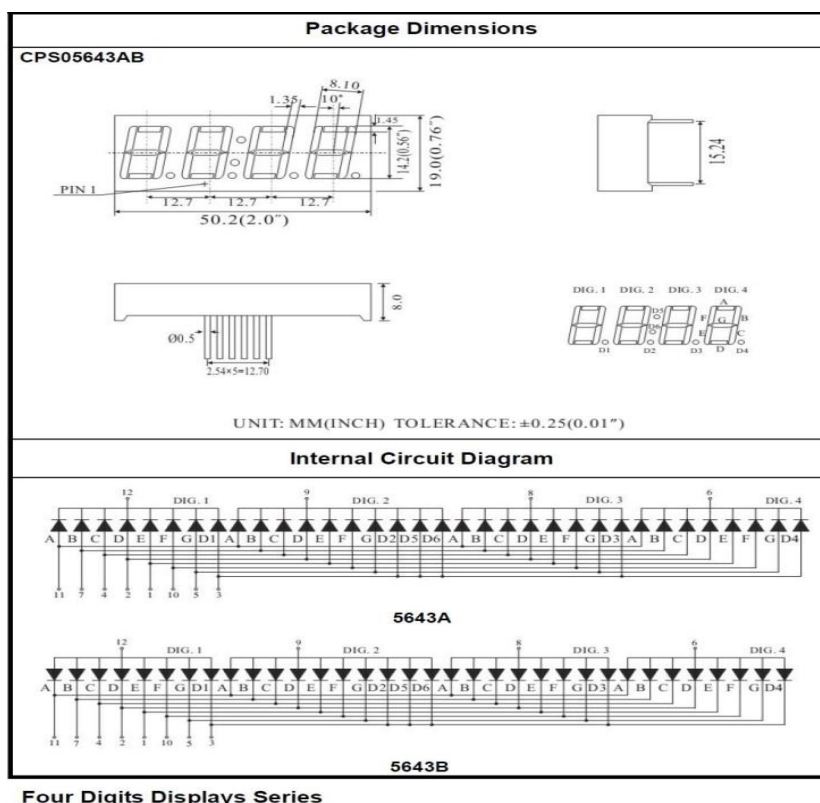
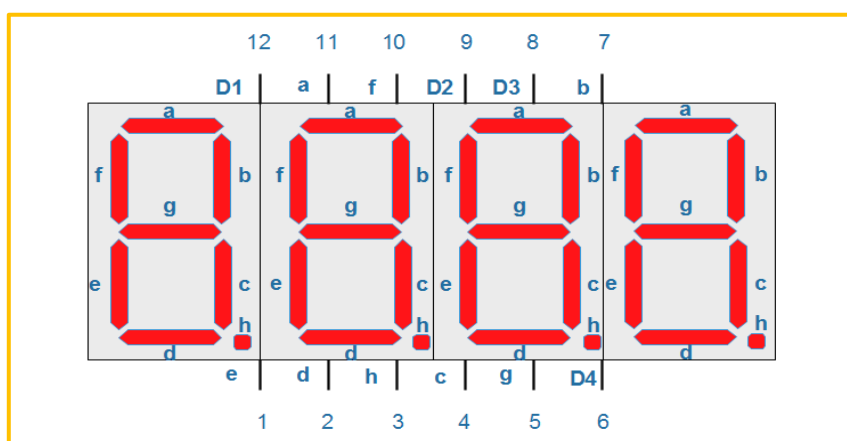


Эксперимент с отображением 4-значного с 7-сегментным цифровым дисплеем

Введение 4-значной 7-сегментной цифровой трубки

Мы использовали 7-сегментные цифровые трубки раньше. Когда мы хотим отобразить несколько чисел, нам нужна многозначная трубка. Здесь мы представляем четыре 7-сегментные цифровые трубки, фактически каждая 7-сегментная цифровая трубка практически такая же, как и трубки, использованные выше. В этом эксперименте мы будем использовать Arduino для управления четырьмя 7-сегментными цифровыми трубками.



Four Digits Displays Series

Четыре цифровые трубки имеют 12 контактов. В верхнем левом углу находится 12 крупнейших стежков. В дополнение к 8 сегментам, используемым для отображения «adbcdefg», есть четыре других контакта D1, D2, D3 и D4, используемые в качестве «битовых» контактов. Когда «битовый» вывод четырехзначной трубки высокий, загорается соответствующая трубка. Принцип отображения четырех цифровых трубок заключается в непрерывном сканировании D1, D2, D3, D4, и затем соответствующие восьмисегментные трубки будут последовательно загораться. Поскольку скорость освещения высока, человеческий глаз ее не увидит, поэтому кажется, что одновременно отображаются четыре цифровых трубки.

Основываясь на принципах, описанных выше, мы теперь создаем имитируемую бомбу обратного отсчета, как в фильме. Бомба взорвется через минуту.

● Экспериментальный принцип

Самая важная цель этой программы - как динамически сканировать четыре цифровые трубки. Фактически, благодаря предыдущему эксперименту с одним цифровым дисплеем, показ четырех цифровых ламп довольно прост. Поскольку это относится к общей анодной трубке, сначала мы устанавливаем D1, D2, D3, D4 на низкий уровень, все светодиоды выключаются, затем выводим таблицу истинности «adbcdefg» на соответствующий порт gpiO, выбираем соответствующий Из бит булавки и продолжайте сканирование. Как добиться 1-минутного отсчета? Мы задерживаем часы примерно на 1 с, вычитаем 1 из времени обратного отсчета и постоянно обновляем время отображения.

● Цель эксперимента

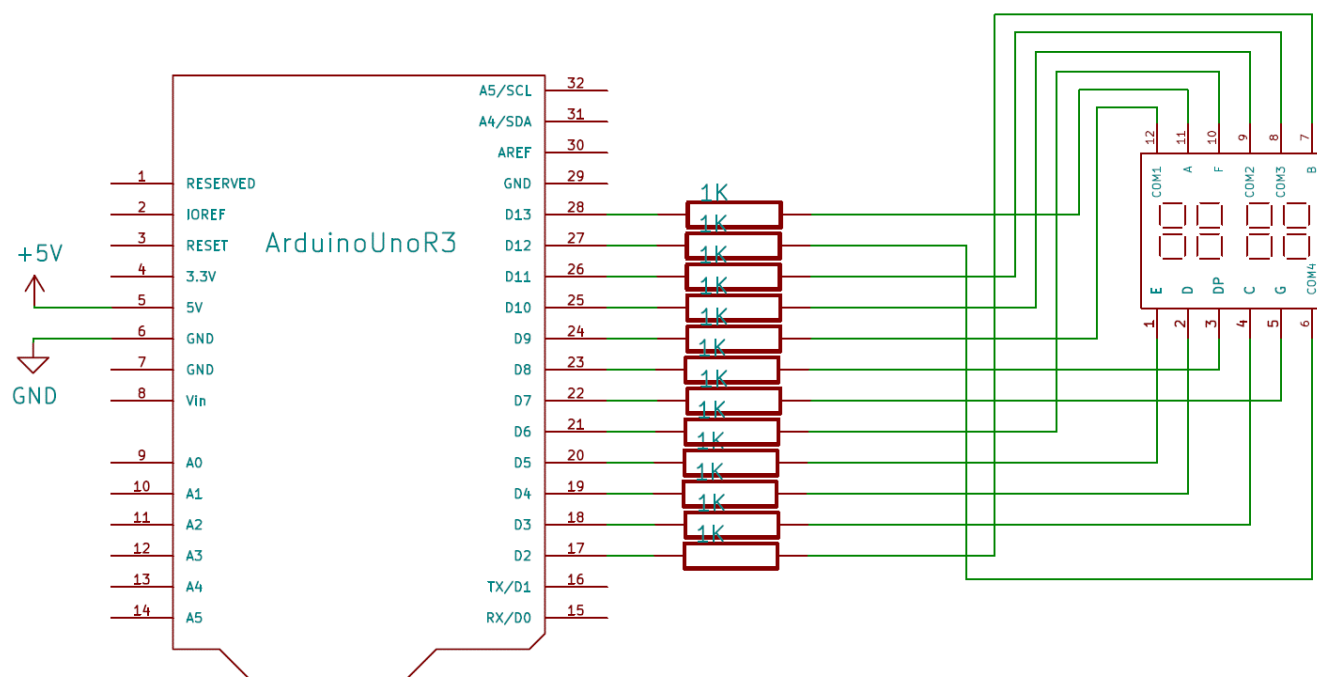
Цель состоит в том, чтобы отобразить 1-минутный отсчет путем динамического сканирования 4-значного 7-сегментного дисплея.

● Компоненты

- Материнская плата Keywish Arduino UNO R3
- Макетная плата
- USB-кабель для передачи данных
- Четырехзначная семисегментная цифровая трубка* 1
- Сопротивление 1к* 12
- Несколько перемычек

Принципиальная схема

4_Digit_7_Segment_Digital_Tube



● Проводка

arduino Uno 控制板	数码管
6	(f)
2	(b)
3	(c)
4	(d)
5	(e)
13	(a)
7	(g)
8	(h)
9	(d1)
10	(d2)
11	(d3)
12	(d4)

● Программа

```
#include "SegmentDisplay.h"

#define LED_A 13      // define Arduino GPIO1 for led a
#define LED_B 2       // define Arduino GPIO2 for led b
#define LED_C 3       // define Arduino GPIO3 for led c
#define LED_D 4       // define Arduino GPIO4 for led d
#define LED_E 5       // define Arduino GPIO5 for led e
#define LED_F 6       // define Arduino GPIO6 for led f
#define LED_G 7       // define Arduino GPIO7 for led g
#define LED_H 8       // define Arduino GPIO8 for led h
#define LED_D1 9
#define LED_D2 10
#define LED_D3 11
#define LED_D4 12

SegmentDisplay _4Bit_7SegmentDisplay(LED_A, LED_B, LED_C, LED_D, LED_E, LED_F, LED_G,
LED_H, LED_D1, LED_D2, LED_D3, LED_D4);

int ShowTime = 60, count = 0;

void setup()
{
    Serial.begin(9600);
    _4Bit_7SegmentDisplay.TurnOffAllLed();
}

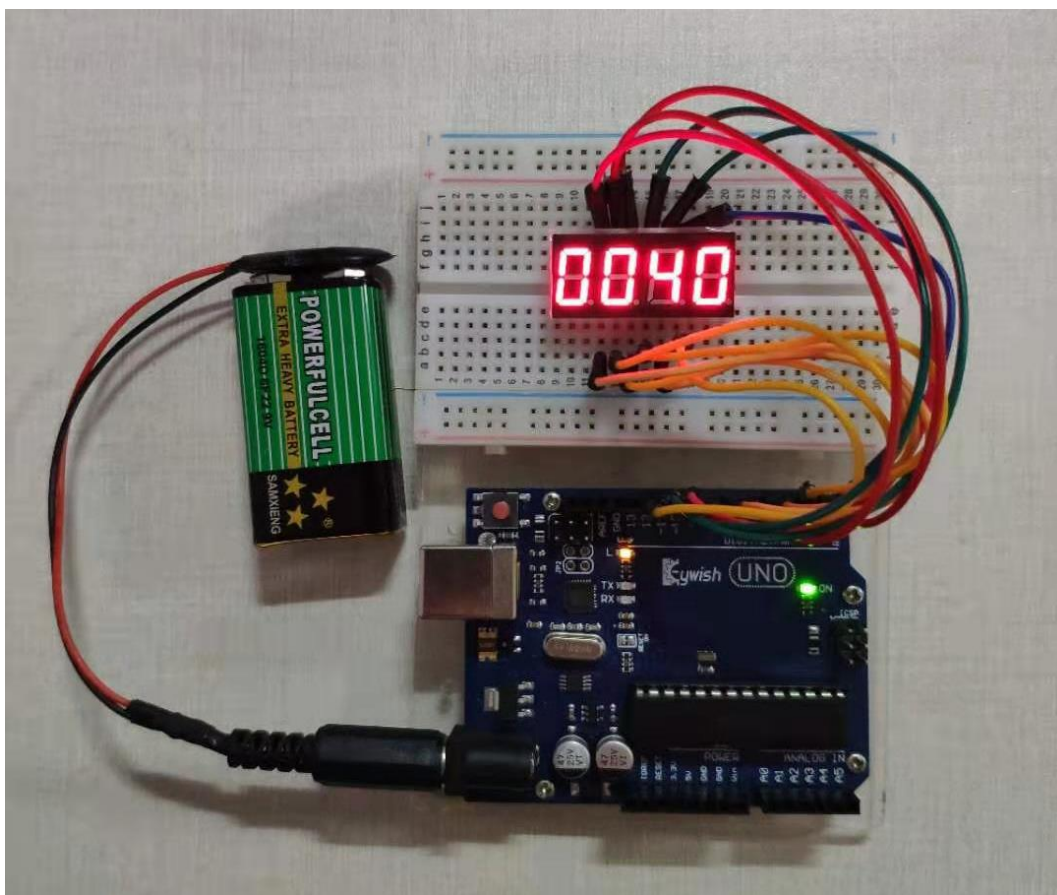
void loop()
{
    if (count++ > 50 )
    {
        ShowTime-- ;
        count = 0 ;
        Serial.println(ShowTime);
    }
    _4Bit_7SegmentDisplay.DisplayChar((int) ShowTime);
    delay(5);
    if (ShowTime == 0) {
        _4Bit_7SegmentDisplay.TurnOffAllLed();
        while(1);
    }
}
```

Примечание: 4 цифры преобразуются из числа number2dis в значение AscII, например, мы хотим преобразовать в «1234», это должно быть следующим

Мы реализовали это в функции библиотеки цифрового дисплея трубки, идея заключается в следующем:

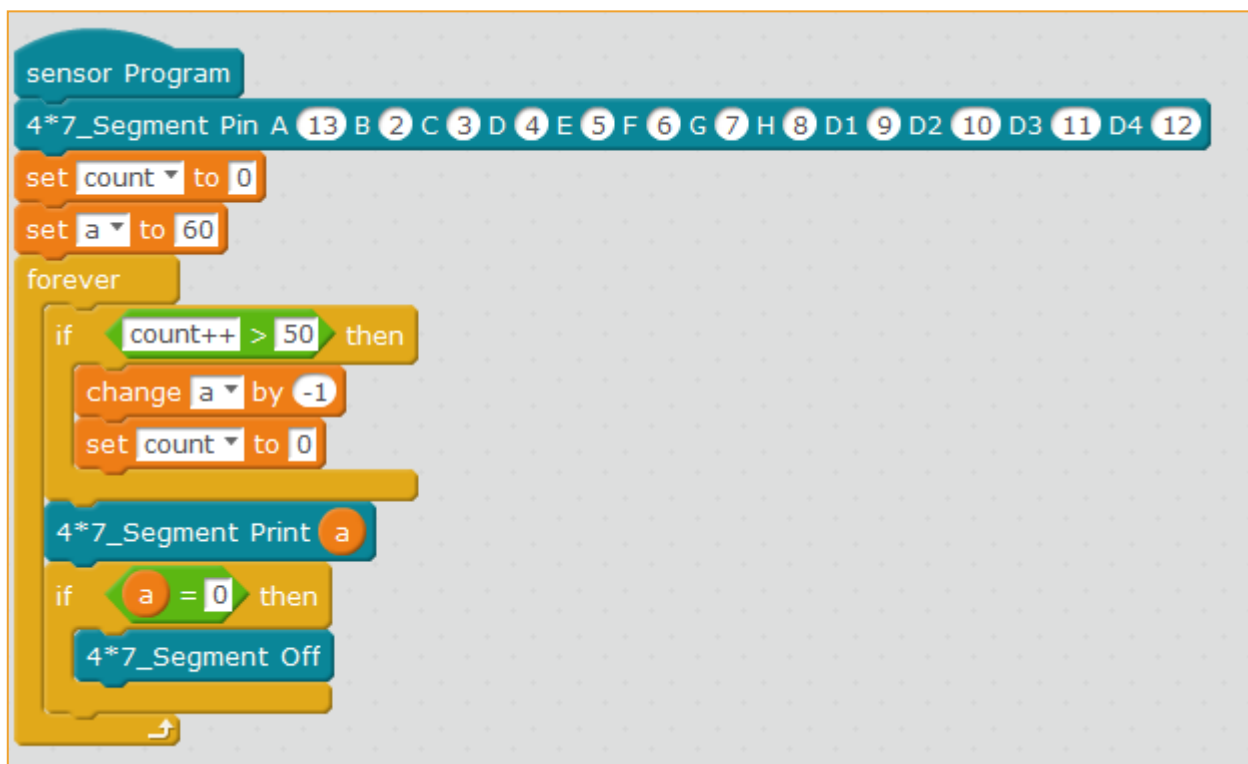
Loop	numble	bit_base	disp
1	1234	1000	1
2	234	100	2
3	34	10	3
4	4	1	4

● Результаты эксперимента



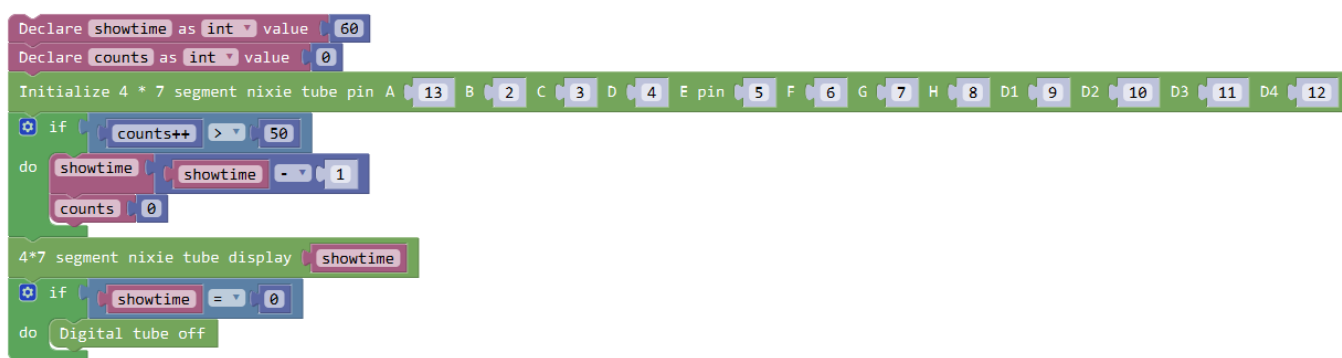
● Программа графического программирования mBlock

mBlock пишет программу 4-разрядной 7-сегментной цифровой трубки, как показано ниже:



● Программа графического программирования Mixly

Mixly пишет программу 4-значной 7-сегментной цифровой трубки, как показано на рисунке ниже:



- Программа графического программирования MagicBlock

: MagicBlock пишет программу цифровой трубки, как показано на рисунке ниже:

