

DYNAMIC OPTICAL TRACKING IN LI-FI SYSTEMS

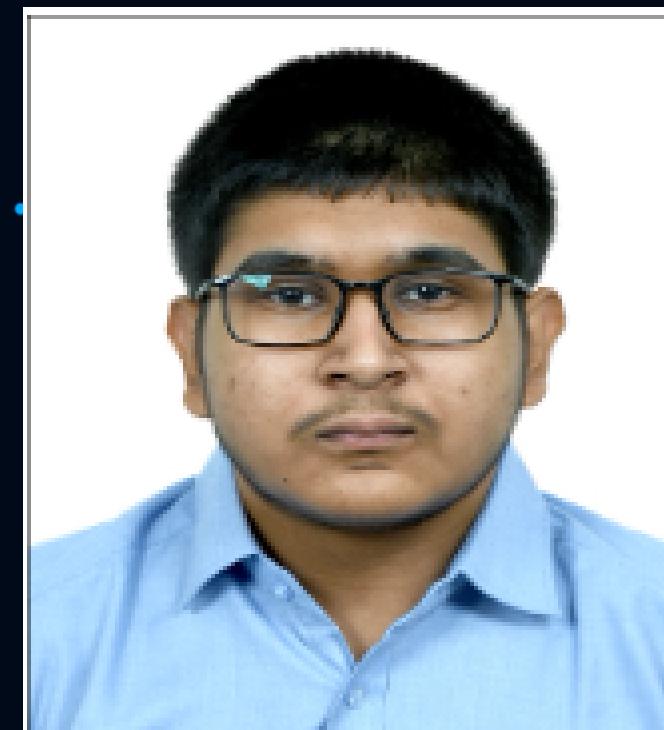
Li-Fi (Light Fidelity) is an *optical wireless communication* technology that uses *visible light* instead of radio waves for *data transmission*. It offers *ultra-fast speeds* (up to 100 Gbps), enhanced *security* since light cannot penetrate walls, *EMI-free operation* suitable for hospitals and aircraft, and a *massive available spectrum*. This project focuses on *dynamic optical tracking*, where the *transmitter automatically follows the receiver's movement* to maintain *stable, high-speed communication* and prevent *signal dropouts*.

The main objective is to design a *smart, Arduino-based Li-Fi prototype* that *auto-adjusts* the LED transmitter's direction according to the receiver's position, ensuring *uninterrupted optical data transfer* in a *simple and affordable setup*. The system uses an *Arduino Uno*, *four LDR sensors*, an *LED transmitter*, a *servo motor* for tracking, and basic *resistors and wiring*. The communication process involves four stages — ***Transmission*** (LED signal), ***Reception*** (LDR detection), ***Tracking*** (servo alignment), and ***Decoding*** (Arduino processing). The software continuously *reads light intensities, compares values*, and *adjusts servo angles* for smooth and responsive alignment.

As the receiver moves, *variations in light intensity* are detected by the LDRs, prompting the servo to *reposition the LED* for optimal illumination and *stable data flow*. This system has applications in *high-speed indoor communication, autonomous robots and drones, smart lighting, underwater data transmission, interactive displays*, and *secure point-to-point communication*. References include *Wikipedia*, *GeeksforGeeks*, *pureLiFi*, *tech@gov*, *TinkerCad*, and *OpenAI/Gemini* for design, simulation, and visualization support.

DYNAMIC OPTICAL TRACKING IN LI-FI SYSTEMS

PRESENTED BY : SOUREEN MAJUMDER
ROLL NO : 002411102058



CONTENTS



- INTRODUCTION
- OBJECTIVE
- COMPONENTS
- CIRCUIT DIAGRAM
- CODE
- SIMULATION ON
TINKERCAD
- WORKING
- APPLICATIONS
- REFERENCES

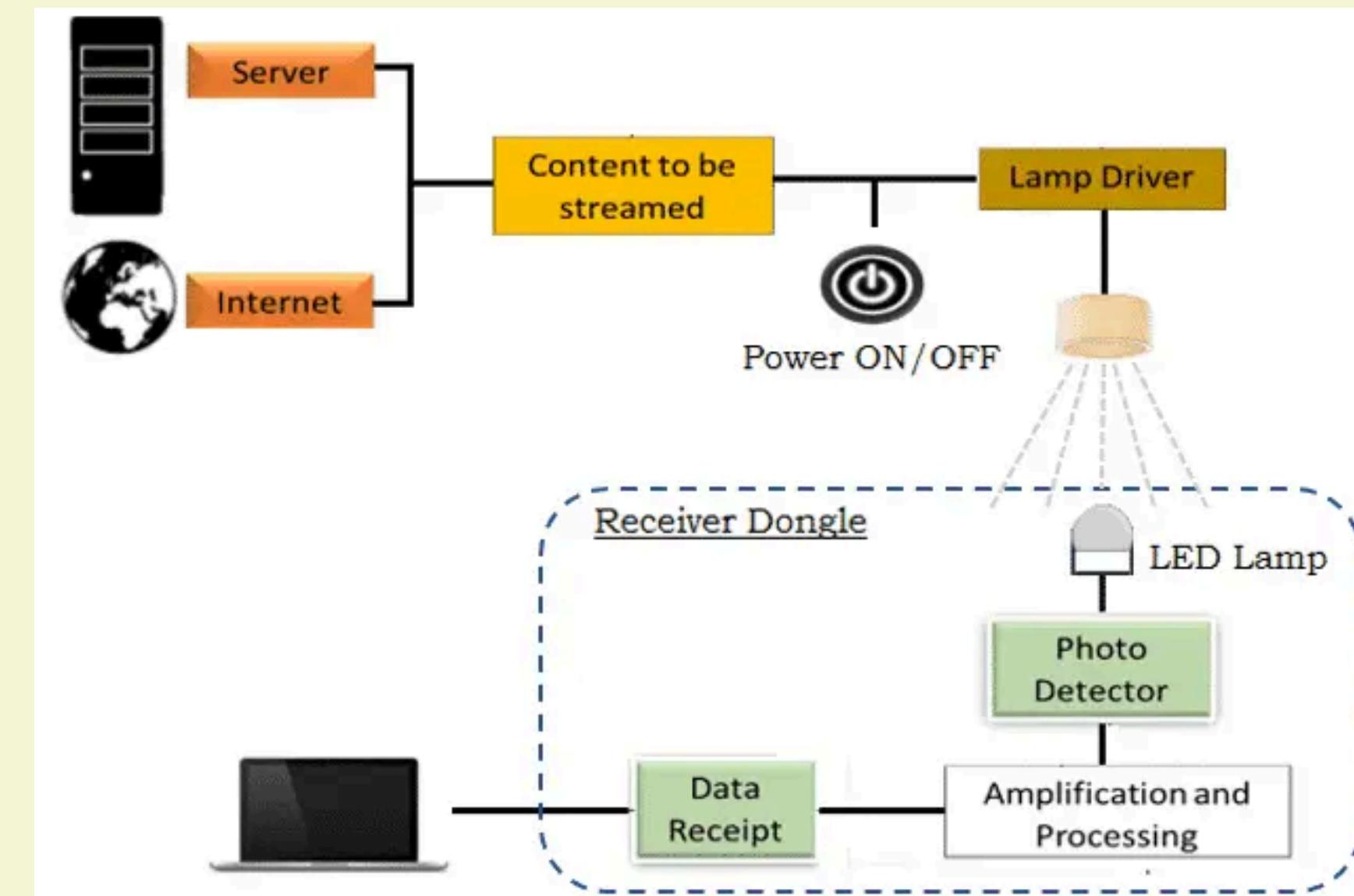
INTRODUCTION

- **WHY LI-FI ?**

1. Ultra-Fast Speeds (Up to 100 Gbps)
2. Highly Secure (Can't penetrate walls)
3. EMI-Free (Safe in hospitals, aircrafts)
4. Massive Spectrum (10,000x RF)

- **DYNAMIC TRACKING**

1. Follows receiver movement
2. Reduces signal dropouts
3. Allows uninterrupted, high-speed communication, even if you're walking or moving.



OBJECTIVE

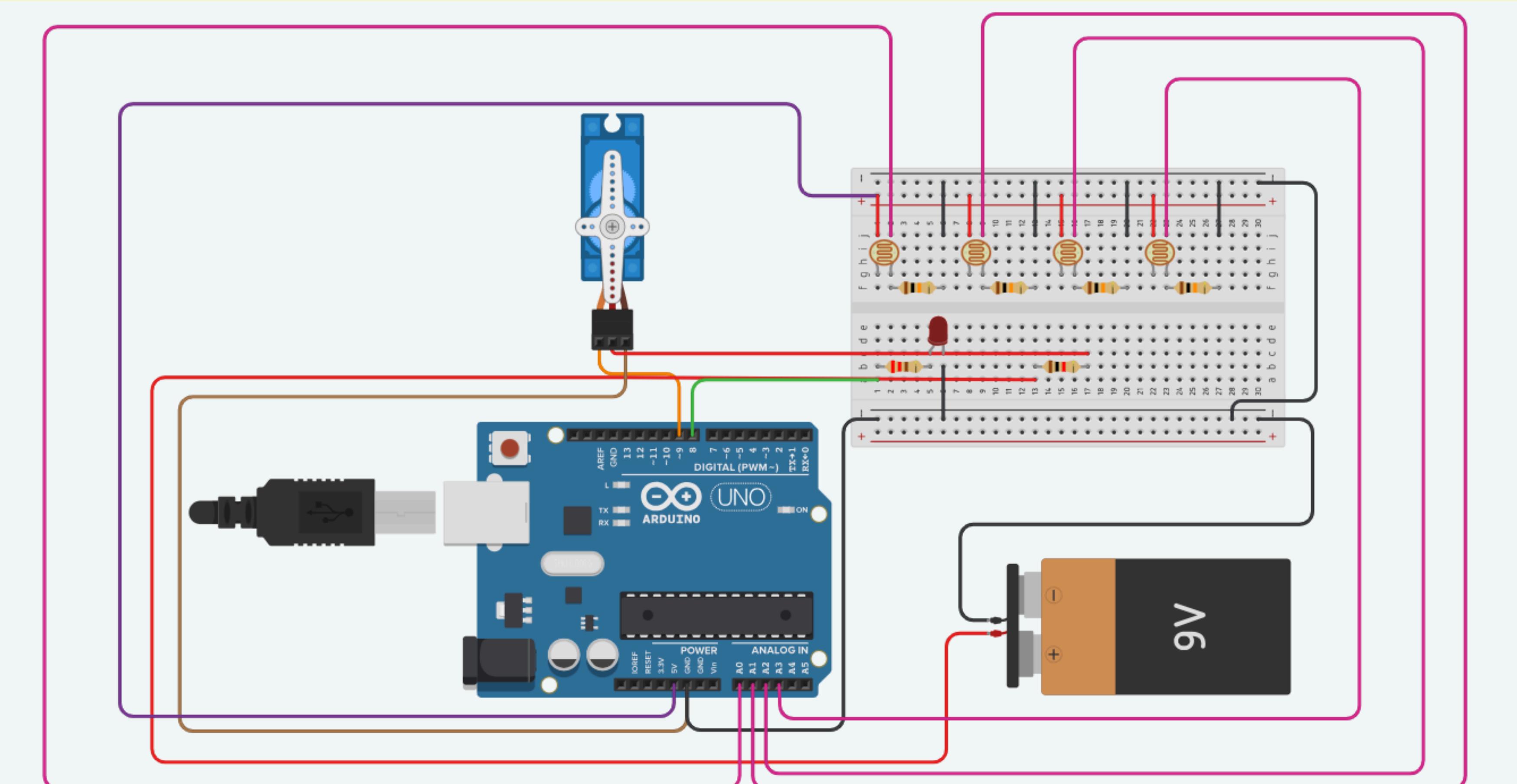
- Create a smart Li-Fi system prototype that auto-adjusts based on receiver movement.
- Maintain continuous light-based communication without manual alignment.
- Negates the limitations of static Li-Fi systems.
- Simple, affordable, Arduino-based setup.

COMPONENTS

1. Arduino Uno R3 (Microcontroller)
2. $4 \times$ LDRs (Light Dependent Resistors) – LDR1, LDR2, LDR3, LDR4
3. $4 \times 10\text{ k}\Omega$ resistors (for LDR voltage divider)
4. $1 \times$ LED (for Li-Fi transmitter)
5. $1 \times 220\text{ }\Omega$ resistor (for LED current limiting)
6. $1 \times$ Servo motor (for movement tracking)
7. Breadboard
8. 9V DC power supply (for servo motor)
9. $1 \times 1\text{ k}\Omega$ resistor (for servo motor current limiting)
10. Jumper Wires

CIRCUIT DIAGRAM

6

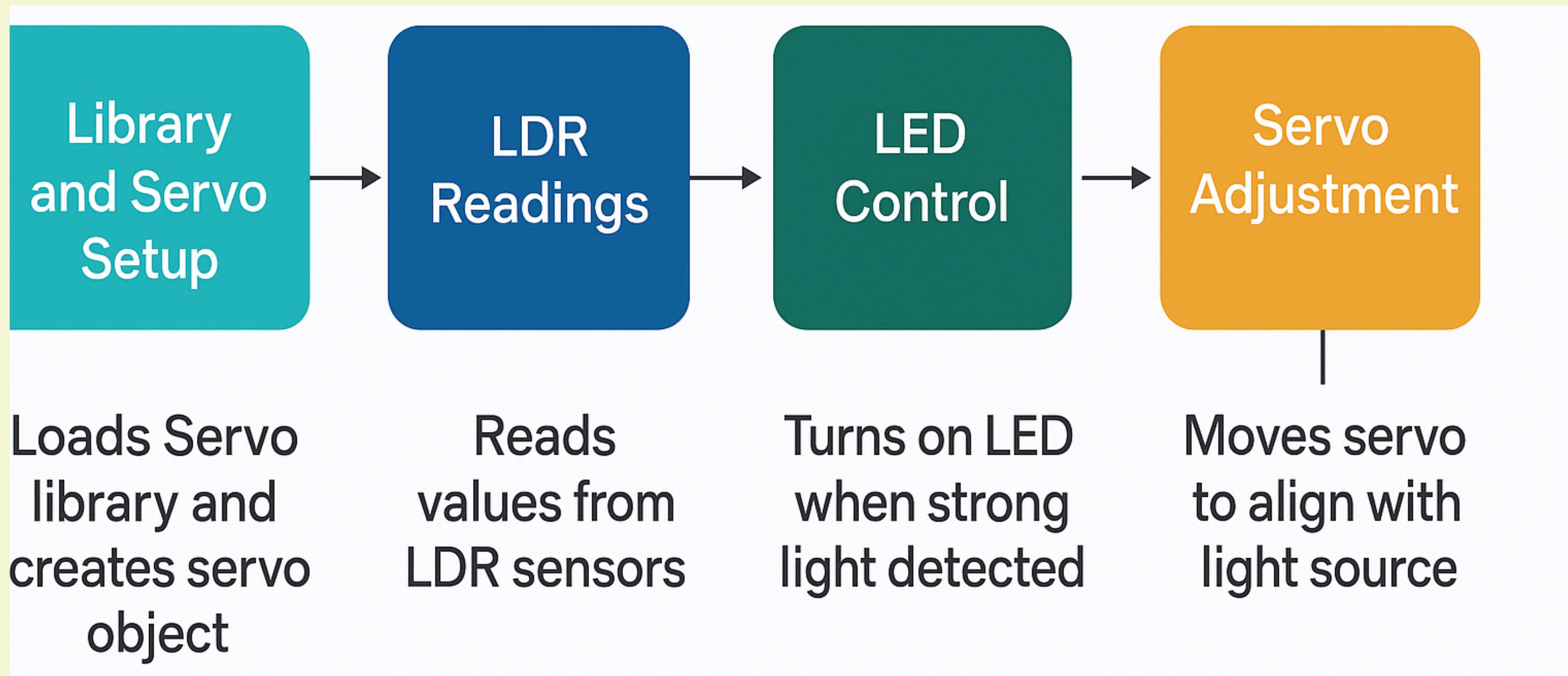


TRANSFER OF INFORMATION

FOUR Major Processes

1. TRANSMISSION (LED : LIGHT SIGNAL)
2. RECEPTION (LDR : ELECTRICAL SIGNAL)
3. TRACKING (SERVO : ALIGNMENT)
4. DECODING (ARDUINO : USABLE DATA)

REPRESENTATION OF CODE AS A FLOWCHART



CODE BLOCK I

- LIBRARY AND SERVO SETUP
- PIN ASSIGNMENT OF LEDs
- INITIALISATION OF SERVO POSITION (90 degrees)
- SETTING OF LED AS OUTPUT
- INITIALISATION OF LDRs IN ANALOG PWM PINS

```
#include <Servo.h>

Servo trackerServo;

// LDR analog pins
const int LDR1 = A0;
const int LDR2 = A1;
const int LDR3 = A2;
const int LDR4 = A3;

const int LED_PIN = 8;
int servoPos = 90; // Start at middle

void setup() {
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
  trackerServo.attach(9);
  trackerServo.write(servoPos);
}
```

CODE BLOCK II

- ANALOG READ OF INTENSITY OF LIGHT
- SENSOR READINGS ON SERIAL MONITOR
- COMPARING OF INTENSITY OF LDRs
- CONVERSION FROM CANDELA TO NUMERICAL VALUE (0-123).

```
void loop() {
    // Read LDR values
    int val1 = analogRead(LDR1);
    int val2 = analogRead(LDR2);
    int val3 = analogRead(LDR3);
    int val4 = analogRead(LDR4);

    // Debug output
    Serial.print("LDR1: "); Serial.print(val1);
    Serial.print(" | LDR2: "); Serial.print(val2);
    Serial.print(" | LDR3: "); Serial.print(val3);
    Serial.print(" | LDR4: "); Serial.println(val4);

    // Average top and bottom readings
    int topAvg = (val1 + val2) / 2;
    int bottomAvg = (val3 + val4) / 2;

    // Average left and right readings
    int leftAvg = (val1 + val3) / 2;
    int rightAvg = (val2 + val4) / 2;
```

CODE BLOCK III

- INDICATION OF STRENGTH OF INTENSITY
- CONFIRMS THE OPTICAL SIGNAL
- ADJUSTMENT OF SERVO ANGLE BASED ON BRIGHTNESS
- OPERATION WITHIN SAFE LIMITING ANGLE
- SMALL DELAY (reduces jitter, smoothness)

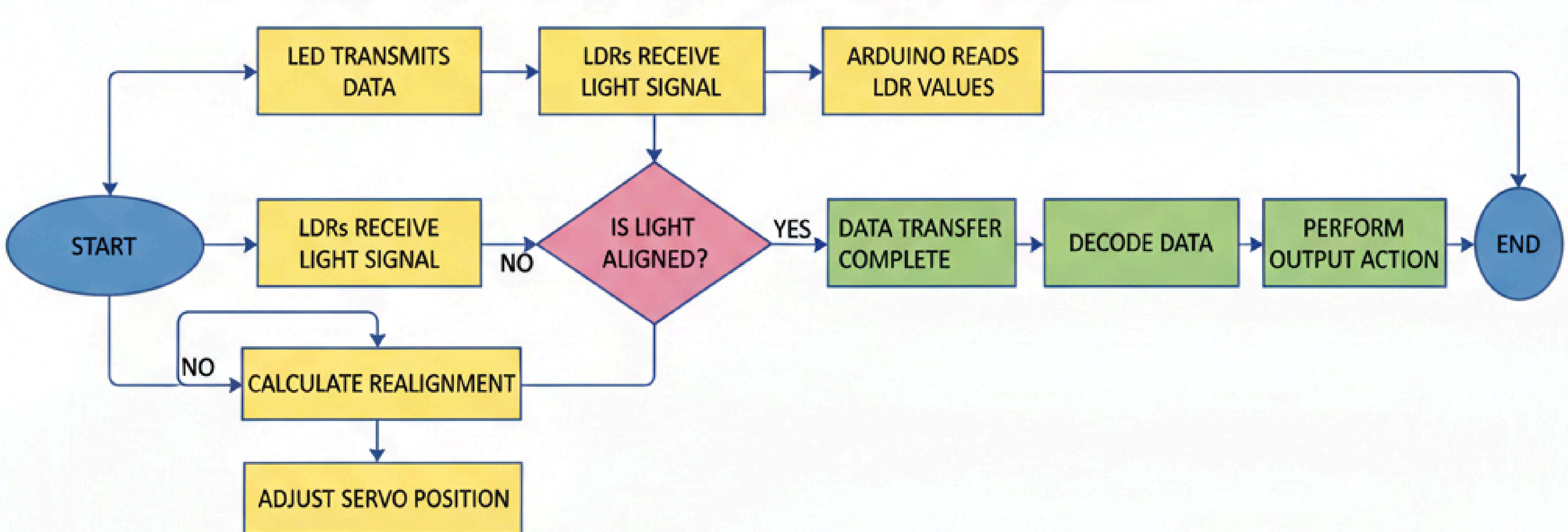
```
// LED ON if strong light detected
if (topAvg > 500 || bottomAvg > 500) {
    digitalWrite(LED_PIN, HIGH);
} else {
    digitalWrite(LED_PIN, LOW);
}

// Adjust servo horizontally
if (leftAvg > rightAvg + 50) {
    servoPos += 1; // Turn left
} else if (rightAvg > leftAvg + 50) {
    servoPos -= 1; // Turn right
}

// Constrain servo position
servoPos = constrain(servoPos, 0, 180);
trackerServo.write(servoPos);

delay(50);
```

WORKING MECHANISM



SIMULATION I

Photoresistor

Name LDR4 (RECIEVER 4)

```

28 // Debug output
29 Serial.print("LDR1: "); Serial.print(val1);
30 Serial.print(" | LDR2: "); Serial.print(val2);
31 Serial.print(" | LDR3: "); Serial.print(val3);
32 Serial.print(" | LDR4: "); Serial.println(val4);
33
34 // Average top and bottom readings
35 int topAvg = (val1 + val2) / 2;
36 int bottomAvg = (val3 + val4) / 2;
37
38 // Average left and right readings
39 int leftAvg = (val1 + val3) / 2;
40 int rightAvg = (val2 + val4) / 2;
41
42 // LED ON if strong light detected
43 if (topAvg > 500 || bottomAvg > 500) {
44   digitalWrite(LED_PIN, HIGH);
45 } else {
46   digitalWrite(LED_PIN, LOW);
47 }
48
49 // Adjust servo horizontally
50 if (leftAvg > rightAvg + 50) {
51   servoPos += 1; // Turn left
52 } else if (rightAvg > leftAvg + 50) {

```

Serial Monitor

LDR1: 54 | LDR2: 54 | LDR3: 54 | LDR4: 54
LDR1: 54 | LDR2: 54 | LDR3: 54 | LDR4: 54
LDR1: 54 | LDR2: 54 | LDR3: 54 | LDR4: 54
LDR1: 54 | LDR2: 54 | LDR3: 54 | LDR4: 54
LDR1: 54 | LDR2: 54 | LDR3: 54 | LDR4: 54

SIMULATION II

The screenshot shows a simulation setup for an Arduino Uno R3. The circuit consists of an Arduino Uno connected to a breadboard. On the breadboard, four LDRs (Photoresistors) are arranged in a cross pattern. The top-left LDR is labeled "LDR4 (RECIEVER 4)". A red wire connects the output of this LDR to digital pin 8 of the Arduino. The Arduino is also connected to a servo motor, which is mounted on a black base. The servo's control pin is connected to digital pin 9. A red LED is connected to digital pin 13 via a current-limiting resistor. The Arduino is powered by a USB cable, and its pins are labeled with their functions: AREF, GND, 1.3, 1.2, 1.1, 1.0, 9, AREF, GND, 3.3V, POWER, ANALOG IN, A0, A1, A2, A3, A4, A5, IOREF, RESET, 3.3V, GND, Vin, and TX, RX.

The code in the editor is as follows:

```

28 // Debug output
29 Serial.print("LDR1: "); Serial.print(val1);
30 Serial.print(" | LDR2: "); Serial.print(val2);
31 Serial.print(" | LDR3: "); Serial.print(val3);
32 Serial.print(" | LDR4: "); Serial.println(val4);
33
34 // Average top and bottom readings
35 int topAvg = (val1 + val2) / 2;
36 int bottomAvg = (val3 + val4) / 2;
37
38 // Average left and right readings
39 int leftAvg = (val1 + val3) / 2;
40 int rightAvg = (val2 + val4) / 2;
41
42 // LED ON if strong light detected
43 if (topAvg > 500 || bottomAvg > 500) {
44     digitalWrite(LED_PIN, HIGH);
45 } else {
46     digitalWrite(LED_PIN, LOW);
47 }
48
49 // Adjust servo horizontally
50 if (leftAvg > rightAvg + 50) {
51     servoPos += 1; // Turn left
52 } else if (rightAvg > leftAvg + 50) {
53     servoPos -= 1; // Turn right
54 }
```

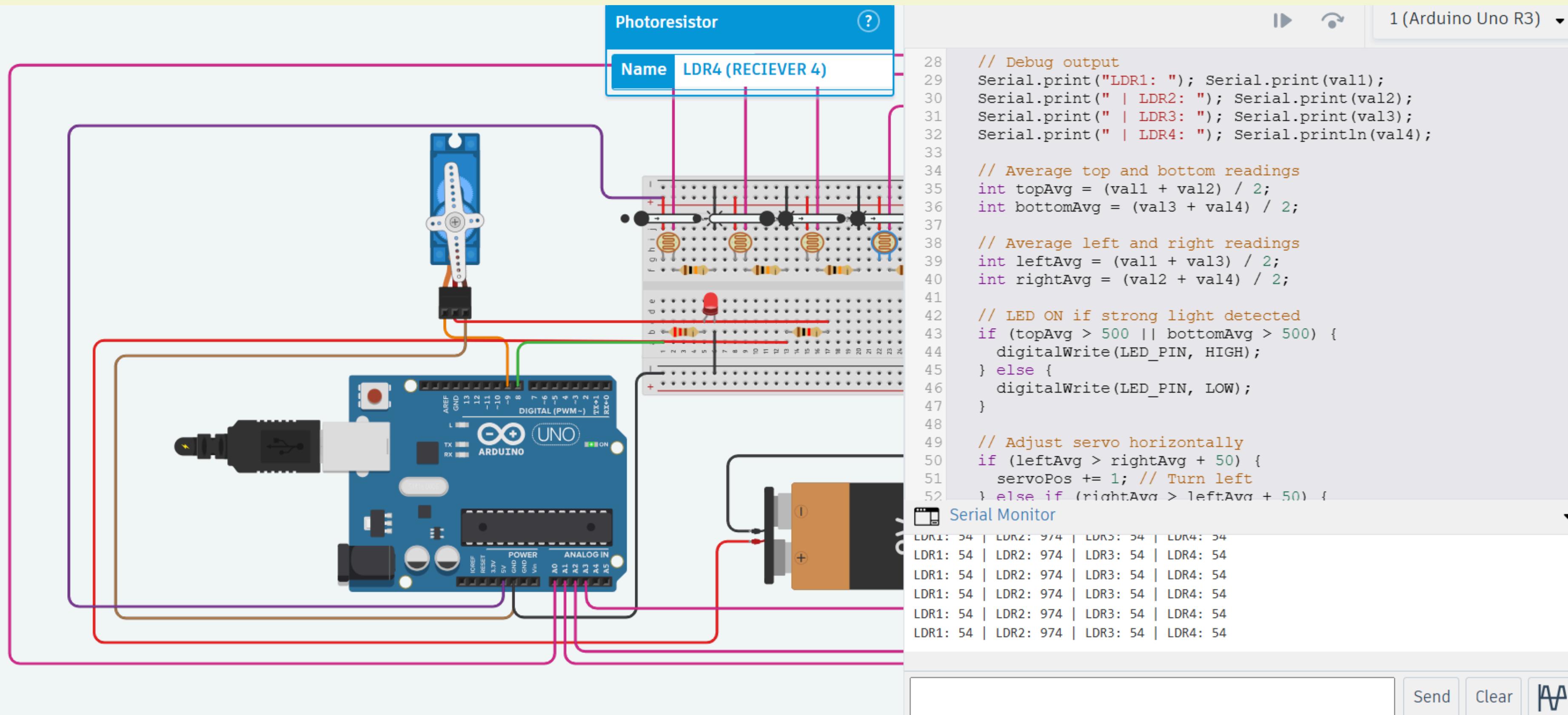
The Serial Monitor window shows the following data:

```

LDR1: 974 | LDR2: 54 | LDR3: 54 | LDR4: 54
LDR1: 974 | LDR2: 54 | LDR3: 54 | LDR4: 54
LDR1: 974 | LDR2: 54 | LDR3: 54 | LDR4: 54
LDR1: 974 | LDR2: 54 | LDR3: 54 | LDR4: 54
LDR1: 974 | LDR2: 54 | LDR3: 54 | LDR4: 54
LDR1: 974 | LDR2: 54 | LDR3: 54 | LDR4: 54
```

Buttons at the bottom right include "Send", "Clear", and a logo.

SIMULATION III



SIMULATION IV

The screenshot shows a simulation interface for an Arduino Uno R3. On the left, a breadboard circuit is connected to an Arduino Uno. The circuit includes four photoresistors (LDRs) labeled LDR1 through LDR4, which are positioned above a horizontal track. A red LED is also connected to the breadboard. On the right, the Arduino IDE displays the following code:

```

// Debug output
Serial.print("LDR1: "); Serial.print(val1);
Serial.print(" | LDR2: "); Serial.print(val2);
Serial.print(" | LDR3: "); Serial.print(val3);
Serial.print(" | LDR4: "); Serial.println(val4);

// Average top and bottom readings
int topAvg = (val1 + val2) / 2;
int bottomAvg = (val3 + val4) / 2;

// Average left and right readings
int leftAvg = (val1 + val3) / 2;
int rightAvg = (val2 + val4) / 2;

// LED ON if strong light detected
if (topAvg > 500 || bottomAvg > 500) {
    digitalWrite(LED_PIN, HIGH);
} else {
    digitalWrite(LED_PIN, LOW);
}

// Adjust servo horizontally
if (leftAvg > rightAvg + 50) {
    servoPos += 1; // Turn left
} else if (rightAvg > leftAvg + 50) {
}

```

The serial monitor window at the bottom shows the following data:

```

LDR1: 54 | LDR2: 54 | LDR3: 974 | LDR4: 54
LDR1: 54 | LDR2: 54 | LDR3: 974 | LDR4: 54
LDR1: 54 | LDR2: 54 | LDR3: 974 | LDR4: 54
LDR1: 54 | LDR2: 54 | LDR3: 974 | LDR4: 54
LDR1: 54 | LDR2: 54 | LDR3: 974 | LDR4: 54
LDR1: 54

```

At the bottom right, there are "Send" and "Clear" buttons.

SIMULATION V

Breadboard Small

Name 1

```

28 // Debug output
29 Serial.print("LDR1: "); Serial.print(val1);
30 Serial.print(" | LDR2: "); Serial.print(val2);
31 Serial.print(" | LDR3: "); Serial.print(val3);
32 Serial.print(" | LDR4: "); Serial.println(val4);
33
34 // Average top and bottom readings
35 int topAvg = (val1 + val2) / 2;
36 int bottomAvg = (val3 + val4) / 2;
37
38 // Average left and right readings
39 int leftAvg = (val1 + val3) / 2;
40 int rightAvg = (val2 + val4) / 2;
41
42 // LED ON if strong light detected
43 if (topAvg > 500 || bottomAvg > 500) {
44     digitalWrite(LED_PIN, HIGH);
45 } else {
46     digitalWrite(LED_PIN, LOW);
47 }
48
49 // Adjust servo horizontally
50 if (leftAvg > rightAvg + 50) {
51     servoPos += 1; // Turn left
52 } else if (rightAvg > leftAvg + 50) {
53     servoPos -= 1; // Turn right
54 }
55
56 Serial Monitor
57
58 LDR1: 54 | LDR2: 54 | LDR3: 937 | LDR4: 929
59 LDR1: 54 | LDR2: 54 | LDR3: 937 | LDR4: 929
60 LDR1: 54 | LDR2: 54 | LDR3: 937 | LDR4: 929
61 LDR1: 54 | LDR2: 54 | LDR3: 937 | LDR4: 929
62 LDR1: 54 | LDR2: 54 | LDR3: 937 | LDR4: 929
63 LDR1: 54 | LDR2: 54 | LDR3: 937 | LDR4: 929
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
648
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
748
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
848
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
888
889
889
890
891
892
893
894
895
896
897
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
928
928
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
968
969
970
971
972
973
974
975
976
977
978
978
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999

```

Send Clear

APPLICATIONS I

1. HIGH SPEED INDOOR COMMUNICATION

- Fast wireless internet using LED
- Keeps the receiver aligned

2. AUTONOMOUS ROBOTS & DRONES

- Navigate by tracking light.
- Useful in warehouses, manufacturing plants, search-and-rescue operations.

3. SMART LIGHTING SYSTEM

- Adjust direction, brightness.
- Saves energy, improves comfort.

APPLICATIONS II

1. UNDERWATER DATA TRANSMISSION

- Keep divers or underwater vehicles aligned with transmitters.

2. INTERACTIVE DISPLAYS & EXHIBITS

- Museums or exhibitions.
- Location-based information

3. SECURE POINT-TO-POINT COMMUNICATION

- Ensures strong links even with slight movements.

REFERENCES

- Wikipedia
- YouTube (Diversify Club Channel)
- GeeksforGeeks <https://www.geeksforgeeks.org/computer-networks/introduction-of-light-fidelity-li-fi/>
- pureLiFi <https://www.purelifi.com/about-lifi/>
- tech@gov
<https://tec.gov.in/pdf/Studypaper/lifi%20study%20paper%20-%20approved.pdf>
- OpenAI, Gemini (images, flowchart designs)
- TinkerCad (running the simulations and designing the circuit)

THANK YOU