



BÁO CÁO THỰC HÀNH

CÔNG NGHỆ INTERNET OF THINGS HIỆN ĐẠI

Lab 1

TÊN BÀI THỰC HÀNH

GVHD: **Phan Trung Phát**

Lớp: **NT532.021**

Họ và tên	MSSV
Nguyễn Thành Đăng	21520683
Nguyễn Trần Bảo Quốc	21520421

ĐÁNH GIÁ KHÁC (*):

Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình (1)	1,5 tuần
Link Video thực hiện (2) (nếu có)	Bài 1 Bài 2 Bài 3 Bài 4 Bài 5 Link drive
Ý kiến (3) (nếu có) + Khó khăn + Đề xuất ...	
Điểm tự đánh giá (4)	10/10

(*): phần (1) và (4) bắt buộc thực hiện.

Phần bên dưới là báo cáo chi tiết của nhóm/cá nhân thực hiện.



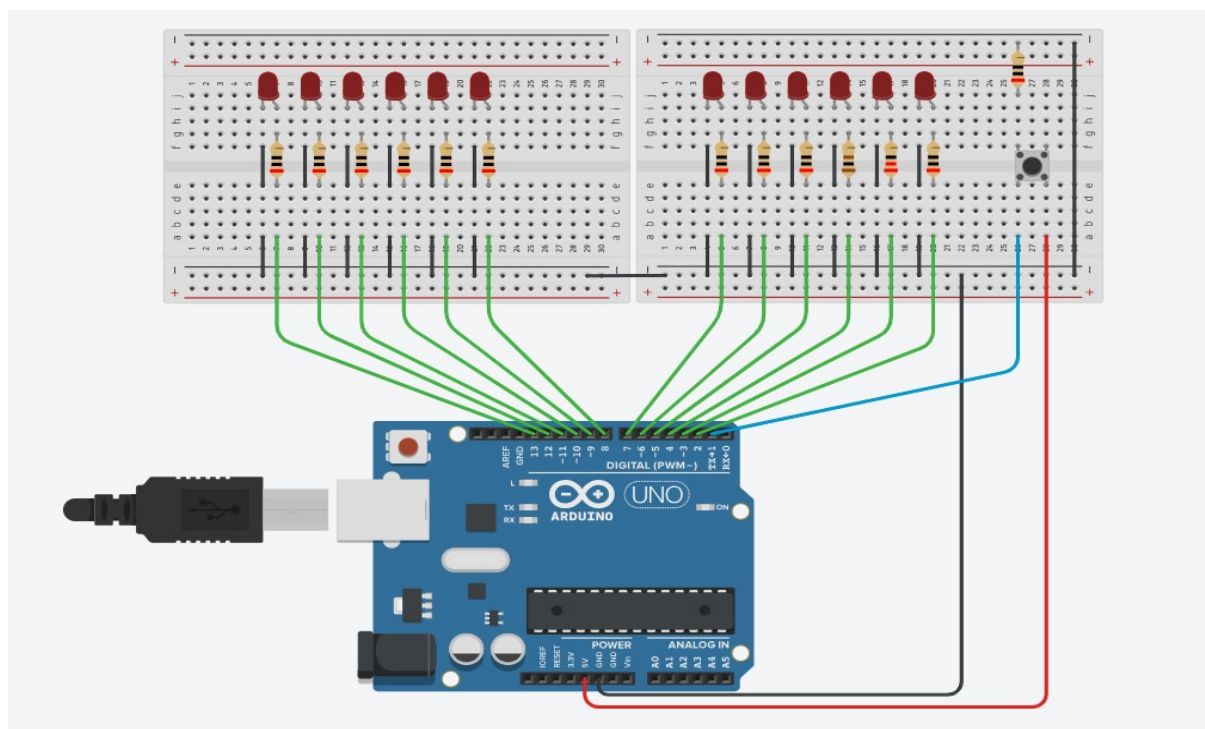
LƯU HÀNH NỘI BỘ



Câu hỏi 1. Xây dựng kịch bản gồm có 12 đèn LED và 1 nút bấm. Tùy số lượng lần bấm nút thì số lượng đèn sẽ sáng lên tương ứng. Sau khi đạt ngưỡng tối đa thì quay lại với giá trị 1 đèn sáng.

1. Minh chứng:

Link demo bài 1: [Tại đây](#)



Hình 1. Mô hình nối dây bài 1

```
1  int count = 0;
2  void setup() {
3      pinMode(1, INPUT);
4      for(int i = 2; i <= 13; i++)
5      {
6          pinMode(i, OUTPUT);
7      }
8  }
9  void loop() {
10     if (digitalRead(1) == HIGH) {
11         count ++;
12     }
13     delay(160);
14     if (count > 12)
15     {
16         count -= 12;
17         for (int i = 1; i < 12; i++)
18         {
19             digitalWrite(i+2, LOW);
20         }
21     }
22     else
23     {
24         for(int i = 0; i < count; i++)
25         {
26             digitalWrite(i+2, HIGH);
27         }
28     }
29 }
```

Hình 2. Code thực thi



2. Giải thích:

Khai báo biến `count = 0` với nhiệm vụ đếm số lần nhấn nút.

Trong hàm **setup()** cấu hình **MODE** cho chân 1 (chân button) là **INPUT** để nhận tín hiệu từ button và chân 2 đến chân 13 (chân dương led) là **OUTPUT** để đưa tín hiệu ra ngoài (line 3-8).

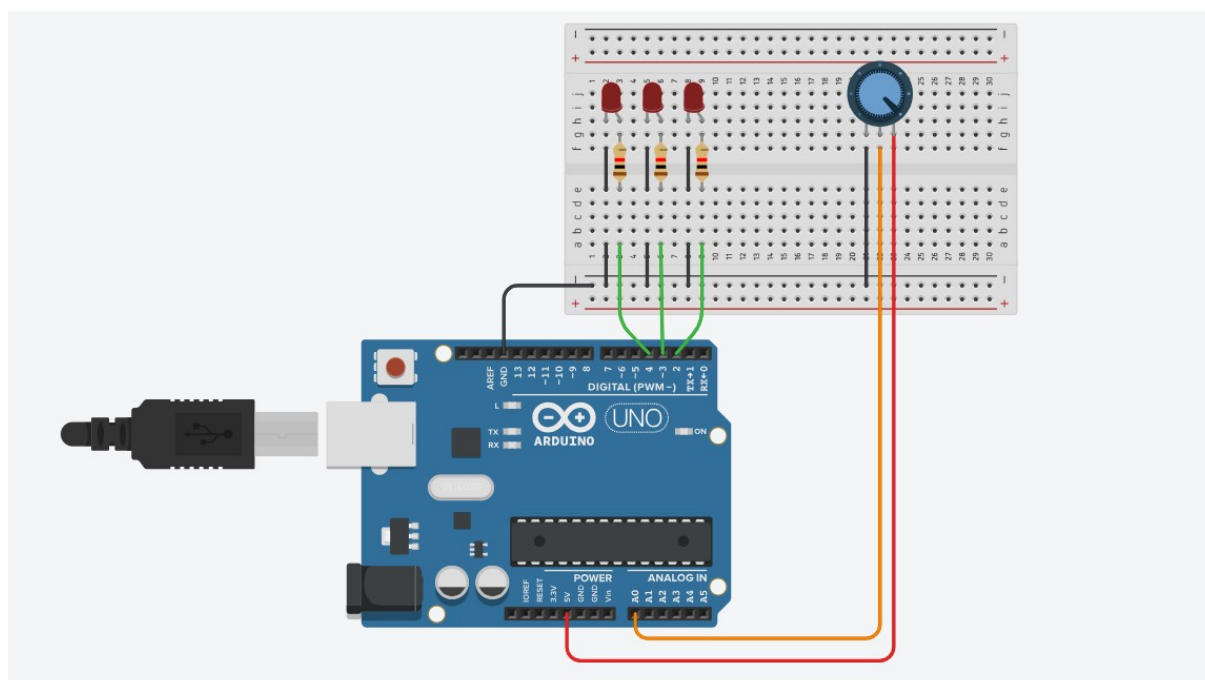
Trong hàm **loop()** thực hiện đếm số lần nhấn nút (line 10 -> line 12). Trong trường hợp `count > 12` (vượt quá số led) thì `count` được đặt lại là 1 (line 16) đồng thời thực hiện việc tắt led số 2 -> led số 12 (line 17 -> 20). Nếu trường hợp `count <= 12` thì bật led số 1 đến led số `count` (line 22 -> line 26).

Câu hỏi 2. Xây dựng kịch bản gồm có 3 đèn LED và 1 biến trở. Điều chỉnh tốc độ đèn lần lượt sáng dựa vào giá trị của biến trở được điều khiển. Giá trị của biến trở chia thành 3 mức: chậm - trung bình - nhanh. Các kết quả này được thể hiện tại Serial.

Lưu ý: đèn được sáng theo chiều từ trái sang phải và ngược lại.

1. Minh chứng:

Link demo bài 2: [Tại đây](#)



Hình 3. Mô hình nối dây bài 2

```

1 void setup()
2 {
3     for(int i = 2; i <= 4; i++)
4     {
5         pinMode(i,OUTPUT);
6     }
7     pinMode(A0, INPUT);
8     Serial.begin(9600);
9 }
10 void flashingLights(int delayTime) {
11     for(int i = 4; i >= 2; i--)
12     {
13         digitalWrite(i, HIGH);
14         delay(delayTime);
15     }
16     for(int i = 4; i >= 2; i--)
17     {
18         digitalWrite(i, LOW);
19         delay(delayTime);
20     }
21
22     for(int i = 2; i <= 4; i++)
23     {
24         digitalWrite(i, HIGH);
25         delay(delayTime);
26     }
27     for(int i = 2; i <= 4; i++)
28     {
29         digitalWrite(i, LOW);
30         delay(delayTime);
31     }
32 }
33 void loop()
34 {
35     int value = analogRead(A0);
36     if (value == 0) {
37         for(int i = 4; i >= 2; i--)
38         {
39             digitalWrite(i, LOW);
40             delay(100);
41         }
42     }
43     else if (value > 0 && value <= 341) {
44         Serial.println("SLOW");
45         flashingLights(1500);
46     }
47     else if (value > 341 && value < 682) {
48         Serial.println("MEDIUM");
49         flashingLights(1000);
50     }
51     else {
52         Serial.println("FAST");
53         flashingLights(200);
54     }
55     delay(100);
56 }

```

Hình 4. Code thực thi

2. Giải thích:

Trong hàm **setup()** cấu hình **MODE** cho chân A0 (chân button) là **INPUT** để nhận tín hiệu từ button và chân 2 đến chân 4 (chân



đương led) là **OUTPUT** để đưa tín hiệu ra ngoài, khởi tạo thông số kết nối của Monitor. (line 3-8)

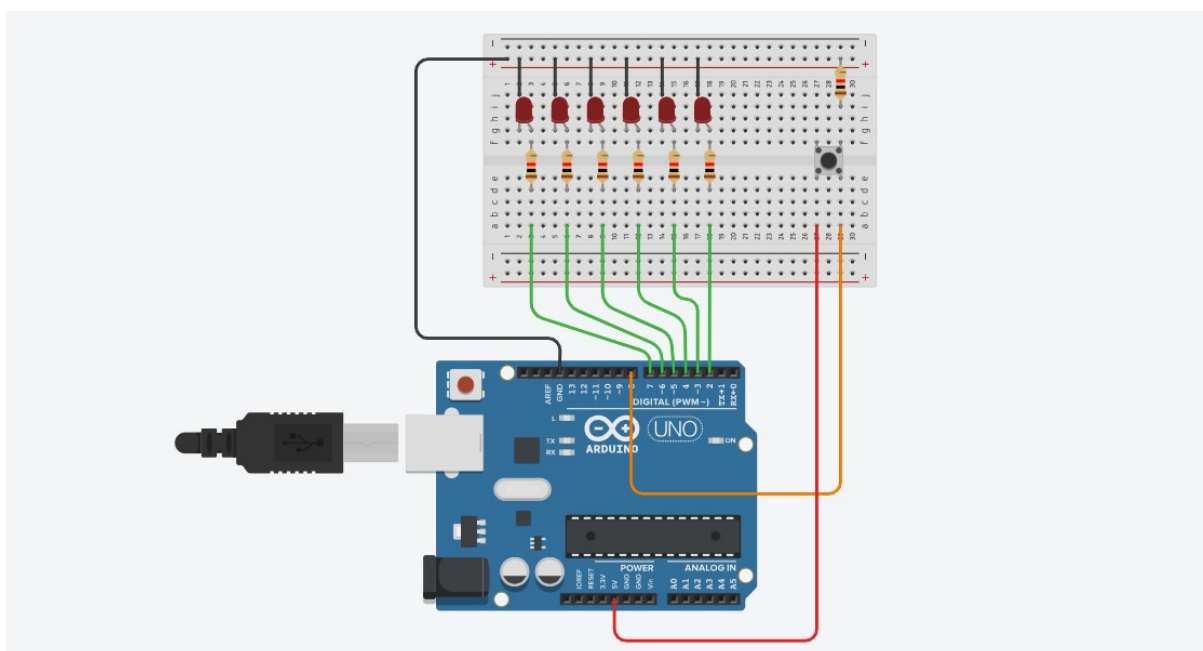
Trong hàm **flashingLights(int delayTime)** thực hiện việc bật led từ chân 4 đến chân 2 (mỗi lần sáng tuần tự thì phải đợi một khoảng delayTime để sáng led tiếp theo), tương tự làm với tắt led để tạo hiệu ứng led di chuyển (line 11 -> line 20). Tương tự, đổi chiều led sáng từ chân 2 đến chân 4 để tạo hiệu ứng led di chuyển quay về (line 22 -> line 31).

Trong hàm **loop()** thực hiện việc đọc tín hiệu analog từ biến trở (chân A0). Nếu value nhận được là 0 thì tắt cả đèn đều tắt. (line 36 - 42) Nếu value bé hơn 342 thì gọi hàm **flashingLights** truyền vào tham số 1500ms để hiệu ứng mỗi led sáng là 1.5s đồng thời in ra màn hình Monitor là **"LOW"** (line 43-46). Nếu value thuộc (341,682) thì hàm **flashLights** truyền vào tham số 1000ms (hiệu ứng led lúc này là 1s nhanh hơn so với LOW), in ra màn hình Monitor là **"MEDIUM"** (line 47 - 50). Và nếu không thuộc các trường hợp trên thì tham số truyền vào là 200ms, in ra màn hình Monitor là **"FAST"** (line 51-54), tốc độ di chuyển của led tăng lên đáng kể.

Câu hỏi 3. Mô phỏng trò chơi “nén xúc sắc**”, xây dựng kịch bản gồm có 6 đèn và 1 nút bấm. Khi bấm lần đầu tiên thì tắt cả các đèn lần lượt sáng lên và hiển thị lên số mà xúc sắc tung ra dựa vào số đèn được hiển thị. Số đèn này sẽ phải dừng lại 1 giây để người dùng theo dõi và sẽ chớp tắt cùng lúc 6 lần. Cứ tiếp tục như thế khi bấm nút lần tiếp theo.**

1. Minh chứng:

Link demo bài 3: [Tại đây](#)



Hình 5. Mô hình bài 3



```
1  int state = 0;
2  bool isFirst = false;
3  int numOfFlash = 0;
4  void setup() {
5      pinMode(8, INPUT);
6      for (int i = 2; i < 8; i++) {
7          pinMode(i, OUTPUT);
8      }
9      Serial.begin(9600);
10 }
11 void loop() {
12     int btn = digitalRead(8);
13
14     delay(200);
15     if (btn == HIGH) {
16         if (isFirst == false)
17         {
18             for (int i = 2; i < 8; i++) {
19                 digitalWrite(i, HIGH);
20                 delay(100);
21                 digitalWrite(i, LOW);
22                 delay(100);
23             }
24             isFirst = true;
25         }
26         int n = random(0, 6) + 1;
27         Serial.println(n);
28
29         state = n;
30         numOfFlash = 0;
31         for(int i = 2; i < n + 2; i++){
32             digitalWrite(i, HIGH);
33         }
34         for(int i = n; i < 7 ;i++){
35             digitalWrite(i+2, LOW);
36         }
37         numOfFlash += 1;
38     }
39     else {
40         if (numOfFlash <= 6 )
41         {
42             for(int i = 0; i < state; i++){
43                 digitalWrite(i+2, HIGH);
44             }
45             delay(1000);
46             for(int i = 0; i < 7; i++){
47                 digitalWrite(i+2, LOW);
48             }
49             numOfFlash++;
50         }
51         else
52         {
53             for(int i = 0; i < state; i++){
54                 digitalWrite(i+2, HIGH);
55             }
56         }
57     }
58 }
```

Hình 6. Code thực thi

2. Giải thích:



Khai báo biến state để lưu trạng thái (số led sáng), khai báo biến `isFirst = false` (kiểm tra xem có phải là lần đầu nhấn nút không), `numOfFlash` để quản lý số lần nháy đèn (cụ thể để xử lý nháy đèn 6 lần rồi dừng). (line 1-3).

Trong hàm **setup()** cấu hình **MODE** cho chân 8 (chân button) là **INPUT** để nhận tín hiệu từ button và chân 2 đến chân 7 (chân dương led) là **OUTPUT** để đưa tín hiệu ra ngoài, khởi tạo thông số kết nối của Monitor.

Trong hàm **loop()** đọc tín hiệu số từ chân 8 (button) kiểm tra xem nếu tín hiệu nhận được là HIGH (có bấm nút) (line 15) thì tiếp tục kiểm tra xem đây là lần bấm đầu tiên hay không (line 16).

- Giả sử đây là lần bấm đầu tiên thì hàm sẽ cho sáng đèn lần lượt từ chân 2 đến chân 7 (line 18-22).
- Sau đó tạo ngẫu nhiên chạy từ 1- 6 (line 26) rồi in số vừa random ra màn hình Monitor.
- Kế tiếp lưu lại trạng thái state (con số random sáng đèn) và đặt `numOfFlash = 0` (số lần chớp đèn) (line 29-30).
- Thực hiện bật số led đã random (số n) (line 31-33) và tắt tất cả số đèn phía sau còn lại (line 34-36). Đồng thời tăng `numOfFlash` lên 1 lần (line 37).

Nếu button không nhận được tín hiệu (không bấm nút) thì kiểm tra xem `numOfFlash` đã đủ 6 lần chưa.

- Nếu chưa đủ 6 lần thì tiếp tục thực hiện nháy đèn (line 42-49) và tăng `numOfFlash` lên thêm 1 lần nháy đèn để lặp lại quá trình cho tới khi nào đủ 6 lần.
- Nếu đã đủ số lần thì sáng toàn bộ đèn (chỉ sáng state số led).

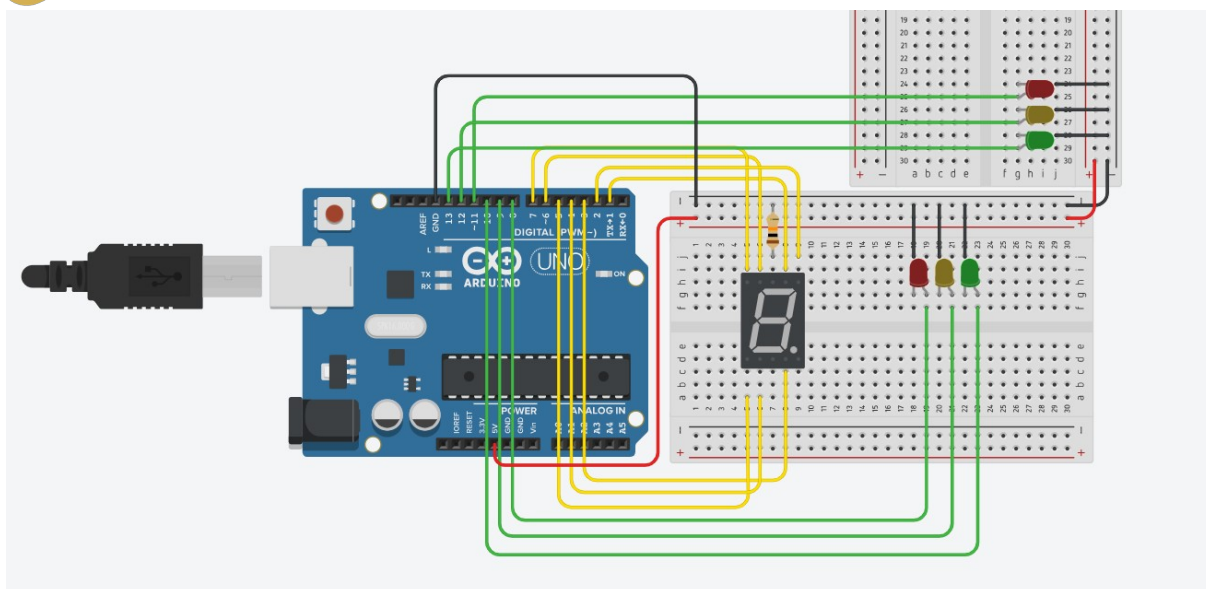
Lưu ý thời gian nháy đèn là 1s theo như yêu cầu của đề bài.

Câu hỏi 4. Mô phỏng “đèn giao thông ở giao lộ”, xây dựng kịch bản gồm có 1 đèn LED 7 đoạn, 6 đèn khác màu (đỏ, vàng, xanh). LED 7 đoạn sẽ đếm ngược từ số lượng thời gian quy định trong mỗi 1 giây, khi đến 0 thì chuyển sang sáng 1 đèn khác. Bắt đầu từ xanh → vàng → đỏ, cứ như thế và lặp lại. Kịch bản gồm 2 giao lộ A và B, khi A sáng đèn xanh hoặc đèn vàng thì đèn đỏ của B sẽ sáng để điều tiết giao thông ở giao lộ và ngược lại.

Ghi chú: đèn đỏ sáng trong 9 giây, đèn vàng sáng trong 3 giây và đèn xanh sáng trong 6 giây.

1. Minh chứng

Link demo bài 4: [Tại đây](#)



Hình 7. Mô hình nối dây bài 4



```
1  const int a = 1;
2  const int b = 2;
3  const int c = 3;
4  const int d = 4;
5  const int e = 5;
6  const int f = 6;
7  const int g = 7;
8
9  const int x1 = 10;
10 const int x2 = 13;
11 const int v1 = 9;
12 const int v2 = 12;
13 const int d1 = 8;
14 const int d2 = 11;
15
16 void setup()
17 {
18     for(int i = 1; i < 14; i++)
19     {
20         pinMode(i, OUTPUT);
21     }
22 }
23
24 void countXanh()
25 {
26     sau();
27     delay(1000);
28     nam();
29     delay(1000);
30     bon();
31     delay(1000);
32     ba();
33     delay(1000);
34     hai();
35     delay(1000);
36     mot();
37     delay(1000);
38     khong();
39     delay(1000);
40 }
41 void countDo()
42 {
43     digitalWrite(x2,HIGH);
44     digitalWrite(v2,LOW);
45     digitalWrite(d2,LOW);
46
47     chin();
48     delay(1000);
49     tam();
50     delay(1000);
51     bay();
52     delay(1000);
53     sau();
54     delay(1000);
55     nam();
56     delay(1000);
57     bon();
58     delay(1000);
59
60     digitalWrite(x2,LOW);
61     digitalWrite(v2,HIGH);
62     digitalWrite(d2,LOW);
63     ba();
```



```
64     delay(1000);
65     hai();
66     delay(1000);
67     mot();
68     delay(1000);
69     khong();
70     delay(1000);
71 }
72 void countVang()
73 {
74     ba();
75     delay(1000);
76     hai();
77     delay(1000);
78     mot();
79     delay(1000);
80     khong();
81     delay(1000);
82 }
83 void chayChuongTrinh()
84 {
85     digitalWrite(x1,HIGH);
86     digitalWrite(v1,LOW);
87     digitalWrite(d1,LOW);
88
89     digitalWrite(x2,LOW);
90     digitalWrite(v2,LOW);
91     digitalWrite(d2,HIGH);
92     countXanh();
93
94     digitalWrite(x1,LOW);
95     digitalWrite(v1,HIGH);
96     digitalWrite(d1,LOW);
97
98     digitalWrite(x2,LOW);
99     digitalWrite(v2,LOW);
100    digitalWrite(d2,HIGH);
101    countVang();
102
103    digitalWrite(x1,LOW);
104    digitalWrite(v1,LOW);
105    digitalWrite(d1,HIGH);
106
107    countDo();
108 }
109 void mot()
110 {
111     digitalWrite(b,LOW);
112     digitalWrite(c,LOW);
113
114     digitalWrite(a,HIGH);
115     digitalWrite(d,HIGH);
116     digitalWrite(e,HIGH);
117     digitalWrite(f,HIGH);
118     digitalWrite(g,HIGH);
119 }
120 void hai()
121 {
122     digitalWrite(a,LOW);
123     digitalWrite(b,LOW);
124     digitalWrite(g,LOW);
```



```
125     digitalWrite(e, LOW);
126     digitalWrite(d, LOW);
127
128     digitalWrite(c, HIGH);
129     digitalWrite(f, HIGH);
130 }
131 void ba()
132 {
133     digitalWrite(a, LOW);
134     digitalWrite(b, LOW);
135     digitalWrite(g, LOW);
136     digitalWrite(c, LOW);
137     digitalWrite(d, LOW);
138
139     digitalWrite(e, HIGH);
140     digitalWrite(f, HIGH);
141 }
142 void bon()
143 {
144     digitalWrite(f, LOW);
145     digitalWrite(g, LOW);
146     digitalWrite(b, LOW);
147     digitalWrite(c, LOW);
148
149     digitalWrite(a, HIGH);
150     digitalWrite(d, HIGH);
151     digitalWrite(e, HIGH);
152 }
153 void nam()
154 {
155     digitalWrite(a, LOW);
156     digitalWrite(f, LOW);
157     digitalWrite(g, LOW);
158     digitalWrite(c, LOW);
159     digitalWrite(d, LOW);
160
161     digitalWrite(b, HIGH);
162     digitalWrite(e, HIGH);
163 }
164 void sau()
165 {
166     digitalWrite(a, LOW);
167     digitalWrite(f, LOW);
168     digitalWrite(g, LOW);
169     digitalWrite(c, LOW);
170     digitalWrite(d, LOW);
171     digitalWrite(e, LOW);
172
173     digitalWrite(b, HIGH);
174 }
175 void bay()
176 {
177     digitalWrite(a, LOW);
178     digitalWrite(b, LOW);
179     digitalWrite(c, LOW);
180
181     digitalWrite(d, HIGH);
182     digitalWrite(e, HIGH);
183     digitalWrite(f, HIGH);
184     digitalWrite(g, HIGH);
185 }
```

```

186 void tam()
187 {
188     digitalWrite(a, LOW);
189     digitalWrite(f, LOW);
190     digitalWrite(g, LOW);
191     digitalWrite(c, LOW);
192     digitalWrite(d, LOW);
193     digitalWrite(e, LOW);
194     digitalWrite(b, LOW);
195 }
196 void chin()
197 {
198     digitalWrite(a, LOW);
199     digitalWrite(f, LOW);
200     digitalWrite(g, LOW);
201     digitalWrite(c, LOW);
202     digitalWrite(d, LOW);
203     digitalWrite(b, LOW);
204
205     digitalWrite(e, HIGH);
206 }
207 void khong()
208 {
209     digitalWrite(a, LOW);
210     digitalWrite(f, LOW);
211     digitalWrite(b, LOW);
212     digitalWrite(c, LOW);
213     digitalWrite(d, LOW);
214     digitalWrite(e, LOW);
215
216     digitalWrite(g, HIGH);
217 }
218 void loop()
219 {
220     chayChuongTrinh();
221 }

```

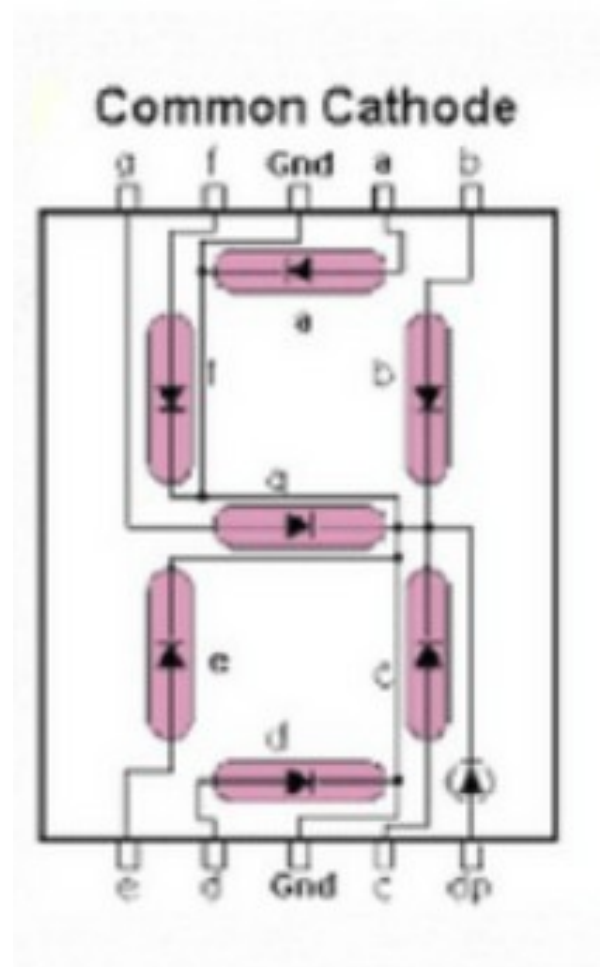
Hình 8. Code thực thi

2. Giải thích

Khai báo biến từ chân 1 đến chân 7 cho các chân của led 7 đoạn. Ở đây sử dụng led 7 đoạn Cathode (Chân COM nối GND).

Trong hàm **setup()** cấu hình **MODE** cho chân 1 đến chân 13 (chân dương led) là **OUTPUT** để đưa tín hiệu ra ngoài.

Trong hàm display number (mot, hai, ba, bon, nam, sau, bay, tam, chin, khong) chỉ dùng để hiển thị led 7 đoạn. (line 109-217)



- Tại các chân của led 7 đoạn thì digitalWrite ở mode LOW để led vị trí đó có thể sáng do led 7 đoạn nhóm đang sử dụng là loại Cathode (ví dụ số một thì chân b, c mode LOW, còn chân a, d, e, f, g mode HIGH). Chân COM nối GND.

Trong hàm **countXanh** thì thực hiện chạy hàm sau -> nam -> bon -> ba -> hai -> mot -> khong. Mỗi lần chuyển hàm thì delay 1000ms tương ứng 1s (Mục đích khiến cho led 7 đoạn hiển thị đếm ngược từ 6 -> 1). Tương tự, cho hàm **countVang** là 3 -> 0.

Tuy nhiên trong hàm **countDo** sẽ thực hiện việc hiển thị display đếm ngược led 7 đoạn từ 9 -> 0, đèn đỏ chiều thuận bật, trong lúc này, đèn vàng ở chiều đi khác tắt đèn xanh ở chiều đi đó bật. Khi đếm ngược đến 3 thì đèn vàng đó chuyển thành bật và xanh chuyển thành tắt. (line 41-71)

Trong hàm **chayChươngTrinh** :

- Bật led xanh1 thì led do1, vang1, xanh2, vang2 tắt, do2 bật đồng thời khởi chạy hàm countXanh (đếm ngược từ 6-0 trên led 7 đoạn) (line 85-92) ← **Khái quát**: Bật đèn xanh ở chiều ngang thì chỉ có đèn đỏ ở đường chiều dọc sáng đèn.



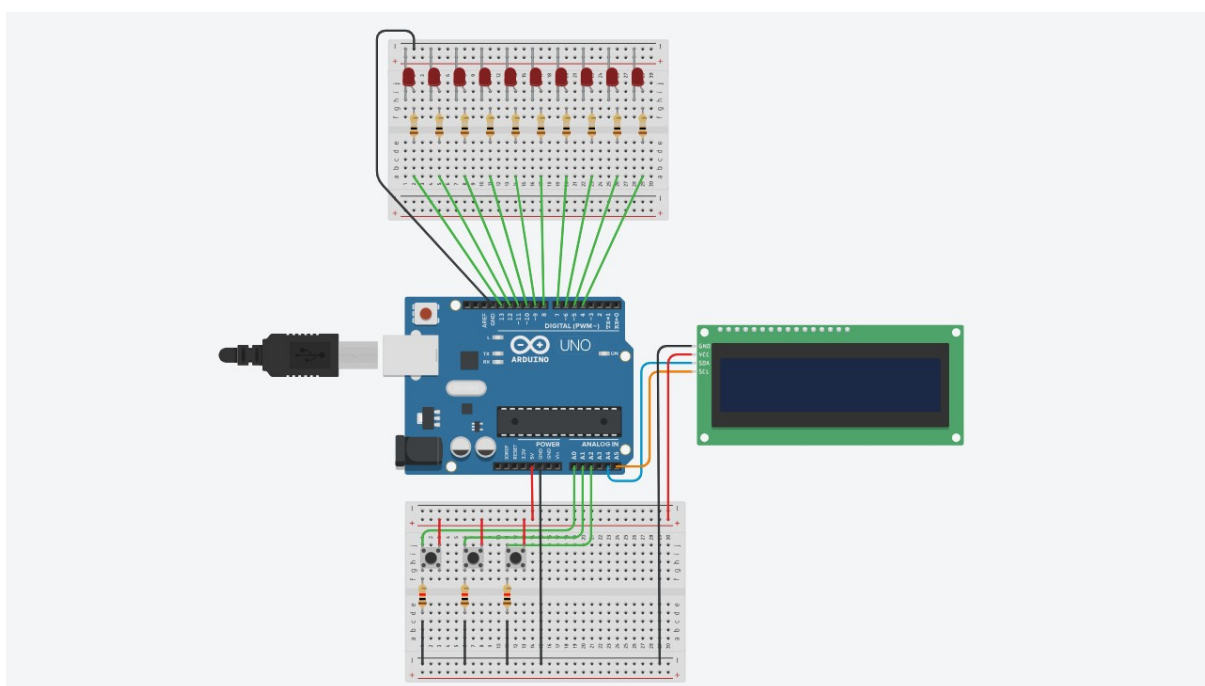
- Tiếp đến bật led vang1 thì led do1, xanh1, xanh2, vang2 tắt, do2 bật đồng thời khởi chạy hàm countVang (đếm ngược từ 6-0 trên led 7 đoạn) (line 94 -101). ← **Khái quát:** Kết thúc bật đèn xanh ở chiều ngang thì chỉ đèn vàng ở chiều ngang và đèn đỏ ở đường chiều dọc sáng đèn.
- Tiếp đến led do1 bật, led xanh1, vang1, do2 tắt, xanh2 và vang2 sáng theo điều kiện của hàm countDo (cụ thể là led 7 đoạn từ 9 -3 là led xanh2 sáng, 3-0 thì led vang2 sáng) (line 103-107). ← **Khái quát:** Bật đèn đỏ ở chiều ngang thì chỉ có đèn xanh và vàng ở đường chiều dọc sáng đèn theo điều kiện.

Trong hàm **loop()** sẽ thực hiện chạy hàm **chayChươngTrinh** liên tục để tạo vòng tuần hoàn.

Câu hỏi 5. Mô phỏng trò chơi “thử tài đoán số”, xây dựng kịch bản gồm có 10 đèn, 3 nút bấm. Trò chơi gồm có 3 level, tùy vào tốc độ nháy đèn để hiển thị. Người chơi sẽ chọn 1 trong 3 nút bấm với điều kiện như sau để cộng điểm: chọn vào nút chính xác theo qui tắc (số lượng đèn sáng % 3), nếu chọn sai hoặc không chọn trong vòng (4 - level) giây thì điểm sẽ bị trừ. Nếu chọn sai thì mặc định level sẽ reset về 1. Các thông tin về điểm số và tình trạng cộng điểm hay trừ điểm sẽ được hiển thị qua màn hình LCD.

1. Minh chứng

Link demo bài 5: [Tại đây](#)



Hình 9. Mô hình nối dây bài 5



```
1  #include <Adafruit_LiquidCrystal.h>
2  Adafruit_LiquidCrystal lcd_1(0x27);
3  int level = 1;
4  int point = 0;
5
6  void setup() {
7      lcd_1.begin(16, 2);
8      for(int i = 4; i < 14; i++)
9      {
10         pinMode(i, OUTPUT);
11     }
12     pinMode(A0, INPUT);
13     pinMode(A1, INPUT);
14     pinMode(A2, INPUT);
15
16     Serial.begin(9600);
17 }
18 void ledOnOff(int d, int num)
19 {
20     for (int i=1; i<=num; i++) {
21         digitalWrite(i+3, HIGH);
22     }
23     delay(d);
24     for (int i=1; i<=10; i++) {
25         digitalWrite(i+3, LOW);
26     }
27     delay(200);
28 }
29 void ledOn(int num, int Level) {
30     for (int i = 0; i < 4; i++) {
31         if (level == 1 )
32         {
33             ledOnOff(1000, num);
34         }
35         else if (level == 2)
36         {
37             ledOnOff(600, num);
38         }
39         else ledOnOff(300, num);
40     }
41 }
42
43 void printLCD() {
44     int n = random(11);
45
46     int ans = n%3;
47     Serial.println(ans);
48     ledOn(n, level);
49
50     lcd_1.setCursor(0, 0);
51     lcd_1.print("LEVEL: ");
52
53     lcd_1.setCursor(6, 0);
54     lcd_1.print(level);
55
56     lcd_1.setCursor(0, 1);
57     lcd_1.print("POINT: ");
```



```
59 lcd_1.setCursor(6, 1);
60 lcd_1.print(point);
61
62 lcd_1.setCursor(9, 0);
63 lcd_1.print("TIME: ");
64 int b0, b1, b2;
65 for (int i=4-level; i>=0; i--) {
66     lcd_1.setCursor(14, 0);
67     lcd_1.print(i);
68
69     b0 = digitalRead(A0);
70     b1 = digitalRead(A1);
71     b2 = digitalRead(A2);
72
73     if (ans == 0) {
74         if (b0 == 1) {
75             point ++;
76             if (point%3 == 0)
77                 level ++;
78             return ;
79         }
80         else if (b1==1 || b2==1) {
81             level = 1;
82             point = 0; //f
83             return ;
84         }
85     }
86
87     if (ans == 1) {
88         if (b1 == 1) {
89             point ++;
90             if (point%3 == 0)
91                 level ++;
92             return ;
93         }
94         else if (b0==1 || b2==1) {
95             level = 1;
96             point = 0; //f
97             return ;
98         }
99     }
100
101     if (ans == 2) {
102         if (b2 == 1) {
103             point ++;
104             if (point%3 == 0)
105                 level ++;
106             return ;
107         }
108         else if (b0==1 || b1==1) {
109             level = 1;
110             point = 0; //f
111             return ;
112         }
113     }
114     delay(1000);
115 }
```

```

116     if (b0 == 0 && b1 == 0 && b2 == 0) {
117         level = 1;
118         point = 0; //f
119     }
120
121 }
122 void nextRound()
123 {
124     lcd_1.setCursor(0, 1);
125     lcd_1.print("Next round in: ");
126     for (int i=5; i>=0; i--) {
127         lcd_1.setCursor(15, 1);
128         lcd_1.print(i);
129         delay(1000);
130     }
131     lcd_1.clear();
132 }
133
134
135 void printResult() {
136     lcd_1.clear();
137     if (point == 0) {
138         lcd_1.setCursor(0, 0);
139         lcd_1.print("GAME OVER!");
140         lcd_1.setCursor(0, 1);
141         lcd_1.print("New game ...");
142         point = 0;
143     }
144
145     if (point > 7) {
146         level = 1;
147         point = 0;
148         lcd_1.setCursor(0, 0);
149         lcd_1.print("YOU WIN! POINT=");
150         lcd_1.setCursor(15, 0);
151         lcd_1.print(point);
152         lcd_1.setCursor(0, 1);
153         lcd_1.print("New game...");
154     }
155     else {
156         lcd_1.setCursor(0, 0);
157         lcd_1.print("GOODJOB POINT+1");
158         lcd_1.setCursor(0, 1);
159         lcd_1.print("Point: ");
160         lcd_1.setCursor(7,1);
161         lcd_1.print(point);
162         lcd_1.setCursor(9,1);
163         lcd_1.print("Lv: ");
164         lcd_1.print(level);
165     }
166
167 }
168     delay(2000);
169 }
170 void loop()
171 {
172     printLCD();
173     printResult();
174     nextRound();
175 }

```

Hình 10. Code thực thi

2. Giải thích

Tóm tắt trò chơi:



- Chương trình ngẫu nhiên lựa chọn 1 số và biểu diễn số led sáng tương ứng. Tùy vào level của người chơi mà mức độ nhấp đèn là nhanh hay chậm, số lần nhấp đèn ở mỗi level là 4 lần như nhau, nhưng tốc độ tắt của đèn là nhanh hơn khi ở level cao hơn.
- Có 3 nút bấm để người chơi lựa chọn đáp án dựa trên số led sáng % 3. Để tăng lên 1 level thì phải trả lời đúng 3 câu liên tiếp. Khi đó thời gian trả lời ở level này bằng 4 - level (s).
- Người chơi đạt được 8 điểm sẽ dành chiến thắng và bắt đầu game lại từ đầu.

Khai báo thư viện hỗ trợ để in ra LCD (sử dụng *AdaFruit_LiquidCrystal.h*) đồng thời khởi tạo địa chỉ và loại lcd cho I2C. (line 1 - 2, line 7).

Khai báo và khởi tạo giá trị cho level = 1 (level mặc định ban đầu), point = 0 (điểm mặc định) (line 3-4).

Trong hàm **setup()** cấu hình **MODE** cho chân 4 đến chân 13 (chân dương led) là **OUTPUT** để đưa tín hiệu ra ngoài. Chân A0, A1, A2 (các button) **MODE** là **INPUT** để đọc được tín hiệu (đáp án của người chơi). (line 6 - 17).

Hàm **ledOnOff(int d, int num)** là hàm hiển thị số **num** led sáng và delay một khoảng **d** time rồi tắt tất cả (delay này ảnh hưởng đến độ khó của game, khi d càng bé thì đèn tắt càng nhanh). (line 18-28)

Hàm **ledOn(int num, int level)** chính là hàm thực thi ledOnOff 4 lần, nó đảm nhận việc cấp thời gian **d** cho **ledOnOff** tùy theo level của người chơi. (line 29-41)

Trong hàm **printLCD()**:

- Random ngẫu nhiên một số n (đây chính là đề bài câu hỏi cho người chơi). Kết quả của chương trình sẽ được tính bằng $n\%3$. (line 44, 46)
- Sau đó khởi chạy hàm **ledOn** truyền vào tham số n và level, lúc này đèn led sẽ nhấp nháy 4 lần tốc độ dựa trên level hiện tại của người chơi (line 48).
- Đồng thời hiển thị ra màn hình level, point hiện tại và thời gian đếm ngược để trả lời (line 50, 63). Thời gian trả lời sẽ bằng 4 - level (s) (line 65).
- Trong dòng for đợi 4s trả lời thì đọc tín hiệu A0, A1, A2 (đáp án bấm nút của người chơi) (line 69-71):
 - o Giả sử người chơi chọn b0 (kết quả $n\%3 = 0$) là kết quả đúng thì point + 1 (line 75). Kiểm tra xem nếu người chơi đã vượt mốc 3 điểm thì sẽ chuyển sang level cao hơn (line 76 - 77). Nếu kết quả sai thì level sẽ quay lại 1 và điểm bắt đầu lại 0 (line 81, 82). Và đồng thời lcd in ra "GOODJOB POINT+1".

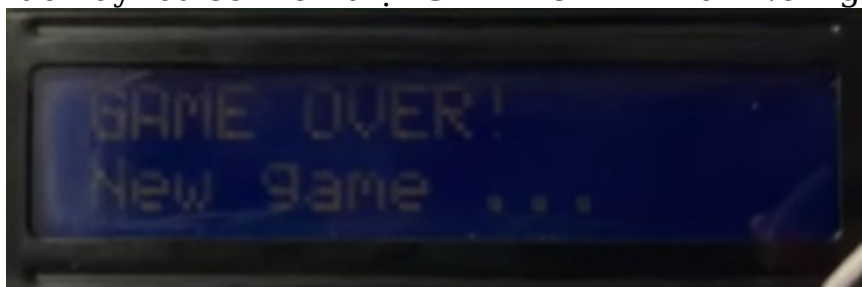


- o Tương tự với b1,và b2.
- o Nếu trường hợp không nhận được bất kì tín hiệu nào từ các button và hết thời gian trả lời thì level, point quay về mặc định (point =0, level = 1) (line 116-119).

Hàm **nextRound()** chỉ để mục đích đợi 6s khi bắt đầu round tiếp theo và hiển thị ra màn hình lcd (line 122,132).

Hàm **printResult()** thực hiện reset lcd (line 136) để làm sạch lcd cho vòng lặp tiếp theo. Kiểm tra xem point sau khi đã chạy xong hàm **printLCD**:

- Nếu point = 0 là đồng nghĩa người dùng đã trả lời sai nên lúc này lcd sẽ hiển thị "GAME OVER" và "New game.."



- Nếu point > 7 khi này lcd sẽ hiển thị "YOU WIN! POINT=" "New game..." đồng thời đưa level và point lại giá trị mặc



định.

Hàm **loop()** thực thi hàm **printLCD** để chạy trò chơi, sau đó thực thi hàm **printResult** để hiển thị kết quả của câu đó. Cuối cùng là hàm **nextRound** để báo cho người chơi chuẩn bị cho round chơi kế tiếp (line 170-175).



HẾT