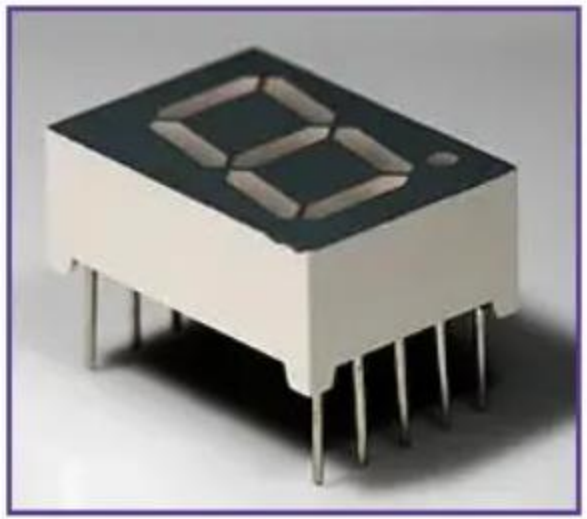
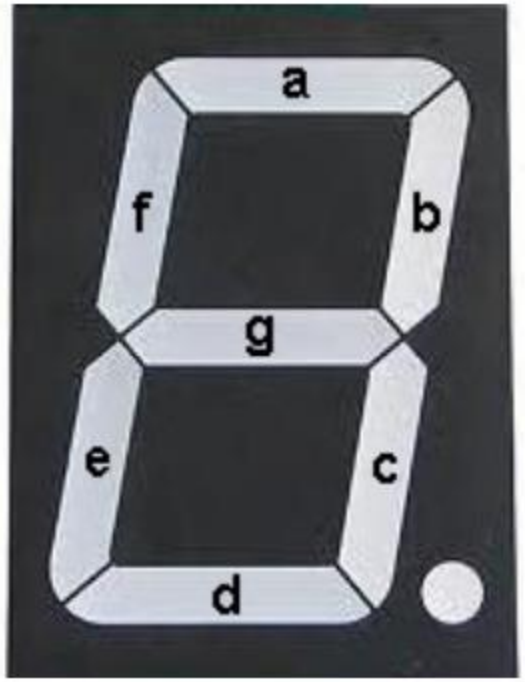
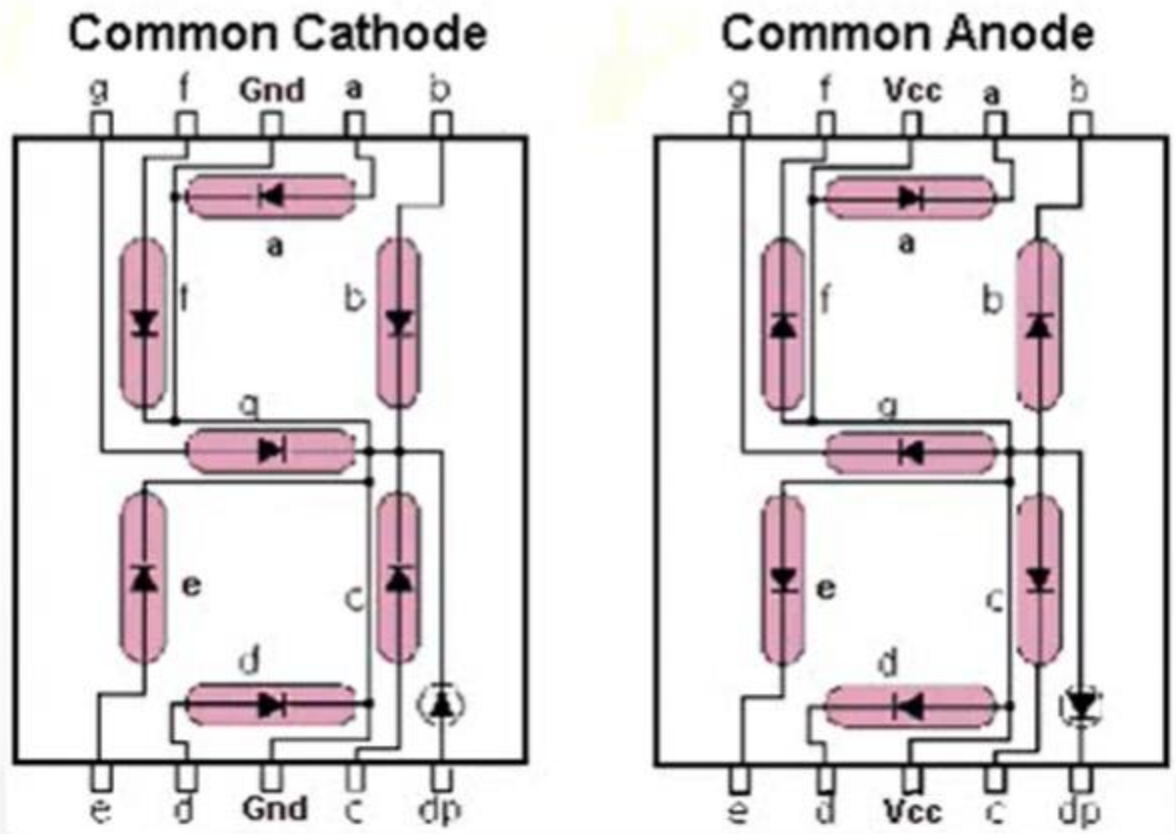
ĐIỀU KHIỂN LED 7 ĐOẠN





+ Có 2 loại: Anode chung (chung cực dương) và Cathode chung (chung cực âm)..



1.3 Lắp mạch

Anode chung:

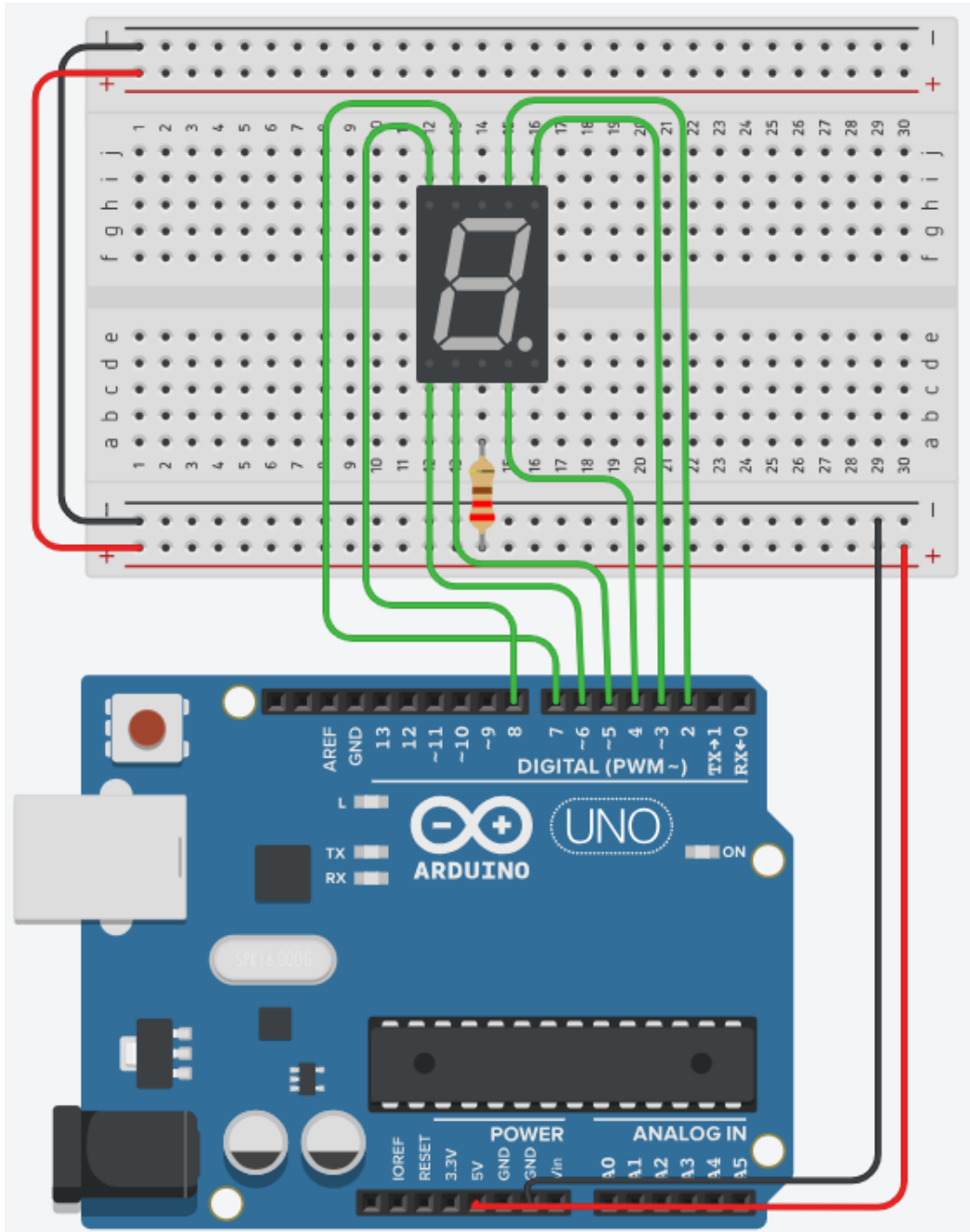
5V - 220Ω – Com (LED)

Các chân Led 7 đoạn nối

Với Uno theo thứ tự: a – 2;

b – 3; c – 4; d – 5; e – 6;

f – 7; g – 8; dp – 9 (nếu dùng)



Cathode chung:

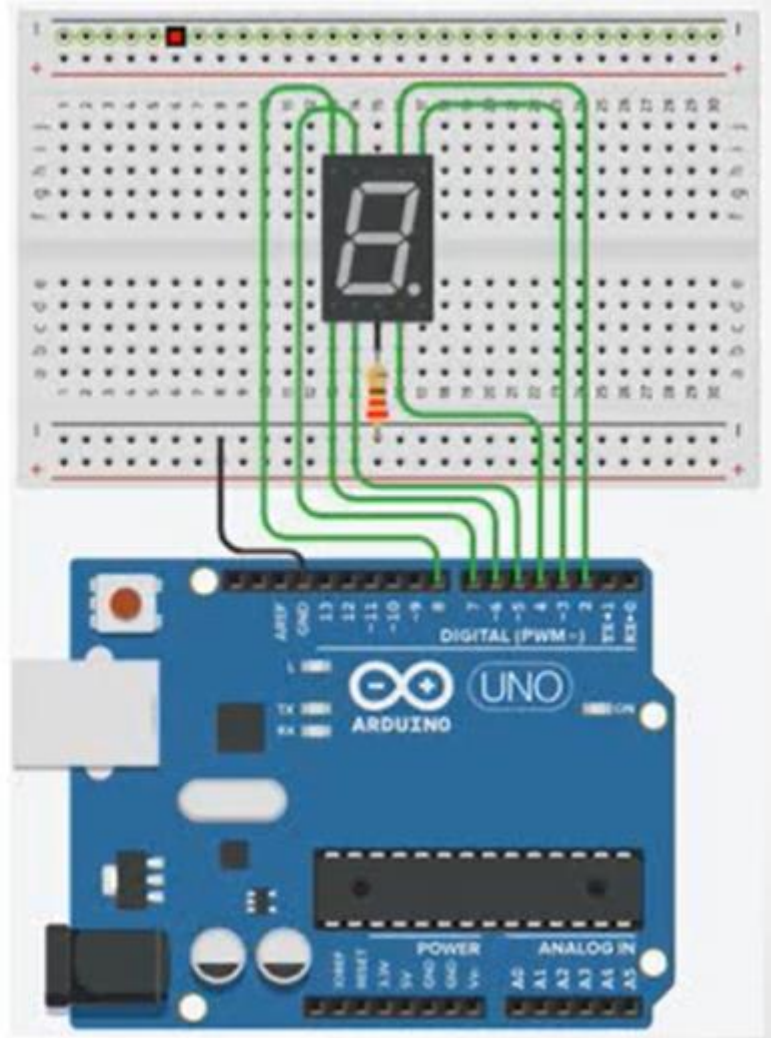
Gnd - 220Ω – Com (LED)

Các chân Led 7 đoạn nối

với Uno theo thứ tự: a – 2;

b – 3; c – 4; d – 5; e – 6;

f – 7; g – 8; dp – 9 (nếu dùng).



1.4 Hướng dẫn lập trình

+ Hiển thị các số trên Led 7 đoạn, phải điều khiển các đoạn bật, tắt theo tổ hợp nhất định.

+ Tổ hợp trạng thái các đoạn a-g, dp có thể mã hóa bằng số nhị phân 8 bit, bắt đầu bằng kí tự B.

Mã có dạng: B-a-b-c-d-e-f-g-dp (với a-g,dp có thể nhận kí tự 1 hoặc 0).

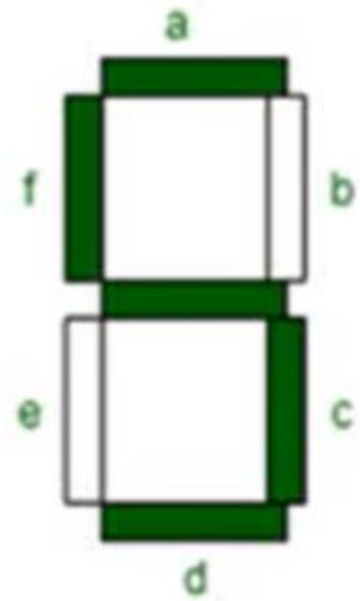
+ Các bit trong mã nhị phân khi đọc sẽ là giá trị tương ứng với các chân **VĐK** (nối với LED 7 đoạn).

+ Ví dụ: Cần hiển thị chữ số thập phân “5” các led sáng, tắt trên LED 7 thanh:

a – sáng; b – tắt ; c – sáng; d – tắt;

e – tắt; f – sáng; g – sáng; dp – tắt.

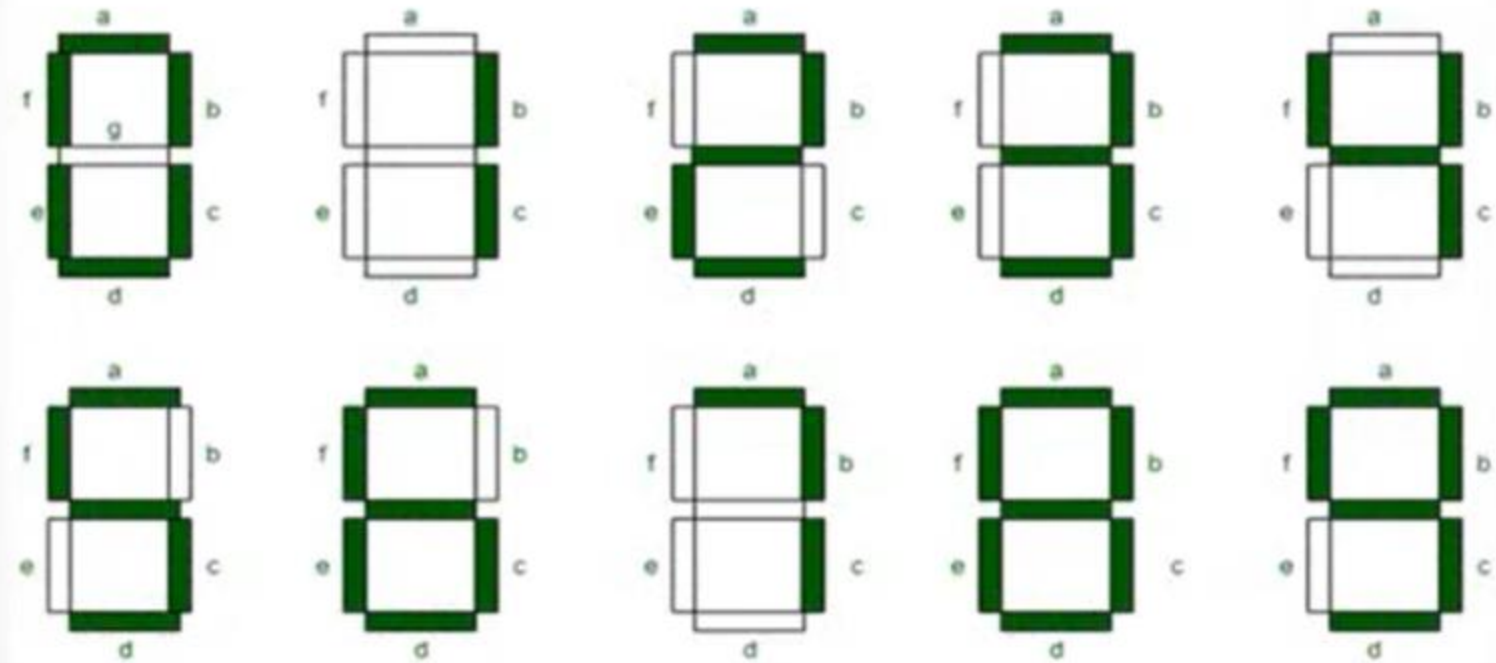
Khi đó mã nhị phân (B-a-b-c-d-e-f-g-dp) tương ứng với trường hợp:



+ Âm chung: B10110110

+ Dương chung: B01001001

Trạng thái các đoạn khi led 7 đoạn hiển thị chữ số:



Bảng mã nhị phân:

Số thập phân	Trạng thái các đoạn (1 - bật, 0 - tắt)								Mã hóa bằng số nhị phân	
	a	b	c	d	e	f	g	dp	âm chung	dương chung
0	1	1	1	1	1	1	0	0	B11111100	B00000011
1	0	1	1	0	0	0	0	0	B01100000	B10011111
2	1	1	0	1	1	0	1	0	B11011010	B00100101
3	1	1	1	1	0	0	1	0	B11110010	B00001101
4	0	1	1	0	0	1	1	0	B01100110	B10011001
5	1	0	1	1	0	1	1	0	B10110110	B01001001
6	1	1	1	1	1	0	1	0	B10111110	B01000001
7	1	1	1	0	0	0	0	0	B11100000	B00011111
8	1	1	1	1	1	1	1	0	B11111110	B00000001
9	1	1	1	0	1	1	1	0	B11101110	B00010001

- Khai báo

+ Khai báo mảng chứa mã nhị phân:

Chọn mã tương ứng với loại Led 7 đoạn (dương chung hoặc âm chung)

Sử dụng mảng (kiểu byte):

```
const byte ma_led7[10] = {…,
                          …,
                          …}
```

- Âm chung

```
const byte ma_led7[10] = {
                          B11111100, // số 0
                          B01100000, // số 1
                          B11011010, // số 2
                          B11110010, // số 3
                          B01100110, // số 4
                          B10110110, // số 5
                          B10111110, // số 6
                          B11100000, // số 7
                          B11111110, // số 8
                          B11101110 // số 9
                          }
```

- Dương chung

```
const byte ma_led7[10] = {
                          B00000011, // số 0
                          B10011111, // số 1
                          B00100101, // số 2
                          B00001101, // số 3
                          B10011001, // số 4
                          B01001001, // số 5
                          B01000001, // số 6
                          B00011111, // số 7
                          B00000001, // số 8
                          B00010001, // số 9
                          }
```

+ Khai báo các chân Uno: từ chân 2 - 9

+ Bằng mảng (kiểu int), tốt nhất nên sử dụng các chân cạnh nhau và theo đúng thứ tự các đoạn từ a–g, dp.

```
int chan_led7[8] = {2,3,4,5,6,7,8,9}
```

- Thiết đặt trong hàm setup()

+ Thiết lập chân trên Uno: Có thể thiết lập từng chân bằng những lệnh đơn, ví dụ: pinMode(4, OUTPUT), như vậy cần số dòng lệnh bằng số chân cần thiết lập, hoặc sử dụng vòng lặp và mảng chứa các chân Uno đã khai báo.

```
for (int i=0; i<=7; i++)
    pinMode(chan_led7[i], OUTPUT);
```

+ Thiết lập Serial để giám sát

```
Serial.begin(9600);
```

+ Viết hàm gửi tín hiệu hiển thị các số trên Led 7 đoạn

Mã nhị phân B-a-b-c-d-e-f-g-dp

```
void PrintLed7(int number)
{
  number = constrain(number,0,9);
  boolean isBitSet;
  for (int i = 7; i >= 0; i--)
    {
      isBitSet = bitRead(MaLed7[number], i);
      digitalWrite(Led7Pins[i], isBitSet);
    }
}
```

- Vòng lặp loop()

Sử dụng vòng lặp để điều khiển số cần hiển thị tăng, giảm, số chẵn, số lẻ…

Sử dụng hàm printLed7(i) với số cần hiển thị để đưa tín hiệu điều khiển tới Led 7 đoạn hiển thị đúng chữ số.

Bài tập lập trình

Lập trình điều khiển led 7 đoạn trong các trường hợp sau:

- Hiển thị lần lượt các số từ 0 đến 9, khoảng trễ giữa các số là 2s.
- Hiển thị các số từ 0 – 9 và ngược lại từ 9 – 0, khoảng trễ 2s.
- Hiển thị số chẵn: 0, 2, 4, 6, 8 và số lẻ: 1, 3, 5, 7.

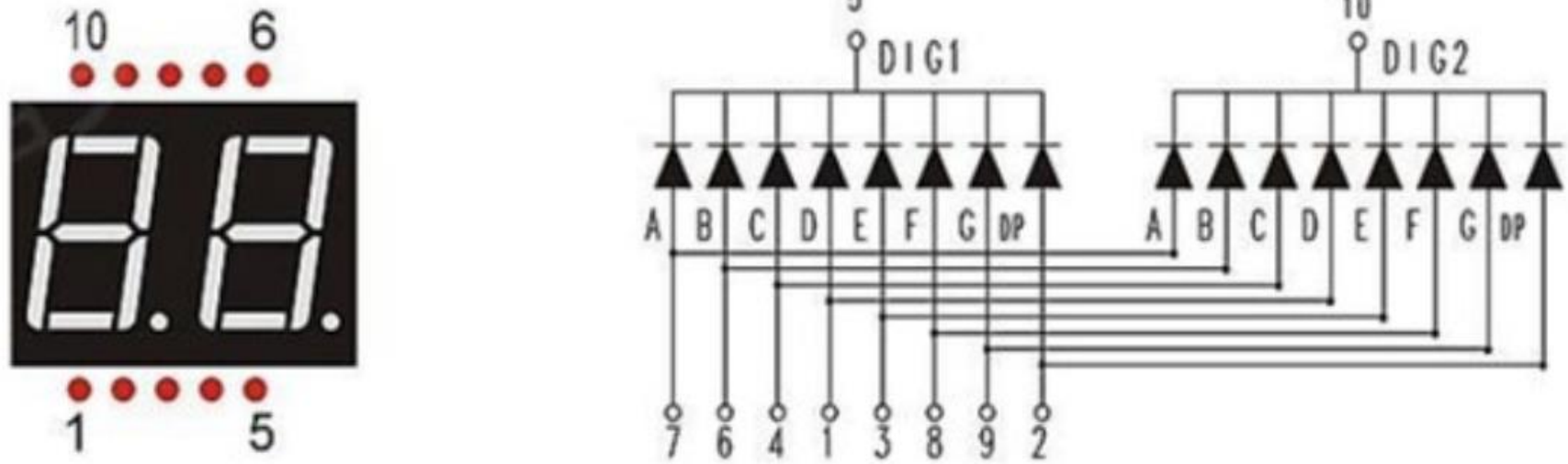
2. Điều khiển modun led 7 đoạn

2.1. Module Led 7 đoạn

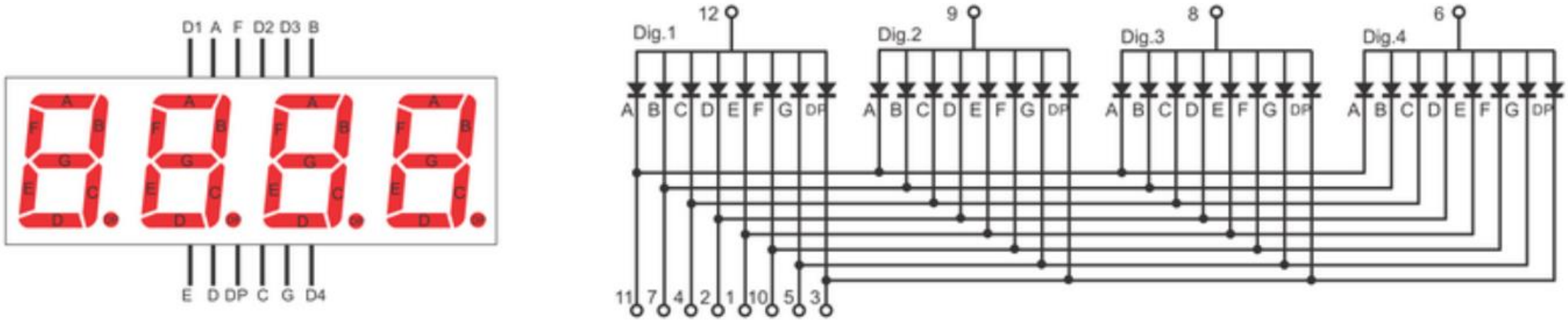
+ Module Led 7 đoạn là sự kết hợp nhiều Led 7 đoạn đơn để có thể hiển thị các số co nhiều chữ số.

+ Module có các loại 2 chữ số, 3 chữ số và 4 chữ số

+ Cấu tạo module Led 7 đoạn tương tự như led 7 đoạn đơn, tuy nhiên các chân a – f và dp chung cho tất cả các Led 7 đoạn đơn, ngoài ra mỗi Led sẽ có một chân chung, được kí hiệu là digit (chữ số).



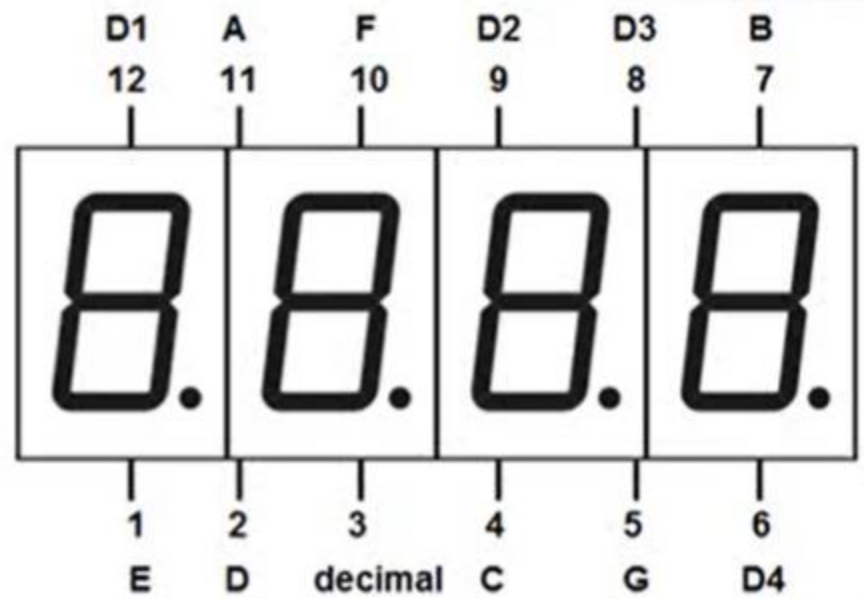
Hình module Led 7 đoạn 2 chữ số âm chung



4 Digit - 7 Segment Display - Pin Out

2.2 Lắp mạch

LED 7	Uno
1 (e)	6
2 (d)	5
3 (dp)	9
4 (c)	4
5 (g)	8
6 (D4)	13
7 (b)	3
8 (D3)	12
9 (D2)	11
10 (f)	7
11 (a)	2
12 (D1)	10



Chân a-f,dp: 2-9

Chân D1, D2, D3, D4: trở 220Ω + (10, 11, 12, 13)

2.3 Hướng dẫn lập trình

- Khai báo:

+ Khai báo mã nhị phân: Chọn mã tương ứng với Led 7 đoạn và sử dụng mảng (Qui tắc mã B-a-b-c-d-e-f-g-dp)

```
const byte MaLed7[10] = {Bxxxxxxxx,  …} // loại dương chung
```

+ Khai báo chân (Uno) nối với các đoạn a - g,dp của led 7 đoạn bằng mảng

```
const int Led7Pins[8] = {2,3,…,9}; //a-g,dp
```

- Khai báo:

+ Khai báo chân (Uno) kết nối với các chân D1, D2, D3, D4 của Led 7 đoạn (có thể dùng bí danh để thuận lợi trong lập trình).

```
int cs_nghin = 10;        // D1
int cs_tram = 11;         // D2
int cs_chuc = 12;         // D3
int cs_donvi = 13;        // D4
```

- **Thiết lập setup()**

+ Thiết lập chân của Led7Pins[] và Digit:

```
for (int i = 0; i <= 7; i++)
    pinMode(Led7Pins[i], OUTPUT);
pinMode(cs_nghin, OUTPUT);
pinMode(cs_tram, OUTPUT);
pinMode(cs_chuc, OUTPUT);
pinMode(cs_donvi, OUTPUT);
```

+ Có thể thiết lập Serial để kiểm soát

- **Viết hàm hiển thị cữ số trên led 7 đoạn**

```
void PrintLed7(int number) {
        number = constrain(number,0,9);
        boolean isBitSet;
        for (int i = 7; i >= 0; i--) {
            isBitSet = bitRead(MaLed7[number], i);
            digitalWrite(Led7Pins[i], isBitSet);
        }
}
```

- Vòng lặp loop()

+ Tách các chữ số (ví dụ một số có 2 chữ số so = 68) và gọi hàm printLed7() để hiển thị từng led7:

```
        int dv = so % 10;  // dv = 8 chia dư lấy được số 8
// cho led7 hiển thị số đơn vị sáng
        digitalWrite(cs_donvi, 0);
// cho led7 hiển thị số hàng chục tắt
        digitalWrite(cs_chuc, 1);
// hiển thị chữ số 8 ở vị trí hàng đơn vị
        printLed7(dv);
```

```
// chuc = 6 chia nguyên lấy số 6
        int chuc = so / 10;
// cho led7 hiển thị số đơn vị tắt
        digitalWrite(cs_donvi, 1);
// cho led7 hiển thị số hàng chục sáng
        digitalWrite(cs_chuc, 0);
// hiển thị chữ số 6 ở vị trí hàng chục
        printLed7(chuc);
```

+ Mô đun 4 Led 7 đoạn, có thể hiển thị 4 chữ số tương ứng là hàng đơn vị, chục, trăm, nghìn (ví dụ số có 4 chữ số so = 6789) nên phải thêm các biến tram, nghin.

```
int dv = (so % 100) % 10; // 6789 % 100 = 89 % 10 = 9
int chuc = (so % 100) / 10;  // 6789 % 100 = 89 / 10 = 8
int tram = (so / 100) % 10; // 6789 / 100 = 67 % 10 = 7
int nghin = (so / 100) / 10; // 6789 / 100 = 67 / 10 = 6
```

Bài tập lập trình module 4 led 7 đoạn:

- Hiển thị các số tự nhiên từ 0 đến 9999, độ trễ 0,3s.
- Hiển thị các số chẵn từ 0 đến 9998, độ trễ 0,3s.

3. Điều khiển module LED 7 đoạn dùng IC 74HC595

3.1. Module Led 7 đoạn dùng IC 74HC595

Module được thiết kế dễ dàng điều khiển và hiển thị thông tin lên 4 led 7 đoạn chỉ với 3 chân giao tiếp thông qua IC ghi dịch 74HC595.

Ngoài ra mạch còn có khả năng mở rộng thêm các led tiếp theo qua cổng đầu ra nối tiếp, mạch có bộ thư viện đi kèm dễ sử dụng

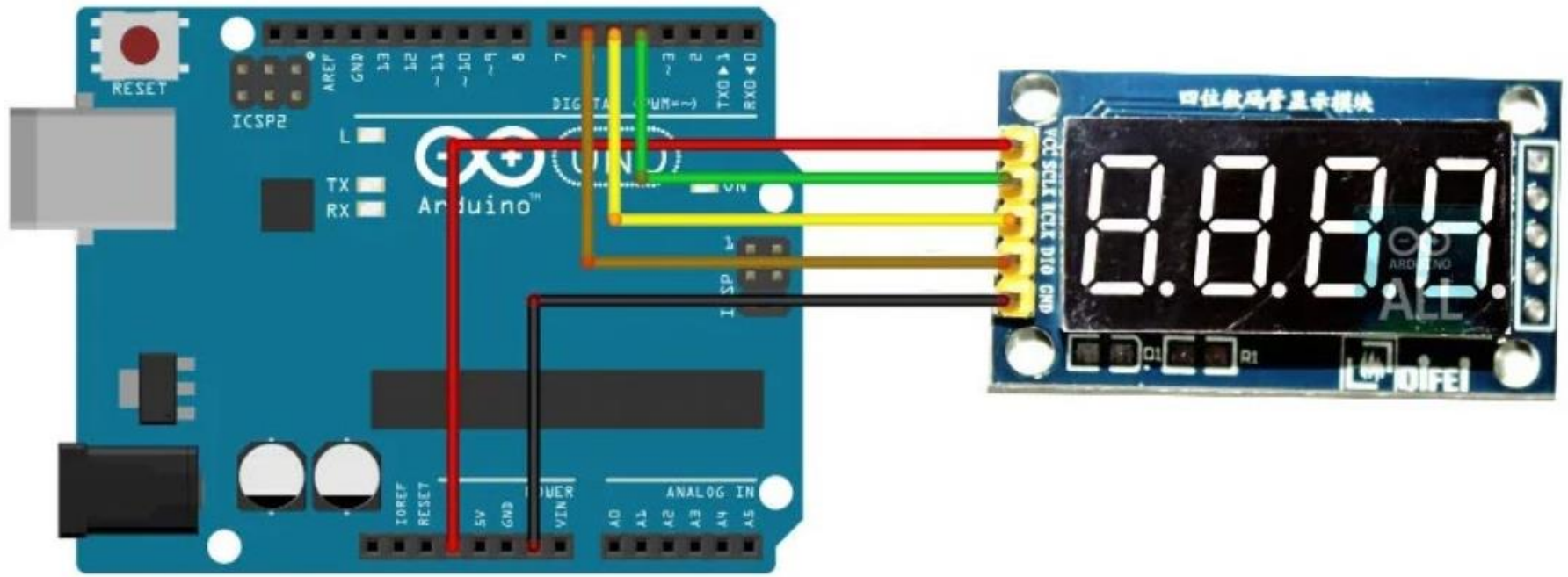
3.1. Module Led 7 đoạn dùng IC 74HC595

Mô đun gồm 5 chân:

2 chân cấp nguồn (VCC, GND) và

3 chân điều khiển DATA IN (DIO), LATCH (RCLK), CLOCK (SCLK).

Mạch điều khiển lắp theo sơ đồ dưới đây:



Sơ đồ kết nối module Led 7 đoạn 74HC 595 với Arduino Uno

VCC – 5V (Uno)

GND – GND (Uno)

CLOCK (SCLK) – 4 (Uno)

LATCH (RCLK) – 5 (Uno)

DATA IN (DIO) – 6 (Uno)

3.4. Hướng dẫn lập trình

+ Cài đặt thư viện: ShiftRegister74HC595.h

+ Khai báo:

+ Thư viện sử dụng:

```
#include <ShiftRegister74HC595.h>
```

+ Khai báo chân Uno kết nối với module led7.

```
int dataPin = 4;
int clockPin = 5;
int latchPin = 6;
```

+ Tạo đối tượng ghi dịch theo cú pháp:

ShiftRegister74HC595 <số led7 của mô đun> sr (dataPin, clockPin, latchPin);

+ Khai báo mảng chứa mã nhị phân của các số hiển thị trên led 7 đoạn. Lập mã nhị phân tương tự

```
uint8_t  numberB[] = {B11000000, //0
                      ...
                      B10011000  //9
                      };
```

- Viết hàm hỗ trợ:

+ Hàm hiển thị trên mô đun led 7 đoạn

```
void printLed7() {
// Tách số hiển thị thành chữ số đơn vị, chục, trăm, nghìn
int dv    = (so /1000) % 10;
int chuc = (so / 100) % 10;
int tram = (so / 10) % 10;
int nghin  = (so /10);
```

```
// Gửi các chữ số đến mô đun led 7 đoạn
uint8_t numberToPrint[]= {numberB[nghin], numberB[tram],
numberB[chuc], numberB[dv]};
sr.setAll(numberToPrint); }
```

- **Vòng lặp void loop():**

```
// Đếm từ 0 → 9999
for (int i = 0; i<= 9999; i++){
    printLed7(i);
    delay(500); }
// Đếm từ 9999 → 0
for (int i = 9999; i>= 0; i--){
    printLed7(i);
    delay(500); }
}
```