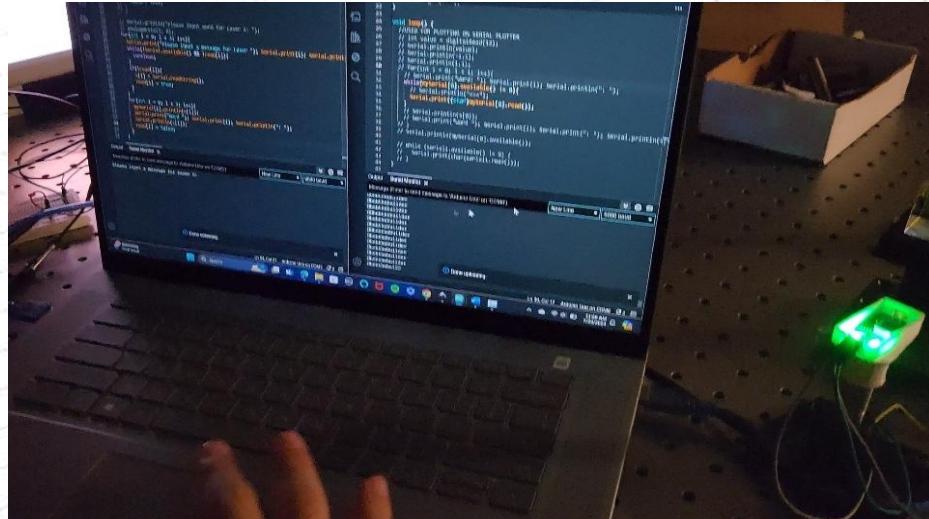
- **Precision**
 - Varies with calibration, power supplied to laser, small changes to position of equipment
- **Communication Protocol**
 - Unable to implement serial on green laser
 - Created own time-based protocol at cost of speed
- **Hardware limitations**
 - Arduino can only send one serial signal at a time
 - Not “true” WDM, but would be if using multiple arduinos for simultaneous serial transmission

Three Beam System - Corrected Version:



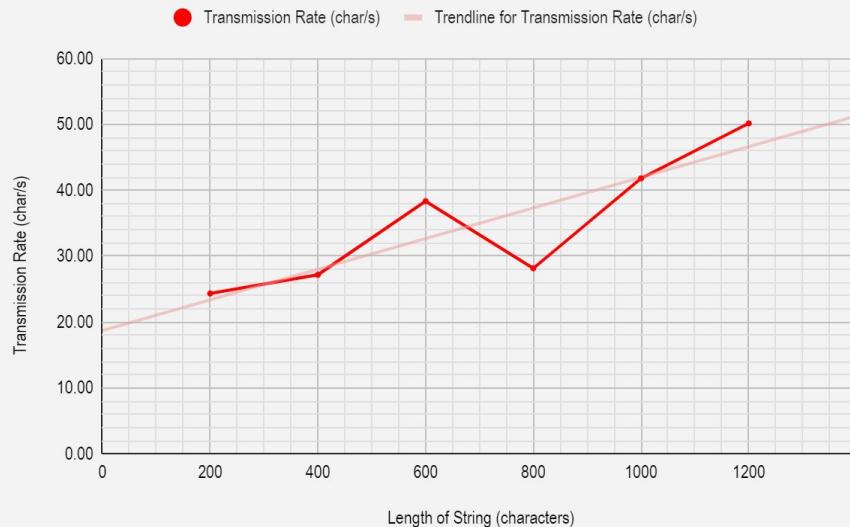
Testing: Transmission Speed

- Timed the duration from sending message to completely receiving
- Calculated transmission rate
- (number of characters/total time)
- Sent messages from 200-1000 characters (200 char intervals)



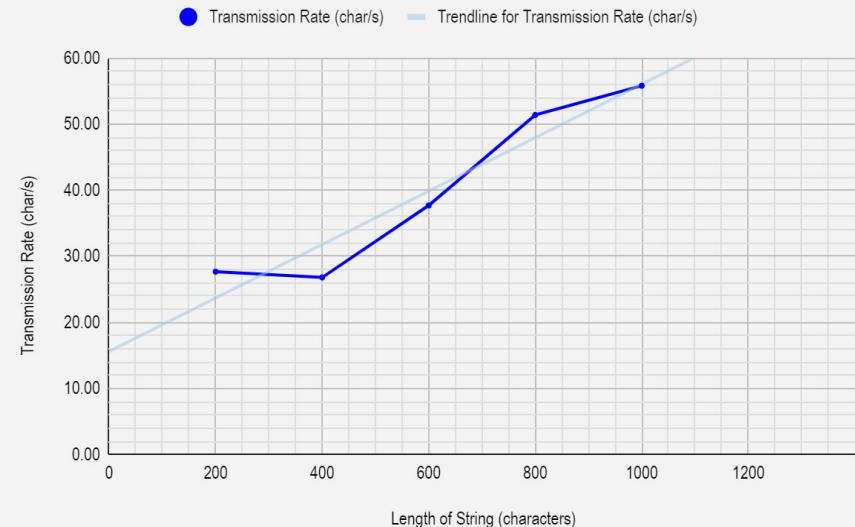
Effect of String Length on Transmission Speed

Red Laser Channel



Effect of String Length on Transmission Speed

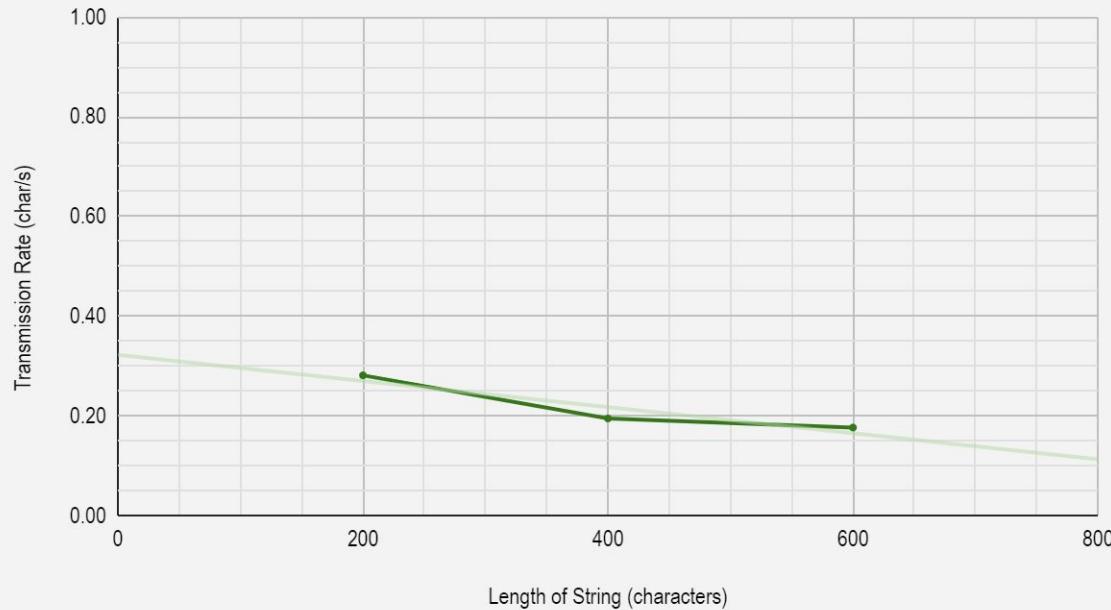
Blue Laser Channel



Effect of String Length on Transmission Speed

Green Laser Channel

● Transmission Rate (char/s) └ Trendline for Transmission Rate (char/s)



Analysis: Transmission Speed

- Red and blue channels increased in rate as character count increased
 - Implies large fixed delay
 - Each additional character has less effect on duration
 - Hardware considerations: arduino, diode
 - Rate of change of characters per second per character was higher for blue
- Green channel remained slow
 - Since each character is a unique pulse length, transmission is slow
 - Very small rate of change

Average transmission speed (characters per second): Red - 34.99, Green - 0.22, Blue - 39.87

Testing: Accuracy

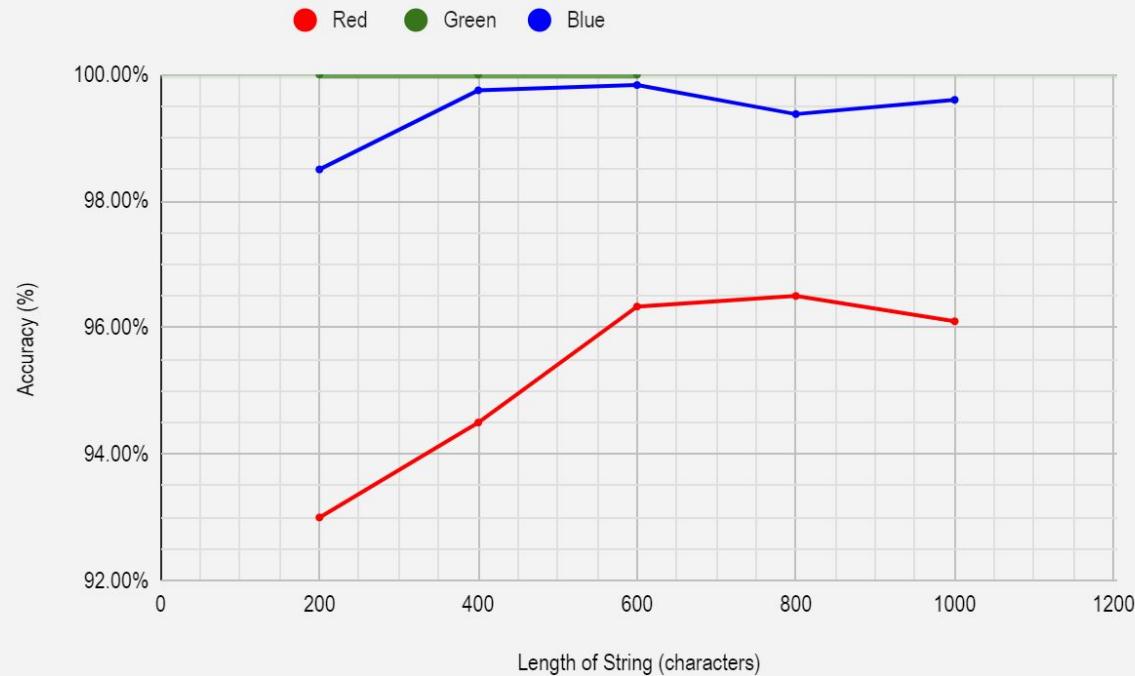
- Generated text with recognizable words
- Found number of changes from received text compared to original string
- Counts mutations, additions, subtractions
- Calculated accuracy rate: $(\text{length of original string} - \# \text{ of errors})/\text{length}$

1 Jayden was a creature of habit, his days a predictable rhythm of coffee, code, and the occasional video game. Winter was a whirlwind, her laughter a contagious melody that disrupted the quietude of his existence. They met in the most mundane of places - the office coffee machine. A spilled cup and a shared laugh later, they found themselves drawn to each other like two magnets. Winter's world was a kaleidoscope of colors, her spirit as vibrant as the sunsets she adored. Jayden, in his grayscale world, found himself drawn to her warmth. Slowly, he began to see the world through her eyes, discovering hidden beauty in the ordinary. With Winter, every day was an adventure, a departure from the familiar. Their connection deepened with each shared moment. They explored hidden city corners, discove

1 Jayden was a creature of habit, his days a predictable rhythm of coffee, code, and the occasional video game. Winter was a whirlwind, her laughter a contagious melody that disrupted the quietude of his existence. They met in the most mundane of places - the office coffee machine. A spilled cup and a shared laugh later, they found themselves drawn to each other like two magnets. Winter's world was a kaleidoscope of colors, her spirit as vibrant as the sunsets she adored. Jayden, in his grayscale world, found himself drawn to her warmth. Slowly, he began to see the world through her eyes, discovering hidden beauty in the ordinary. With Winter, every day was an adventure, a departure from the familiar. Their connection deepened with each shared moment. They explored hidden city corners, discove

Effect of String Length on Message Accuracy

All Channels



Analysis: Accuracy

- **Text intelligible**
 - Always above 92% accuracy
 - Red channel tended to get more accurate over time, avg. 95%
- **Green and Blue channels**
 - Green and blue tended to stay consistent, green was always 100%, and blue averaged at 99%
- **Very little disturbance in open air**
- **Likely would not vary with distance; subject of further investigation**
- **Error correction not implemented, but existing performance sufficient to be corrected**

Conclusion

- **WDM optical communication is a very complex and precise method**
 - We needed to make work arounds because certain lasers were not precise enough
 - Spent a lot of time aligning and combining lasers
- **Different communication protocols have their own benefits**
 - Blue and Red lasers implemented serial based pulsing which is very fast but also has more errors
 - Green laser implemented time based pulsing which gave 100% accurate but was very very slow
- **Our system is not perfect and has a lot of flaws and limitations**
 - The lasers were not made to do optical communication
 - Made the most of what we had and implemented many work arounds
- **Other things we hope to test: how disturbances like water and dust affect the transmission**

Acknowledgements

Thank you to our...

Professors: Dr. Sahar, Dr. Tu, and Dr. Ilinykh

Teacher Fellow: Mr. Barrows

TAs: Karl, Daniel, and Jay

And all of our classmates!

Arduino Code

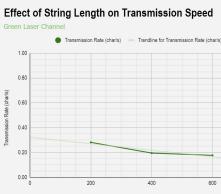
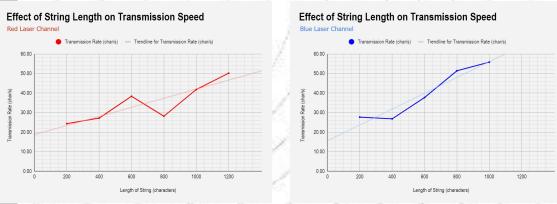
Transmission Code Receiver Code

```

1 //include <SoftwareSerial.h>
2
3 SoftwareSerial mySerial = SoftwareSerial(11, 13, SoftwareSerial(11, 13, SoftwareSerial(11, 13)));
4
5 int p = 0;
6 int mySerial1[3] = {false, false, false};
7
8 String s[3];
9
10 void setup() {
11   Serial.begin(9600);
12   for (int i = 0; i < 3; i++) {
13     mySerial1[i].begin(9600);
14   }
15 }
16
17 void loop() {
18   for (int i = 0; i < 3; i++) {
19     Serial.print("User input a message for user "); Serial.print(i); Serial.println(")");
20     while (mySerial.available() >= 1) readSerial();
21   }
22 }
23
24 void readSerial() {
25   if (Serial.available() > 0) {
26     if (Serial.read() == '1') {
27       mySerial1[0] = true;
28     }
29   }
30   for (int i = 1; i < 3; i++) {
31     mySerial1[i].print("1");
32     if (Serial.available() > 0) {
33       Serial.print("User input a message for user ");
34       Serial.println(")");
35     }
36   }
37 }
38
39 void mySerial() {
40   if (mySerial.available() > 0) {
41     Serial.print("User input a message for user ");
42     Serial.println(")");
43   }
44 }
45
46 void mySerial1() {
47   if (mySerial1[0] == true) {
48     mySerial1[0] = false;
49     mySerial1[1].begin(9600);
50     mySerial1[2].begin(9600);
51   }
52 }
53
54 void mySerial2() {
55   if (mySerial1[1] == true) {
56     mySerial1[1] = false;
57     mySerial1[2].begin(9600);
58   }
59 }
60
61 void mySerial3() {
62   if (mySerial1[2] == true) {
63     mySerial1[2] = false;
64     mySerial1[0].begin(9600);
65   }
66 }
67
68 void mySerial4() {
69   if (mySerial1[0] == true) {
70     mySerial1[0] = false;
71     mySerial1[1].begin(9600);
72   }
73 }
74
75 void mySerial5() {
76   if (mySerial1[1] == true) {
77     mySerial1[1] = false;
78     mySerial1[2].begin(9600);
79   }
80 }
81
82 void mySerial6() {
83   if (mySerial1[2] == true) {
84     mySerial1[2] = false;
85     mySerial1[0].begin(9600);
86   }
87 }
88
89 void mySerial7() {
90   if (mySerial1[0] == true) {
91     mySerial1[0] = false;
92     mySerial1[1].begin(9600);
93   }
94 }
95
96 void mySerial8() {
97   if (mySerial1[1] == true) {
98     mySerial1[1] = false;
99     mySerial1[2].begin(9600);
100 }
101 }
102
103 void mySerial9() {
104   if (mySerial1[2] == true) {
105     mySerial1[2] = false;
106     mySerial1[0].begin(9600);
107   }
108 }
109
110 void mySerial10() {
111   if (mySerial1[0] == true) {
112     mySerial1[0] = false;
113     mySerial1[1].begin(9600);
114   }
115 }
116
117 void mySerial11() {
118   if (mySerial1[1] == true) {
119     mySerial1[1] = false;
120     mySerial1[2].begin(9600);
121   }
122 }
123
124 void mySerial12() {
125   if (mySerial1[2] == true) {
126     mySerial1[2] = false;
127     mySerial1[0].begin(9600);
128   }
129 }
130
131 void mySerial13() {
132   if (mySerial1[0] == true) {
133     mySerial1[0] = false;
134     mySerial1[1].begin(9600);
135   }
136 }
137
138 void mySerial14() {
139   if (mySerial1[1] == true) {
140     mySerial1[1] = false;
141     mySerial1[2].begin(9600);
142   }
143 }
144
145 void mySerial15() {
146   if (mySerial1[2] == true) {
147     mySerial1[2] = false;
148     mySerial1[0].begin(9600);
149   }
150 }
151
152 void mySerial16() {
153   if (mySerial1[0] == true) {
154     mySerial1[0] = false;
155     mySerial1[1].begin(9600);
156   }
157 }
158
159 void mySerial17() {
160   if (mySerial1[1] == true) {
161     mySerial1[1] = false;
162     mySerial1[2].begin(9600);
163   }
164 }
165
166 void mySerial18() {
167   if (mySerial1[2] == true) {
168     mySerial1[2] = false;
169     mySerial1[0].begin(9600);
170   }
171 }
172
173 void mySerial19() {
174   if (mySerial1[0] == true) {
175     mySerial1[0] = false;
176     mySerial1[1].begin(9600);
177   }
178 }
179
180 void mySerial20() {
181   if (mySerial1[1] == true) {
182     mySerial1[1] = false;
183     mySerial1[2].begin(9600);
184   }
185 }
186
187 void mySerial21() {
188   if (mySerial1[2] == true) {
189     mySerial1[2] = false;
190     mySerial1[0].begin(9600);
191   }
192 }
193
194 void mySerial22() {
195   if (mySerial1[0] == true) {
196     mySerial1[0] = false;
197     mySerial1[1].begin(9600);
198   }
199 }
200
201 void mySerial23() {
202   if (mySerial1[1] == true) {
203     mySerial1[1] = false;
204     mySerial1[2].begin(9600);
205   }
206 }
207
208 void mySerial24() {
209   if (mySerial1[2] == true) {
210     mySerial1[2] = false;
211     mySerial1[0].begin(9600);
212   }
213 }
214
215 void mySerial25() {
216   if (mySerial1[0] == true) {
217     mySerial1[0] = false;
218     mySerial1[1].begin(9600);
219   }
220 }
221
222 void mySerial26() {
223   if (mySerial1[1] == true) {
224     mySerial1[1] = false;
225     mySerial1[2].begin(9600);
226   }
227 }
228
229 void mySerial27() {
230   if (mySerial1[2] == true) {
231     mySerial1[2] = false;
232     mySerial1[0].begin(9600);
233   }
234 }
235
236 void mySerial28() {
237   if (mySerial1[0] == true) {
238     mySerial1[0] = false;
239     mySerial1[1].begin(9600);
240   }
241 }
242
243 void mySerial29() {
244   if (mySerial1[1] == true) {
245     mySerial1[1] = false;
246     mySerial1[2].begin(9600);
247   }
248 }
249
250 void mySerial30() {
251   if (mySerial1[2] == true) {
252     mySerial1[2] = false;
253     mySerial1[0].begin(9600);
254   }
255 }
256
257 void mySerial31() {
258   if (mySerial1[0] == true) {
259     mySerial1[0] = false;
260     mySerial1[1].begin(9600);
261   }
262 }
263
264 void mySerial32() {
265   if (mySerial1[1] == true) {
266     mySerial1[1] = false;
267     mySerial1[2].begin(9600);
268   }
269 }
270
271 void mySerial33() {
272   if (mySerial1[2] == true) {
273     mySerial1[2] = false;
274     mySerial1[0].begin(9600);
275   }
276 }
277
278 void mySerial34() {
279   if (mySerial1[0] == true) {
280     mySerial1[0] = false;
281     mySerial1[1].begin(9600);
282   }
283 }
284
285 void mySerial35() {
286   if (mySerial1[1] == true) {
287     mySerial1[1] = false;
288     mySerial1[2].begin(9600);
289   }
290 }
291
292 void mySerial36() {
293   if (mySerial1[2] == true) {
294     mySerial1[2] = false;
295     mySerial1[0].begin(9600);
296   }
297 }
298
299 void mySerial37() {
300   if (mySerial1[0] == true) {
301     mySerial1[0] = false;
302     mySerial1[1].begin(9600);
303   }
304 }
305
306 void mySerial38() {
307   if (mySerial1[1] == true) {
308     mySerial1[1] = false;
309     mySerial1[2].begin(9600);
310   }
311 }
312
313 void mySerial39() {
314   if (mySerial1[2] == true) {
315     mySerial1[2] = false;
316     mySerial1[0].begin(9600);
317   }
318 }
319
320 void mySerial40() {
321   if (mySerial1[0] == true) {
322     mySerial1[0] = false;
323     mySerial1[1].begin(9600);
324   }
325 }
326
327 void mySerial41() {
328   if (mySerial1[1] == true) {
329     mySerial1[1] = false;
330     mySerial1[2].begin(9600);
331   }
332 }
333
334 void mySerial42() {
335   if (mySerial1[2] == true) {
336     mySerial1[2] = false;
337     mySerial1[0].begin(9600);
338   }
339 }
340
341 void mySerial43() {
342   if (mySerial1[0] == true) {
343     mySerial1[0] = false;
344     mySerial1[1].begin(9600);
345   }
346 }
347
348 void mySerial44() {
349   if (mySerial1[1] == true) {
350     mySerial1[1] = false;
351     mySerial1[2].begin(9600);
352   }
353 }
354
355 void mySerial45() {
356   if (mySerial1[2] == true) {
357     mySerial1[2] = false;
358     mySerial1[0].begin(9600);
359   }
360 }
361
362 void mySerial46() {
363   if (mySerial1[0] == true) {
364     mySerial1[0] = false;
365     mySerial1[1].begin(9600);
366   }
367 }
368
369 void mySerial47() {
370   if (mySerial1[1] == true) {
371     mySerial1[1] = false;
372     mySerial1[2].begin(9600);
373   }
374 }
375
376 void mySerial48() {
377   if (mySerial1[2] == true) {
378     mySerial1[2] = false;
379     mySerial1[0].begin(9600);
380   }
381 }
382
383 void mySerial49() {
384   if (mySerial1[0] == true) {
385     mySerial1[0] = false;
386     mySerial1[1].begin(9600);
387   }
388 }
389
390 void mySerial50() {
391   if (mySerial1[1] == true) {
392     mySerial1[1] = false;
393     mySerial1[2].begin(9600);
394   }
395 }
396
397 void mySerial51() {
398   if (mySerial1[2] == true) {
399     mySerial1[2] = false;
400     mySerial1[0].begin(9600);
401   }
402 }
403
404 void mySerial52() {
405   if (mySerial1[0] == true) {
406     mySerial1[0] = false;
407     mySerial1[1].begin(9600);
408   }
409 }
410
411 void mySerial53() {
412   if (mySerial1[1] == true) {
413     mySerial1[1] = false;
414     mySerial1[2].begin(9600);
415   }
416 }
417
418 void mySerial54() {
419   if (mySerial1[2] == true) {
420     mySerial1[2] = false;
421     mySerial1[0].begin(9600);
422   }
423 }
424
425 void mySerial55() {
426   if (mySerial1[0] == true) {
427     mySerial1[0] = false;
428     mySerial1[1].begin(9600);
429   }
430 }
431
432 void mySerial56() {
433   if (mySerial1[1] == true) {
434     mySerial1[1] = false;
435     mySerial1[2].begin(9600);
436   }
437 }
438
439 void mySerial57() {
440   if (mySerial1[2] == true) {
441     mySerial1[2] = false;
442     mySerial1[0].begin(9600);
443   }
444 }
445
446 void mySerial58() {
447   if (mySerial1[0] == true) {
448     mySerial1[0] = false;
449     mySerial1[1].begin(9600);
450   }
451 }
452
453 void mySerial59() {
454   if (mySerial1[1] == true) {
455     mySerial1[1] = false;
456     mySerial1[2].begin(9600);
457   }
458 }
459
460 void mySerial60() {
461   if (mySerial1[2] == true) {
462     mySerial1[2] = false;
463     mySerial1[0].begin(9600);
464   }
465 }
466
467 void mySerial61() {
468   if (mySerial1[0] == true) {
469     mySerial1[0] = false;
470     mySerial1[1].begin(9600);
471   }
472 }
473
474 void mySerial62() {
475   if (mySerial1[1] == true) {
476     mySerial1[1] = false;
477     mySerial1[2].begin(9600);
478   }
479 }
480
481 void mySerial63() {
482   if (mySerial1[2] == true) {
483     mySerial1[2] = false;
484     mySerial1[0].begin(9600);
485   }
486 }
487
488 void mySerial64() {
489   if (mySerial1[0] == true) {
490     mySerial1[0] = false;
491     mySerial1[1].begin(9600);
492   }
493 }
494
495 void mySerial65() {
496   if (mySerial1[1] == true) {
497     mySerial1[1] = false;
498     mySerial1[2].begin(9600);
499   }
500 }
501
502 void mySerial66() {
503   if (mySerial1[2] == true) {
504     mySerial1[2] = false;
505     mySerial1[0].begin(9600);
506   }
507 }
508
509 void mySerial67() {
510   if (mySerial1[0] == true) {
511     mySerial1[0] = false;
512     mySerial1[1].begin(9600);
513   }
514 }
515
516 void mySerial68() {
517   if (mySerial1[1] == true) {
518     mySerial1[1] = false;
519     mySerial1[2].begin(9600);
520   }
521 }
522
523 void mySerial69() {
524   if (mySerial1[2] == true) {
525     mySerial1[2] = false;
526     mySerial1[0].begin(9600);
527   }
528 }
529
530 void mySerial70() {
531   if (mySerial1[0] == true) {
532     mySerial1[0] = false;
533     mySerial1[1].begin(9600);
534   }
535 }
536
537 void mySerial71() {
538   if (mySerial1[1] == true) {
539     mySerial1[1] = false;
540     mySerial1[2].begin(9600);
541   }
542 }
543
544 void mySerial72() {
545   if (mySerial1[2] == true) {
546     mySerial1[2] = false;
547     mySerial1[0].begin(9600);
548   }
549 }
550
551 void mySerial73() {
552   if (mySerial1[0] == true) {
553     mySerial1[0] = false;
554     mySerial1[1].begin(9600);
555   }
556 }
557
558 void mySerial74() {
559   if (mySerial1[1] == true) {
560     mySerial1[1] = false;
561     mySerial1[2].begin(9600);
562   }
563 }
564
565 void mySerial75() {
566   if (mySerial1[2] == true) {
567     mySerial1[2] = false;
568     mySerial1[0].begin(9600);
569   }
570 }
571
572 void mySerial76() {
573   if (mySerial1[0] == true) {
574     mySerial1[0] = false;
575     mySerial1[1].begin(9600);
576   }
577 }
578
579 void mySerial77() {
580   if (mySerial1[1] == true) {
581     mySerial1[1] = false;
582     mySerial1[2].begin(9600);
583   }
584 }
585
586 void mySerial78() {
587   if (mySerial1[2] == true) {
588     mySerial1[2] = false;
589     mySerial1[0].begin(9600);
590   }
591 }
592
593 void mySerial79() {
594   if (mySerial1[0] == true) {
595     mySerial1[0] = false;
596     mySerial1[1].begin(9600);
597   }
598 }
599
600 void mySerial80() {
601   if (mySerial1[1] == true) {
602     mySerial1[1] = false;
603     mySerial1[2].begin(9600);
604   }
605 }
606
607 void mySerial81() {
608   if (mySerial1[2] == true) {
609     mySerial1[2] = false;
610     mySerial1[0].begin(9600);
611   }
612 }
613
614 void mySerial82() {
615   if (mySerial1[0] == true) {
616     mySerial1[0] = false;
617     mySerial1[1].begin(9600);
618   }
619 }
620
621 void mySerial83() {
622   if (mySerial1[1] == true) {
623     mySerial1[1] = false;
624     mySerial1[2].begin(9600);
625   }
626 }
627
628 void mySerial84() {
629   if (mySerial1[2] == true) {
630     mySerial1[2] = false;
631     mySerial1[0].begin(9600);
632   }
633 }
634
635 void mySerial85() {
636   if (mySerial1[0] == true) {
637     mySerial1[0] = false;
638     mySerial1[1].begin(9600);
639   }
640 }
641
642 void mySerial86() {
643   if (mySerial1[1] == true) {
644     mySerial1[1] = false;
645     mySerial1[2].begin(9600);
646   }
647 }
648
649 void mySerial87() {
650   if (mySerial1[2] == true) {
651     mySerial1[2] = false;
652     mySerial1[0].begin(9600);
653   }
654 }
655
656 void mySerial88() {
657   if (mySerial1[0] == true) {
658     mySerial1[0] = false;
659     mySerial1[1].begin(9600);
660   }
661 }
662
663 void mySerial89() {
664   if (mySerial1[1] == true) {
665     mySerial1[1] = false;
666     mySerial1[2].begin(9600);
667   }
668 }
669
670 void mySerial90() {
671   if (mySerial1[2] == true) {
672     mySerial1[2] = false;
673     mySerial1[0].begin(9600);
674   }
675 }
676
677 void mySerial91() {
678   if (mySerial1[0] == true) {
679     mySerial1[0] = false;
680     mySerial1[1].begin(9600);
681   }
682 }
683
684 void mySerial92() {
685   if (mySerial1[1] == true) {
686     mySerial1[1] = false;
687     mySerial1[2].begin(9600);
688   }
689 }
690
691 void mySerial93() {
692   if (mySerial1[2] == true) {
693     mySerial1[2] = false;
694     mySerial1[0].begin(9600);
695   }
696 }
697
698 void mySerial94() {
699   if (mySerial1[0] == true) {
700     mySerial1[0] = false;
701     mySerial1[1].begin(9600);
702   }
703 }
704
705 void mySerial95() {
706   if (mySerial1[1] == true) {
707     mySerial1[1] = false;
708     mySerial1[2].begin(9600);
709   }
710 }
711
712 void mySerial96() {
713   if (mySerial1[2] == true) {
714     mySerial1[2] = false;
715     mySerial1[0].begin(9600);
716   }
717 }
718
719 void mySerial97() {
720   if (mySerial1[0] == true) {
721     mySerial1[0] = false;
722     mySerial1[1].begin(9600);
723   }
724 }
725
726 void mySerial98() {
727   if (mySerial1[1] == true) {
728     mySerial1[1] = false;
729     mySerial1[2].begin(9600);
730   }
731 }
732
733 void mySerial99() {
734   if (mySerial1[2] == true) {
735     mySerial1[2] = false;
736     mySerial1[0].begin(9600);
737   }
738 }
739
740 void mySerial100() {
741   if (mySerial1[0] == true) {
742     mySerial1[0] = false;
743     mySerial1[1].begin(9600);
744   }
745 }
746
747 void mySerial101() {
748   if (mySerial1[1] == true) {
749     mySerial1[1] = false;
750     mySerial1[2].begin(9600);
751   }
752 }
753
754 void mySerial102() {
755   if (mySerial1[2] == true) {
756     mySerial1[2] = false;
757     mySerial1[0].begin(9600);
758   }
759 }
760
761 void mySerial103() {
762   if (mySerial1[0] == true) {
763     mySerial1[0] = false;
764     mySerial1[1].begin(9600);
765   }
766 }
767
768 void mySerial104() {
769   if (mySerial1[1] == true) {
770     mySerial1[1] = false;
771     mySerial1[2].begin(9600);
772   }
773 }
774
775 void mySerial105() {
776   if (mySerial1[2] == true) {
777     mySerial1[2] = false;
778     mySerial1[0].begin(9600);
779   }
780 }
781
782 void mySerial106() {
783   if (mySerial1[0] == true) {
784     mySerial1[0] = false;
785     mySerial1[1].begin(9600);
786   }
787 }
788
789 void mySerial107() {
790   if (mySerial1[1] == true) {
791     mySerial1[1] = false;
792     mySerial1[2].begin(9600);
793   }
794 }
795
796 void mySerial108() {
797   if (mySerial1[2] == true) {
798     mySerial1[2] = false;
799     mySerial1[0].begin(9600);
800   }
801 }
802
803 void mySerial109() {
804   if (mySerial1[0] == true) {
805     mySerial1[0] = false;
806     mySerial1[1].begin(9600);
807   }
808 }
809
810 void mySerial110() {
811   if (mySerial1[1] == true) {
812     mySerial1[1] = false;
813     mySerial1[2].begin(9600);
814   }
815 }
816
817 void mySerial111() {
818   if (mySerial1[2] == true) {
819     mySerial1[2] = false;
820     mySerial1[0].begin(9600);
821   }
822 }
823
824 void mySerial112() {
825   if (mySerial1[0] == true) {
826     mySerial1[0] = false;
827     mySerial1[1].begin(9600);
828   }
829 }
830
831 void mySerial113() {
832   if (mySerial1[1] == true) {
833     mySerial1[1] = false;
834     mySerial1[2].begin(9600);
835   }
836 }
837
838 void mySerial114() {
839   if (mySerial1[2] == true) {
840     mySerial1[2] = false;
841     mySerial1[0].begin(9600);
842   }
843 }
844
845 void mySerial115() {
846   if (mySerial1[0] == true) {
847     mySerial1[0] = false;
848     mySerial1[1].begin(9600);
849   }
850 }
851
852 void mySerial116() {
853   if (mySerial1[1] == true) {
854     mySerial1[1] = false;
855     mySerial1[2].begin(9600);
856   }
857 }
858
859 void mySerial117() {
860   if (mySerial1[2] == true) {
861     mySerial1[2] = false;
862     mySerial1[0].begin(9600);
863   }
864 }
865
866 void mySerial118() {
867   if (mySerial1[0] == true) {
868     mySerial1[0] = false;
869     mySerial1[1].begin(9600);
870   }
871 }
872
873 void mySerial119() {
874   if (mySerial1[1] == true) {
875     mySerial1[1] = false;
876     mySerial1[2].begin(9600);
877   }
878 }
879
880 void mySerial120() {
881   if (mySerial1[2] == true) {
882     mySerial1[2] = false;
883     mySerial1[0].begin(9600);
884   }
885 }
886
887 void mySerial121() {
888   if (mySerial1[0] == true) {
889     mySerial1[0] = false;
890     mySerial1[1].begin(9600);
891   }
892 }
893
894 void mySerial122() {
895   if (mySerial1[1] == true) {
896     mySerial1[1] = false;
897     mySerial1[2].begin(9600);
898   }
899 }
900
901 void mySerial123() {
902   if (mySerial1[2] == true) {
903     mySerial1[2] = false;
904     mySerial1[0].begin(9600);
905   }
906 }
907
908 void mySerial124() {
909   if (mySerial1[0] == true) {
910     mySerial1[0] = false;
911     mySerial1[1].begin(9600);
912   }
913 }
914
915 void mySerial125() {
916   if (mySerial1[1] == true) {
917     mySerial1[1] = false;
918     mySerial1[2].begin(9600);
919   }
920 }
921
922 void mySerial126() {
923   if (mySerial1[2] == true) {
924     mySerial1[2] = false;
925     mySerial1[0].begin(9600);
926   }
927 }
928
929 void mySerial127() {
930   if (mySerial1[0] == true) {
931     mySerial1[0] = false;
932     mySerial1[1].begin(9600);
933   }
934 }
935
936 void mySerial128() {
937   if (mySerial1[1] == true) {
938     mySerial1[1] = false;
939     mySerial1[2].begin(9600);
940   }
941 }
942
943 void mySerial129() {
944   if (mySerial1[2] == true) {
945     mySerial1[2] = false;
946     mySerial1[0].begin(9600);
947   }
948 }
949
950 void mySerial130() {
951   if (mySerial1[0] == true) {
952     mySerial1[0] = false;
953     mySerial1[1].begin(9600);
954   }
955 }
956
957 void mySerial131() {
958   if (mySerial1[1] == true) {
959     mySerial1[1] = false;
960     mySerial1[2].begin(9600);
961   }
962 }
963
964 void mySerial132() {
965   if (mySerial1[2] == true) {
966     mySerial1[2] = false;
967     mySerial1[0].begin(9600);
968   }
969 }
970
971 void mySerial133() {
972   if (mySerial1[0] == true) {
973     mySerial1[0] = false;
974     mySerial1[1].begin(9600);
975   }
976 }
977
978 void mySerial134() {
979   if (mySerial1[1] == true) {
980     mySerial1[1] = false;
981     mySerial1[2].begin(9600);
982   }
983 }
984
985 void mySerial135() {
986   if (mySerial1[2] == true) {
987     mySerial1[2] = false;
988     mySerial1[0].begin(9600);
989   }
990 }
991
992 void mySerial136() {
993   if (mySerial1[0] == true) {
994     mySerial1[0] = false;
995     mySerial1[1].begin(9600);
996   }
997 }
998
999 void mySerial137() {
1000   if (mySerial1[1] == true) {
1001     mySerial1[1] = false;
1002     mySerial1[2].begin(9600);
1003   }
1004 }
1005
1006 void mySerial138() {
1007   if (mySerial1[2] == true) {
1008     mySerial1[2] = false;
1009     mySerial1[0].begin(9600);
1010   }
1011 }
1012
1013 void mySerial139() {
1014   if (mySerial1[0] == true) {
1015     mySerial1[0] = false;
1016     mySerial1[1].begin(9600);
1017   }
1018 }
1019
1020 void mySerial140() {
1021   if (mySerial1[1] == true) {
1022     mySerial1[1] = false;
1023     mySerial1[2].begin(9600);
1024   }
1025 }
1026
1027 void mySerial141() {
1028   if (mySerial1[2] == true) {
1029     mySerial1[2] = false;
1030     mySerial1[0].begin(9600);
1031   }
1032 }
1033
1034 void mySerial142() {
1035   if (mySerial1[0] == true) {
1036     mySerial1[0] = false;
1037     mySerial1[1].begin(9600);
1038   }
1039 }
1040
1041 void mySerial143() {
1042   if (mySerial1[1] == true) {
1043     mySerial1[1] = false;
1044     mySerial1[2].begin(9600);
1045   }
1046 }
1047
1048 void mySerial144() {
1049   if (mySerial1[2] == true) {
1050     mySerial1[2] = false;
1051     mySerial1[0].begin(9600);
1052   }
1053 }
1054
1055 void mySerial145() {
1056   if (mySerial1[0] == true) {
1057     mySerial1[0] = false;
1058     mySerial1[1].begin(9600);
1059   }
1060 }
1061
1062 void mySerial146() {
1063   if (mySerial1[1] == true) {
1064     mySerial1[1] = false;
1065     mySerial1[2].begin(9600);
1066   }
1067 }
1068
1069 void mySerial147() {
1070   if (mySerial1[2] == true) {
1071     mySerial1[2] = false;
1072     mySerial1[0].begin(9600);
1073   }
1074 }
1075
1076 void mySerial148() {
1077   if (mySerial1[0] == true) {
1078     mySerial1[0] = false;
1079     mySerial1[1].begin(9600);
1080   }
1081 }
1082
1083 void mySerial149() {
1084   if (mySerial1[1] == true) {
1085     mySerial1[1] = false;
1086     mySerial1[2].begin(9600);
1087   }
1088 }
1089
1090 void mySerial150() {
1091   if (mySerial1[2] == true) {
1092     mySerial1[2] = false;
1093     mySerial1[0].begin(9600);
1094   }
1095 }
1096
1097 void mySerial151() {
1098   if (mySerial1[0] == true) {
1099     mySerial1[0] = false;
1100     mySerial1[1].begin(9600);
1101   }
1102 }
1103
1104 void mySerial152() {
1105   if (mySerial1[1] == true) {
1106     mySerial1[1] = false;
1107     mySerial1[2].begin(9600);
1108   }
1109 }
1110
1111 void mySerial153() {
1112   if (mySerial1[2] == true) {
1113     mySerial1[2] = false;
1114     mySerial1[0].begin(9600);
1115   }
1116 }
1117
1118 void mySerial154() {
1119   if (mySerial1[0] == true) {
1120     mySerial1[0] = false;
1121     mySerial1[1].begin(9600);
1122   }
1123 }
1124
1125 void mySerial155() {
1126   if (mySerial1[1] == true) {
1127     mySerial1[1] = false;
1128     mySerial1[2].begin(9600);
1129   }
1130 }
1131
1132 void mySerial156() {
1133   if (mySerial1[2] == true) {
1134     mySerial1[2] = false;
1135     mySerial1[0].begin(9600);
1136   }
1137 }
1138
1139 void mySerial157() {
1140   if (mySerial1[0] == true) {
1141     mySerial1[0] = false;
1142     mySerial1[1].begin(9600);
1143   }
1144 }
1145
1146 void mySerial158() {
1147   if (mySerial1[1] == true) {
1148     mySerial1[1] = false;
1149     mySerial1[2].begin(9600);
1150   }
1151 }
1152
1153 void mySerial159() {
1154   if (mySerial1[2] == true) {
1155     mySerial1[2] = false;
1156     mySerial1[0].begin(9600);
1157   }
1158 }
1159
1160 void mySerial160() {
1161   if (mySerial1[0] == true) {
1162     mySerial1[0] = false;
1163     mySerial1[1].begin(9600);
1164   }
1165 }
1166
1167 void mySerial161() {
1168   if (mySerial1[1] == true) {
1169     mySerial1[1] = false;
1170     mySerial1[2].begin(9600);
1171   }
1172 }
1173
1174 void mySerial162() {
1175   if (mySerial1[2] == true) {
1176     mySerial1[2] = false;
1177     mySerial1[0].begin(9600);
1178   }
1179 }
1180
1181 void mySerial163() {
1182   if (mySerial1[0] == true) {
1183     mySerial1[0] = false;
1184     mySerial1[1].begin(9600);
1185   }
1186 }
1187
1188 void mySerial164() {
1189   if (mySerial1[1] == true) {
1190     mySerial1[1] = false;
1191     mySerial1[2].begin(9600);
1192   }
1193 }
1194
1195 void mySerial165() {
1196   if (mySerial1[2] == true) {
1197     mySerial1[2] = false;
1198     mySerial1[0].begin(9600);
1199   }
1200 }
1201
1202 void mySerial166() {
1203   if (mySerial1[0] == true) {
1204     mySerial1[0] = false;
1205     mySerial1[1].begin(9600);
1206   }
1207 }
1208
1209 void mySerial167() {
1210   if (mySerial1[1] == true) {
1211     mySerial1[1] = false;
1212     mySerial1[2].begin(9600);
1213   }
1214 }
1215
1216 void mySerial168() {
1217   if (mySerial1[2] == true) {
1218     mySerial1[2] = false;
1219     mySerial1[0].begin(9600);
1220   }
1221 }
1222
1223 void mySerial169() {
1224   if (mySerial1[0] == true) {
1225     mySerial1[0] = false;
1226     mySerial1[1].begin(9600);
1227   }
1228 }
1229
1230 void mySerial170() {
1231   if (mySerial1[1] == true)
```

*Receiver code is the same for the 3 different Arduinos, only pin number changes.

Data



Optical Communications

By: Jerry, Jeremy, Timothy, Alicia

Goal

Test the efficiency of a free-space Wavelength Division Multiplexing system and the impact the length of a message has on the accuracy and speed of transmission.

Background

Optical communication serves as the basis for quick transmissions of data like texting. Currently, almost all optical communications involve Wavelength Division Multiplexing (WDM) which allows large amounts of data to be sent through a single optical beam by splitting and combining different wavelengths of light.

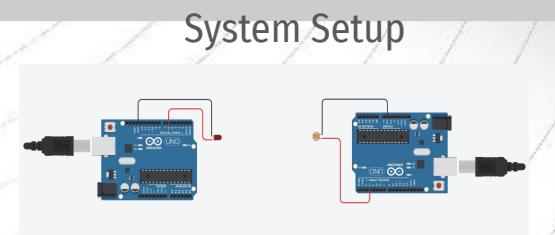


Diagram of one laser test system.
Left is the laser connected to Arduino.
Right is a photoresistor connected to
arduino.

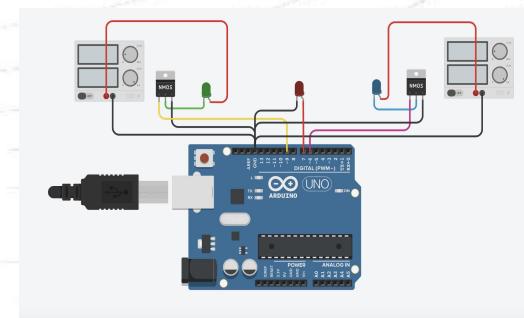
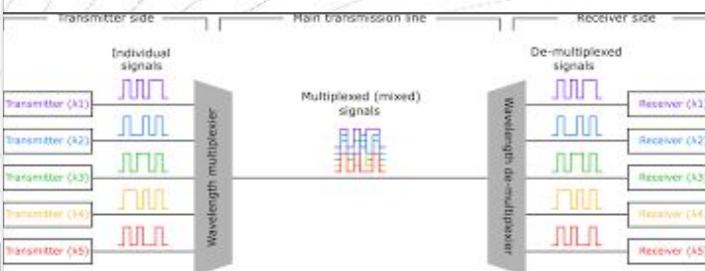
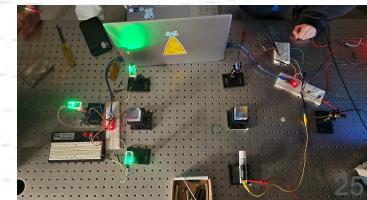


Diagram 3 laser transmission system.
Receiver system is same as before.



Picture of final setup: laser diode on right, photoresistors on left.



Code:



[jerry-zh0u/COSMOSCommProject](https://github.com/jerry-zh0u/COSMOSCommProject)
(github.com)

References

Huang, X.-H. (2018, November 16). *WDM Free-Space Optical Communication System of High-Speed Hybrid Signals*. IEEExplore. Retrieved August 1, 2024, from <https://ieeexplore.ieee.org/document/8537937>

Suh, S. (1987, January). *Pulse width modulation for analog fiber-optic communications*. IEEExplore. Retrieved August 1, 2024, from <https://ieeexplore.ieee.org/document/1075401>

Yeh, C.-H. (n.d.). *Bidirectional free space optical communication (FSO) in WDM access network with 1000-m supportable free space link*. Science Direct. Retrieved August 1, 2024, from
https://www.sciencedirect.com/science/article/abs/pii/S0030401818310265?casa_token=p9gwnOFNDmQAAAAA:Wx2jiIHhctB5nAwSfjvm6t9LDyf62GOswijvogUP65v7vGxRX09MDox5zgDw6CydpCPeB3j1EA