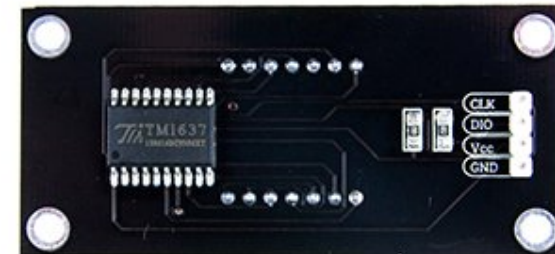
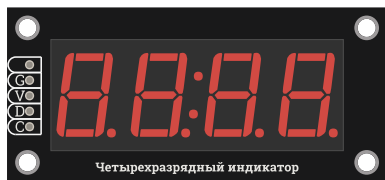


Четырёхразрядный индикатор (Трема-модуль)



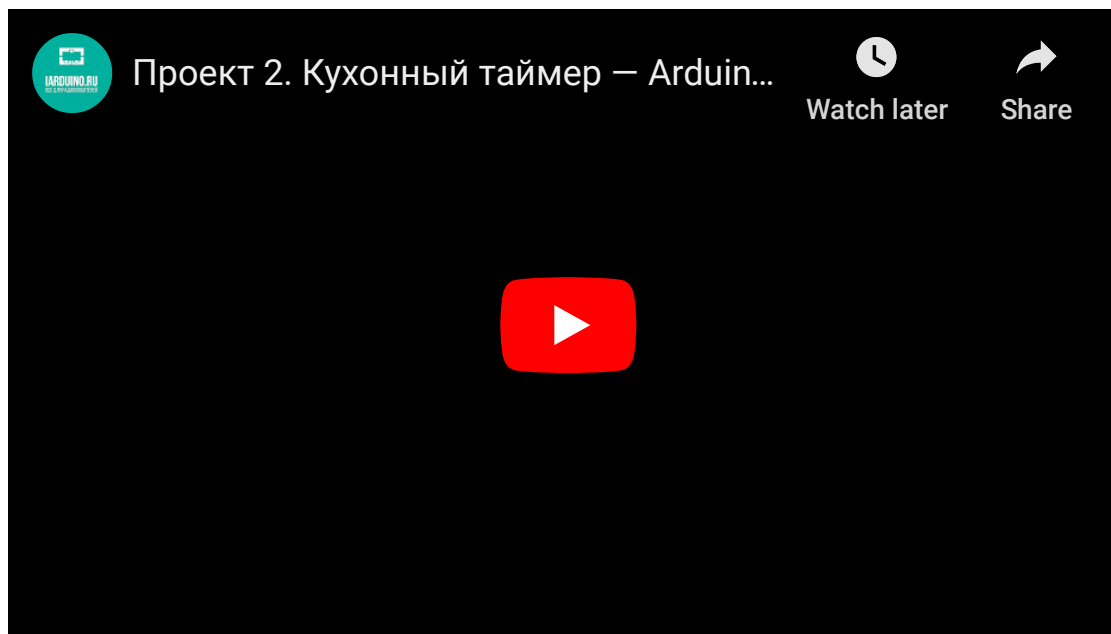
Общие сведения:

[Трема-модуль Четырёхразрядный LED индикатор](#) - позволяет выводить: числа (целые, дробные, отрицательные), время, температуру, текст

и т.д.

Исполнен в линейке Трета-модулей, благодаря этому вам не придется ничего паять и собирать на макетной плате.

Видео:



Спецификация:

- Напряжение питания V_{cc} : 5 В $\pm 10\%$
- Уровень логической «1» на входе: $> 0,7 V_{cc}$
- Уровень логического «0» на входе: $< 0,3 V$
- Потребляемый ток: < 60 мА, при $V_{cc} = 5V$, все сегменты включены на максимальной яркости
- Потребляемый ток: < 200 мкА, при $V_{cc} = 5V$, все сегменты выключены
- Цвет сегментов: красный
- Рабочая температура: $-40 \dots 85$ °С
- Габариты: 65x30x13 мм (без учёта колодки выводов)

- Вес: 20 г

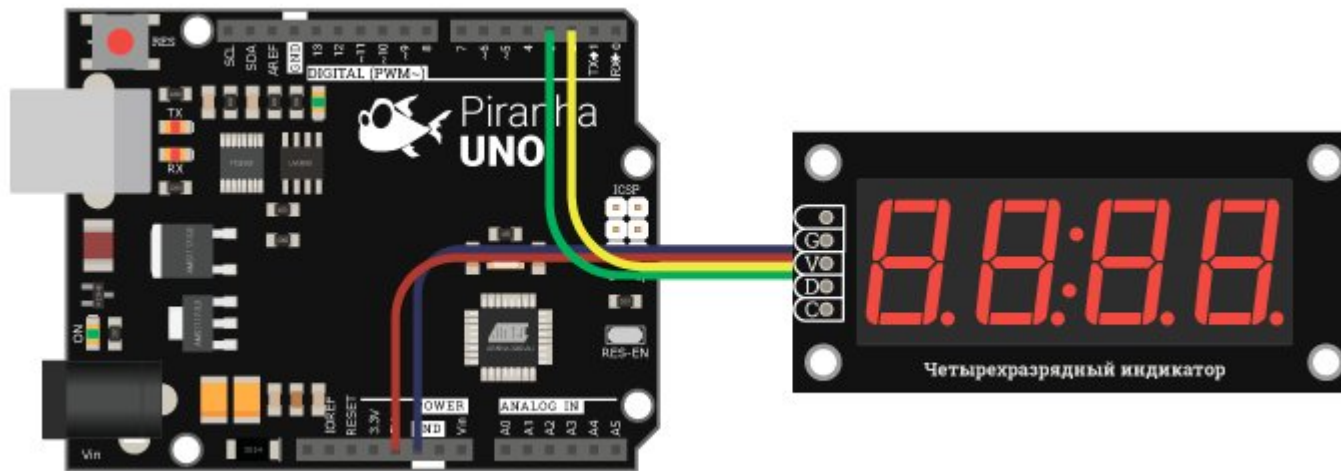
Подключение:

Модуль подключается по двум проводам, к любым цифровым или аналоговым выводам arduino. Номера выводов указываются при объявлении объекта библиотеки: `iarduino_4LED ОБЪЕКТ(CLK , DIO);`

Модуль удобно подключать 3 способами, в зависимости от ситуации:

Способ - 1 : Используя проводной шлейф

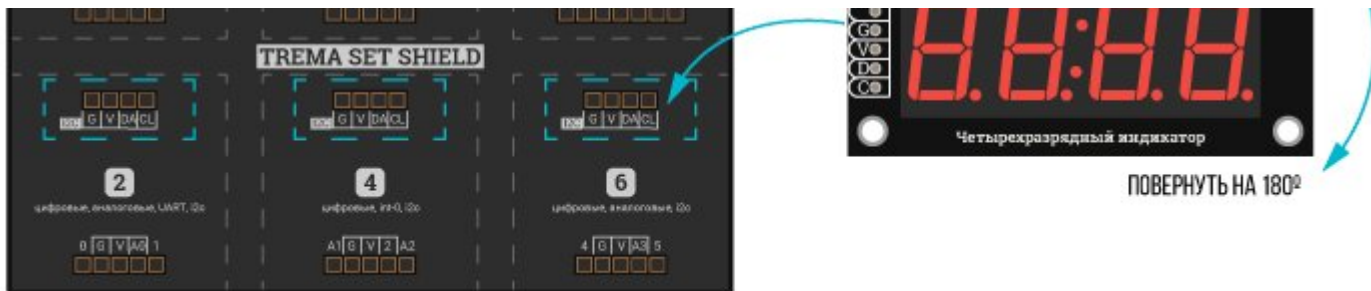
Используя провода «Папа – Мама», подключаем напрямую к контроллеру



Способ - 2 : Используя Trema Set Shield

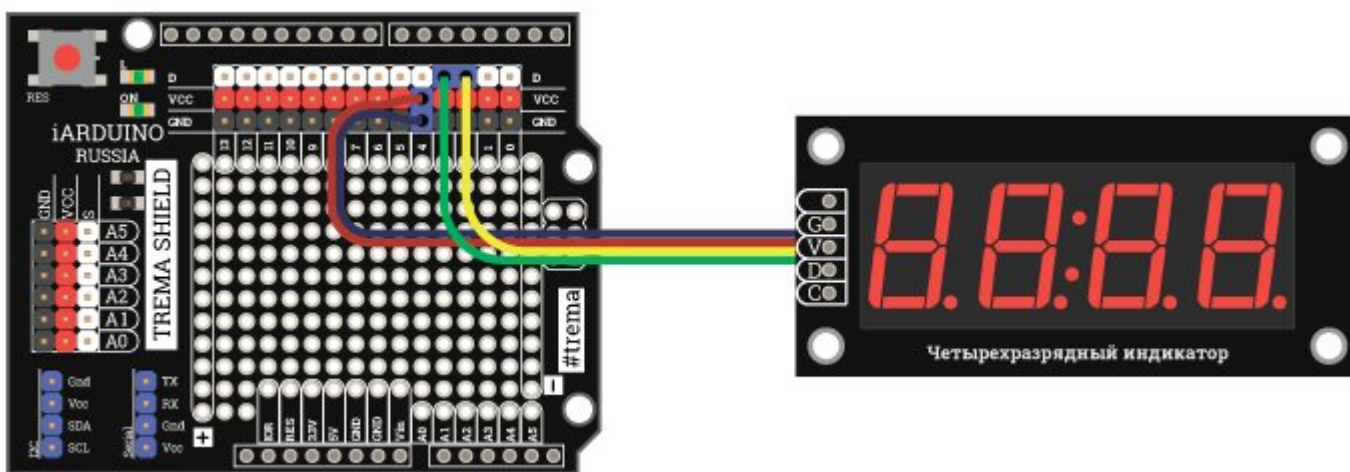
Модуль можно подключить используя Trema Set Shield, в любую ячейку





Способ - 3 : Используя проводной шлейф

Используя два 2-х проводных шлейфа, к Trema Shield, Trema-Power Shield, Motor Shield, Trema Shield NANO и тд.



Подробнее о модуле:

Модуль построен на базе чипа TM1637 и четырехразрядного индикатора. Он позволяет регулировать яркость свечения, выводить числа (целые, дробные, положительные, отрицательные) и символы ("abcdefghijklmnopqrstuvwxyz .,:;*_").

Для работы с модулем, нужно установить библиотеку [iarduino_4LED](#), она позволяет выводить числа, время, температуру и текст. Библиотека работает ТОЛЬКО с [Trema четырёхразрядным LED индикатором](#).

Подробнее про установку библиотеки читайте в нашей [инструкции](#)..

Примеры:

Вывод на индикатор времени, прошедшего с начала старта скетча:

```
#include <iarduino_4LED.h> // подключаем библиотеку iarduino_4LED
iarduino_4LED dispLED(4,5); // объявляем объект для работы с функциями библиотеки iarduino_4LED, с указанием выводов дисплея
float i; // объявляем переменную для хранения числа секунд прошедших с момента старта скетча
void setup(){
    dispLED.begin(); // инициуем LED дисплей
}
void loop(){
    i=(float) millis()/1000; // получаем время в секундах с момента старта скетча в виде числа с плавающей точкой
    dispLED.print( i, 2); // выводим число с двумя знаками после запятой.
}
```

После подачи питания, на дисплее будет отображаться длительность работы Arduino в секундах, с двумя знаками после запятой.

Описание основных функций библиотеки:

Подключение библиотеки:

```
#include <iarduino_4LED.h> // подключаем библиотеку iarduino_4LED
iarduino_4LED dispLED( ВЫВОД_CLK, ВЫВОД_DIO ); // объявляем объект dispLED с указанием выводов подключённых к ( CLK , DIO )
```

Функция begin();

- Назначение: Инициализация индикатора
- Синтаксис: begin();
- Параметры: Нет.
- Возвращаемые значения: Нет.
- Примечание: Необязательная функция (она очищает дисплей и устанавливает среднюю яркость). Вызывается 1 раз в коде setup.

- Пример:

```
dispLED.begin(); // Иницилируем работу с LED индикатором
```

Функция clear();

- Назначение: очистка индикатора
- Синтаксис: clear();
- Параметры: Нет.
- Возвращаемые значения: Нет.
- Пример:

```
dispLED.clear(); // Чистим LED индикатор (все диоды выключатся)
```

Функция light();

- Назначение: Установка яркости свечения индикатора.
- Синтаксис: light (ЧИСЛО);
- Параметры:
 - ЧИСЛО - целое число, определяющее яркость, от 0 (слабое свечение) до 7 (максимальная яркость).
- Возвращаемые значения: Нет.
- Примечание: Если не вызывать функцию begin(), то яркость = 0.
- Пример:

```
dispLED.light(7); // Устанавливаем максимальную яркость свечения LED индикатора
```

Функция point();

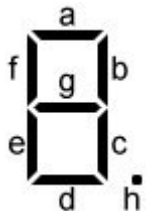
- Назначение: Установка точек.
- Синтаксис: point (ПОЗИЦИЯ , СОСТОЯНИЕ);
- Параметры:

- ПОЗИЦИЯ - целое число, указывающее позицию точки: 1, 2, 3, 4 или 0 для двоеточия.
- СОСТОЯНИЕ - bool, 0 (выключить) или 1 (включить).
- Возвращаемые значения: Нет.
- Примечание:
 - Если первый параметр ПОЗИЦИЯ больше 4, то ЗНАЧЕНИЕ применится ко всем точкам и двоеточию индикатора.
 - Функция управляет только точками и не влияет на ранее установленные цифры.
- Пример:

```
dispLED.point(0, true); // Включаем двоеточие (светится только двоеточие)
dispLED.point(1, true); // Включаем первую точку (светится и двоеточие и самая левая точка)
dispLED.point(0, false); // Выключаем двоеточие (светится только самая левая точка)
```

Функция setLED();

- Назначение: Установка светодиодов (сегментов) индикатора по битам.
- Синтаксис: setLED ([[[[[БАЙТ_№1] , БАЙТ_№2] , БАЙТ_№3] , БАЙТ_№4] , ФЛАГ]);
- Параметры:
 - БАЙТ_№1 - каждый бит этого байта включает свой светодиод (сегмент) в 1 разряде.
 - БАЙТ_№2 - каждый бит этого байта включает свой светодиод (сегмент) в 2 разряде.
 - БАЙТ_№3 - каждый бит этого байта включает свой светодиод (сегмент) в 3 разряде.
 - БАЙТ_№4 - каждый бит этого байта включает свой светодиод (сегмент) в 4 разряде.
 - ФЛАГ - включает двоеточие.
- Возвращаемые значения: Нет.
- Примечание:
 - Каждый бит байта соответствует одному светодиоду (сегменту) в следующем порядке: **hgfedcba** (старший бит управляет сегментом «h», а младший бит управляет сегментом «a»).
 - Все параметры являются необязательными, отсутствие параметра означает что все светодиоды (сегменты) в данном разряде будут выключены.
- Пример:



```
dispLED.setLED(В00000110, В01100011); // В 1 разряде включены сегменты «с» и «b» (как у цифры 1), а во 2 разряде включены сег
```

Функция print();

- Назначение: Вывод числа, массива чисел, текста, времени или температуры.
- Синтаксис: `print (ЗНАЧЕНИЕ , ПАРАМЕТРЫ_ВЫВОДА_ЧИСЛА);`
- Параметры:
 - ЗНАЧЕНИЕ - то, что требуется вывести на индикатор, это может быть:
 - Целое число (как положительное, так и отрицательное).
 - Дробное число (как положительное, так и отрицательное).
 - Массив (из 4 положительных целых чисел, от 0 до 9)
 - Текст (из следующих символов "0123456789 abcdefghijklmnopstu .,:;*_")
 - ПАРАМЕТРЫ_ВЫВОДА_ЧИСЛА - допускается указывать от 0 до 5 параметров:
 - LEN1, LEN2, LEN3 или LEN4 - количество символов в выводимом числе (включая знак минус).
 - POS1, POS2, POS3 или POS4 - позиция от левого края, к которой привязывается выводимое число.
 - LEFT или RIGHT - направление вывода относительно указанной позиции.
 - DEC или HEX - система счисления для выводимого числа.
 - TIME или TEMP - отображение чисел в виде времени или числа в виде температуры.
 - символ типа char - для заполнения им недостающих разрядов
- Возвращаемые значения: Нет.
- Примечание:
 - Параметры указываются только если значением является число (целое или с плавающей точкой).
 - Если первым параметром (после значения) указать число, то оно будет означать количество выводимых разрядов после запятой.
- Примеры:

```
dispLED.print( 12345.6789 , 2 , POS1 , LEN3 , RIGHT ); // число 12345.6789 будет выведено как 5.67
```

Вывод без форматирования:


```
dispLED.print( 1 ); // Вывод целого числа 1.
```



```
dispLED.print( 3.45 ); // Вывод дробного числа 3.45 (по умолчанию выводится 1 знак после запятой)
```



```
dispLED.print( -12 ); // Вывод отрицательного целого числа -12.
```



```
dispLED.print( -3.45 ); // Вывод отрицательного дробного числа -3.45
```



```
dispLED.print( 250 , HEX ); // Вывод числа 250 в шестнадцатеричной системе счисления
```



```
int a[4]={0,3,6,9}; dispLED.print( a ); // Вывод массива из 4 положительных цифр от 0 до 9
```



```
dispLED.print( 23 , TEMP ); // Вывод температуры
```



```
dispLED.print( 23.6 , TEMP ); // Вывод температуры
```



```
dispLED.print( 4 , 5 , TIME ); // Вывод времени
```



```
dispLED.print( "67" ); // Вывод текста
```



```
dispLED.print( "OFF" ); // Вывод текста из букв латинского алфавита
```

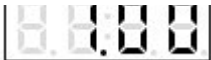





```
dispLED.print( "8:9" ); // Вывод текста с двоеточием
```









Вывод чисел с указанием количества знаков после запятой:

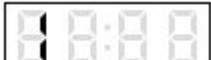







<code>dispLED.print(1 , 2);</code> // Вывод целого числа 1 с 2 знаками после запятой	
<code>dispLED.print(-2 , 1);</code> // Вывод отрицательного целого числа -2 с 1 знаком после запятой	
<code>dispLED.print(3.45 , 2);</code> // Вывод дробного числа 3.45 с 2 знаками после запятой	
<code>dispLED.print(-3.45 , 0);</code> // Вывод отрицательного дробного числа -3.45 без знаков после запятой	

Вывод чисел с указанием их размерности:









<code>dispLED.print(1 , LEN2);</code> // Вывод целого числа 1 в 2 разрядах	
<code>dispLED.print(-1 , LEN3);</code> // Вывод отрицательного целого числа -1 в 3 разрядах	
<code>dispLED.print(2.3 , LEN3);</code> // Вывод дробного числа 2.3 в 3 разрядах	
<code>dispLED.print(-4.567 , 1 , LEN4);</code> // Вывод дробного числа -4.567 в 4 разрядах , с 1 знаком после запятой	
<code>dispLED.print(8901 , LEN1);</code> // Вывод целого числа 8901 в 1 разряде	
<code>dispLED.print(7 , LEN3 , ' ');</code> // Вывод числа 7 в 3 разрядах, с заполнением пустых разрядов символом ' '	

Указание направления сдвига чисел:

<code>dispLED.print(1 , RIGHT);</code> // Вывод целого числа 1 со сдвигом вправо от старшего разряда	
<code>dispLED.print(12 , RIGHT);</code> // Вывод целого числа 12 со сдвигом вправо от старшего разряда	

<code>dispLED.print(12.3 , RIGHT);</code> // Вывод дробного числа 12.3 со сдвигом вправо от старшего разряда	
<code>dispLED.print(1 , LEFT);</code> // Вывод целого числа 1 со сдвигом влево (по умолчанию) от младшего разряда	
<code>dispLED.print(12 , LEFT);</code> // Вывод целого числа 12 со сдвигом влево (по умолчанию) от младшего разряда	
<code>dispLED.print(12.3 , LEFT);</code> // Вывод дробного числа 12.3 со сдвигом влево (по умолчанию) от младшего разряда	

Вывод чисел с привязкой к определённой позиции от левого края:

<code>dispLED.print(74 , POS4);</code> // Вывод целого числа 74 со сдвигом влево (по умолчанию) от 4 позиции	
<code>dispLED.print(74 , POS3);</code> // Вывод целого числа 74 со сдвигом влево (по умолчанию) от 3 позиции	
<code>dispLED.print(74 , POS2);</code> // Вывод целого числа 74 со сдвигом влево (по умолчанию) от 2 позиции	
<code>dispLED.print(74 , POS1);</code> // Вывод целого числа 74 со сдвигом влево (по умолчанию) от 1 позиции	
<code>dispLED.print(74 , POS4 , RIGHT);</code> // Вывод целого числа 74 со сдвигом вправо от 4 позиции	
<code>dispLED.print(74 , POS3 , RIGHT);</code> // Вывод целого числа 74 со сдвигом вправо от 3 позиции	
<code>dispLED.print(74 , POS2 , RIGHT);</code> // Вывод целого числа 74 со сдвигом вправо от 2 позиции	
<code>dispLED.print(74 , POS1 , RIGHT);</code> // Вывод целого числа 74 со сдвигом вправо от 1 позиции	

Параметры вывода числа можно комбинировать в любой последовательности:

за исключением числа, указывающего количество знаков после запятой, оно должно быть первым после выводимого значения

```
dispLED.print( 12345.6789 , 2 , POS1 , LEN3 , RIGHT );  
// Вывод дробного числа 12345.6789  
// выводить 2 знака после запятой  
// POS1 - число вывести на дисплей начиная с 1 позиции от левого края  
// LEN3 - размерность выводимого числа 3 разряда  
// RIGHT - число сдвигать вправо от указанной позиции.
```

A digital display showing the number 5.678. The digits are white on a black background, with a colon between the decimal point and the first digit after it.

Применение:

- Вывод небольших чисел: время, температура, скорость, позиция и т.д.

Ссылки:

- Библиотека [iarduino_4LED](#).
- [Wiki - Установка библиотек в Arduino IDE](#).