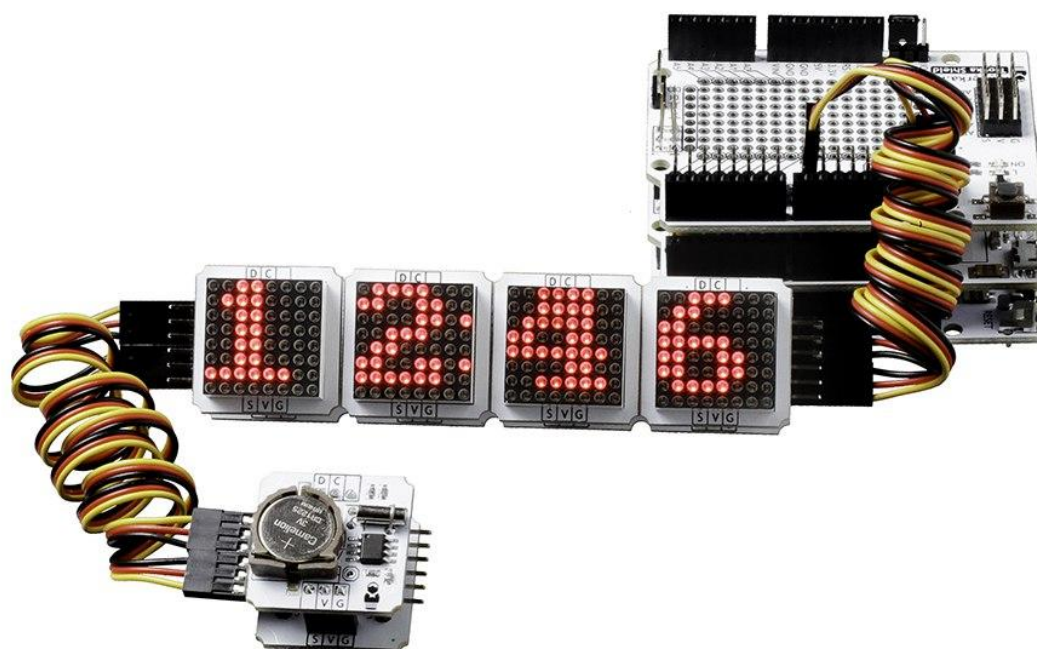


Светодиодная матрица 8×8 (Тройка-модуль)

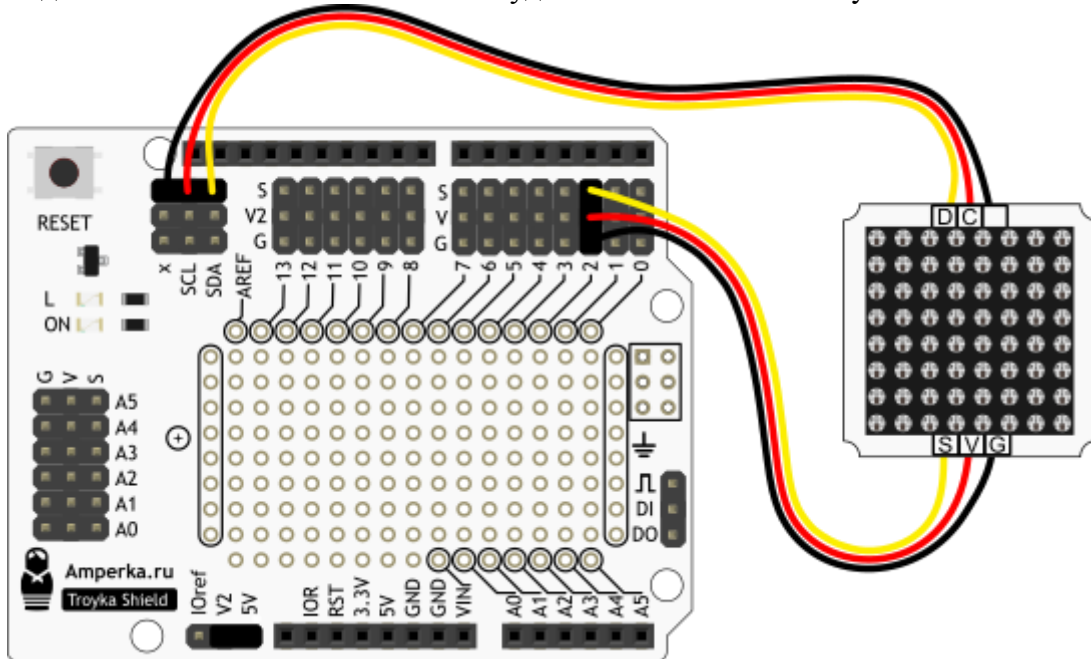
Тройка Led Matrix - цепочка из шестидесяти четырёх красных светодиодов, собранная на одноюнитовом модуле.



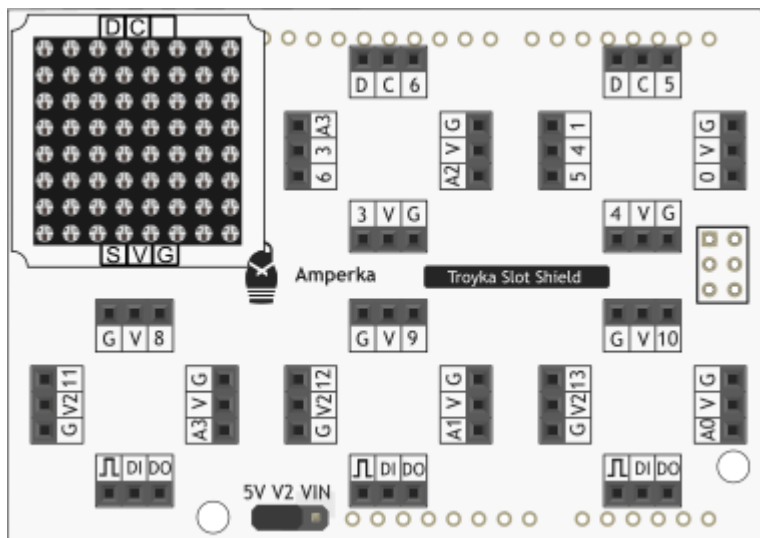
Обычно подключение светодиодных матриц ассоциируется кропотливой работой и острой нехваткой пинов на платах. Мы добавили драйвер управления матрицей. Теперь модуль управляется по скоростному интерфейсу I²C и занимает всего два пина микроконтроллера.

Подключение и настройка

Светодиодная матрица общается с управляющей электроникой по I²C/TWI. При подключении к Arduino или Iskra JS удобно использовать Troyka Shield.



С Troyka Slot Shield можно обойтись без лишних проводов.



Вывод случайных пикселей

Мы будем поочерёдно добавлять и удалять случайные пиксели. В итоге получится довольно забавная мерцающая картина, которая пригодится для создания «инопланетных» интерфейсов в декорациях квестов.

[drawPixels.ino](https://github.com/Amperka/drawPixels.ino)

```
// библиотека для работы I2C  
#include "Wire.h"  
// библиотека для работы со светодиодной матрицей  
#include "TroykaLedMatrix.h"
```

```

// создаём объект matrix для работы с матрицей
TroykaLedMatrix matrix;

void setup()
{
  // начало работы с матрицей
  matrix.begin();
  // очищаем матрицу
  matrix.clear();
}

void loop()
{
  // заполняем матрицу случайным пикселем
  matrix.drawPixel(random(8), random(8));
  // очищаем на матрице случайный пиксель
  matrix.clearPixel(random(8), random(8));
  // ждём 50 мс
  delay(50);
}

```

Вывод изображений

Методы библиотеки позволяют выводить изображение форматом 8×8. В качестве примера выведем сердце на матрицу.

[drawBitmap.ino](#)

```

// библиотека для работы I²C
#include "Wire.h"
// библиотека для работы со светодиодной матрицей
#include "TroykaLedMatrix.h"

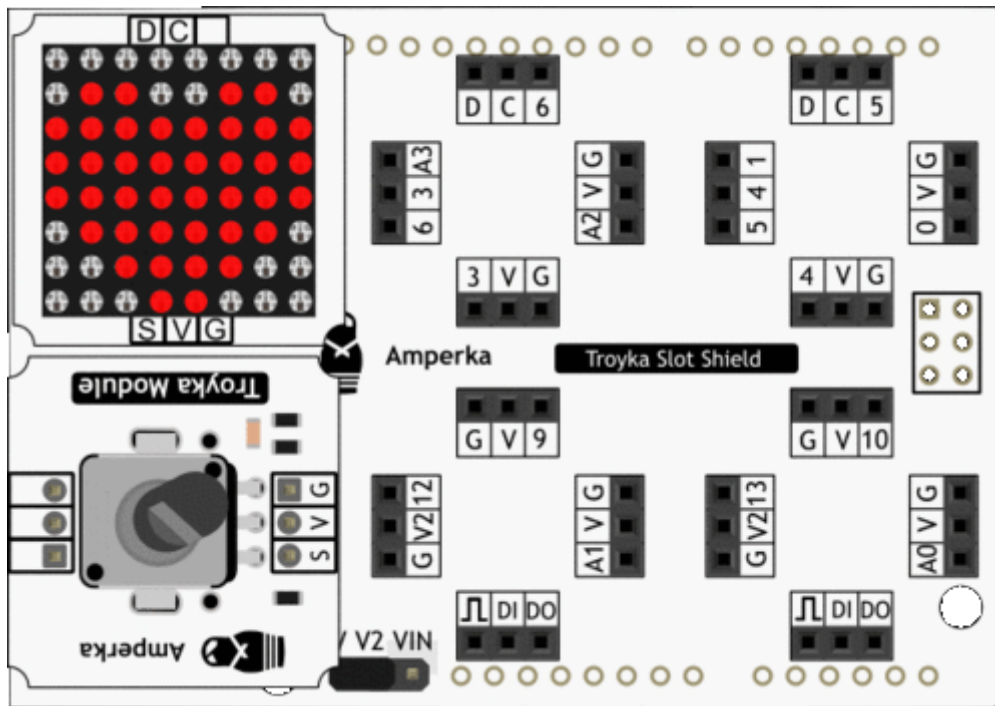
// создаём объект matrix для работы с матрицей
TroykaLedMatrix matrix;

// массив картнки сердца
const uint8_t heart[] {
  0b00000000,
  0b01100110,
  0b11111111,
  0b11111111,
  0b11111111,
  0b01111110,
  0b00111100,
  0b00011000,
};

void setup()
{
  // начало работы с матрицей
  matrix.begin();
  // очищаем матрицу
  matrix.clear();
  // отображаем на матрице сердце
  matrix.drawBitmap(heart);
}

void loop()
{
}

```



Регулировка яркости матрицы

Контроллер матрицы позволяет регулировать яркость светодиодной матрицы. В качестве примера будем изменять яркость свечения в зависимости от положения потенциометра. Для этого добавьте к проекту Потенциометр (Тройка-модуль)

[drawBitmapBrightness.ino](#)

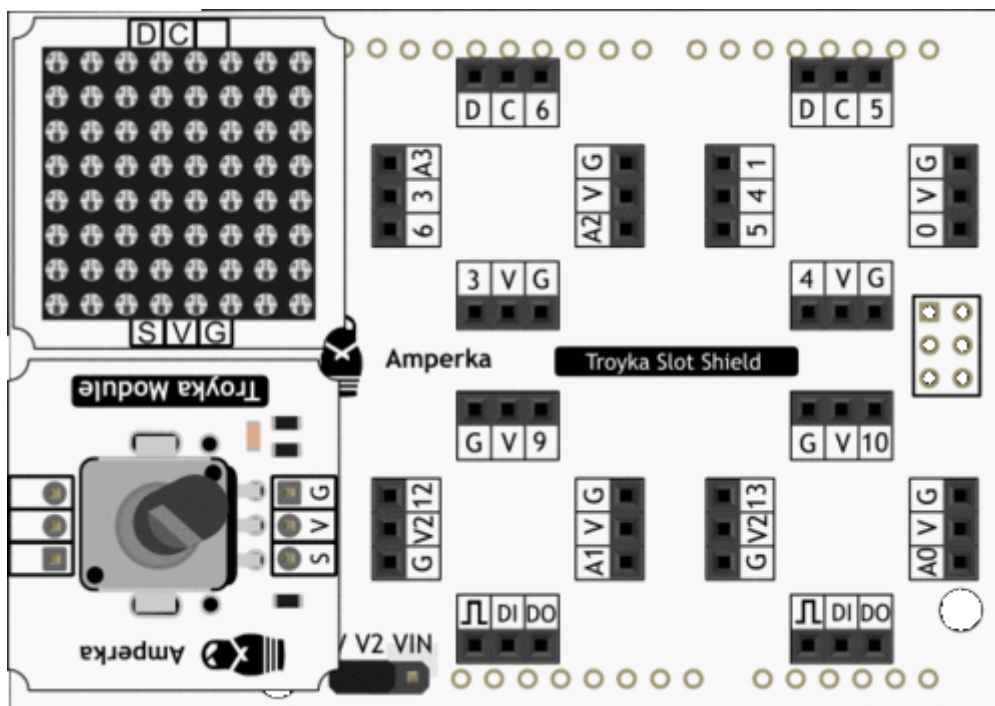
```
// библиотека для работы I2C
#include "Wire.h"
// библиотека для работы со светодиодной матрицей
#include "TrojkaLedMatrix.h"

// создаём объект matrix для работы с матрицей
TrojkaLedMatrix matrix;

// массив картнки сердца
const uint8_t heart[] {
0b00000000,
0b01100110,
0b11111111,
0b11111111,
0b11111111,
0b01111110,
0b00111100,
0b00011000,
};

void setup()
{
  // начало работы с матрицей
  matrix.begin();
  // очищаем матрицу
  matrix.clear();
  // отображаем на матрице сердце
  matrix.drawBitmap(heart);
}
}
```

```
void loop()  
{  
  // считываем значение с потенциометра  
  int sensorValue = analogRead(A3);  
  // в зависимости от значения регулятора  
  // устанавливаем яркость светодиодной матрицы  
  if (sensorValue < 100) {  
    matrix.setCurrentLimit(ROW_CURRENT_05MA);  
  } else if (sensorValue >= 100 && sensorValue < 200) {  
    matrix.setCurrentLimit(ROW_CURRENT_10MA);  
  } else if (sensorValue >= 200 && sensorValue < 300) {  
    matrix.setCurrentLimit(ROW_CURRENT_15MA);  
  } else if (sensorValue >= 300 && sensorValue < 400) {  
    matrix.setCurrentLimit(ROW_CURRENT_20MA);  
  } else if (sensorValue >= 400 && sensorValue < 500) {  
    matrix.setCurrentLimit(ROW_CURRENT_25MA);  
  } else if (sensorValue >= 500 && sensorValue < 600) {  
    matrix.setCurrentLimit(ROW_CURRENT_30MA);  
  } else if (sensorValue >= 600 && sensorValue < 700) {  
    matrix.setCurrentLimit(ROW_CURRENT_35MA);  
  } else if (sensorValue >= 700 && sensorValue < 800) {  
    matrix.setCurrentLimit(ROW_CURRENT_40MA);  
  } else if (sensorValue >= 800 && sensorValue < 900) {  
    matrix.setCurrentLimit(ROW_CURRENT_45MA);  
  } else if (sensorValue >= 900 && sensorValue < 1000) {  
    matrix.setCurrentLimit(ROW_CURRENT_45MA);  
  } else {  
    matrix.setCurrentLimit(ROW_CURRENT_50MA);  
  }  
}
```



Вывод СИМВОЛОВ

Если подключить встроенный в библиотеку шрифт, возможно выводить любой символ из таблицы ASCII. В качестве примера выведем по символно слово «Амперка».

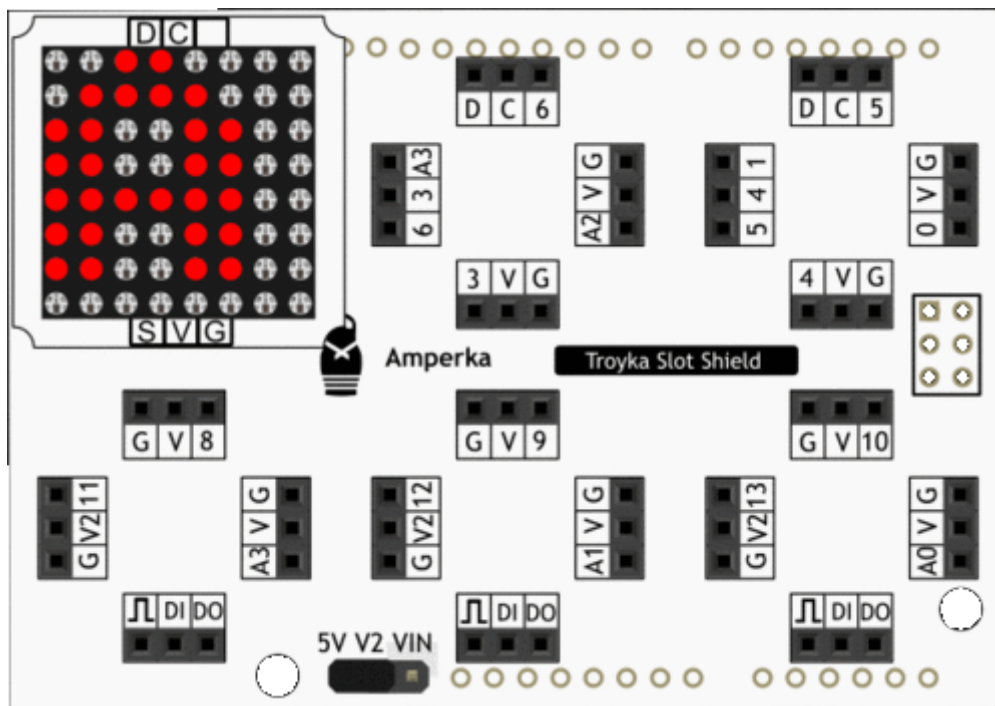
[printSymbolsAmperka.ino](#)

```
// библиотека для работы I2C
#include "Wire.h"
// библиотека для работы со светодиодной матрицей
#include "ТройкаLedMatrix.h"

// создаём объект matrix для работы с матрицей
ТройкаLedMatrix matrix;

void setup()
{
  // начало работы с матрицей
  matrix.begin();
  // очищаем матрицу
  matrix.clear();
  // устанавливаем шрифт
  matrix.selectFont(FONT_8X8_BASIC);
}

void loop()
{
  // выводим поочерёдно слово «Амперка»
  matrix.drawSymbol('A');
  delay(1000);
  matrix.drawSymbol('M');
  delay(1000);
  matrix.drawSymbol('P');
  delay(1000);
  matrix.drawSymbol('E');
  delay(1000);
  matrix.drawSymbol('R');
  delay(1000);
  matrix.drawSymbol('K');
  delay(1000);
  matrix.drawSymbol('A');
  delay(1000);
  // очищаем матрицу
  matrix.clear();
  delay(1000);
}
```



Строка из матриц

Одна матрица хорошо, а три ещё лучше. Драйвер управления светодиодной матрицей имеет функцию выбора адреса с возможностью подключения до четырёх матриц. Установите ещё две матрицы на Troyka Slot Shield, заранее модернизировав их для работы с разными адресами. Следующий код будет отображать на трёх матрицах могучий язык «C++».

lotOfMatrix.ino

```
// библиотека для работы I2C
#include "Wire.h"
// библиотека для работы со светодиодной матрицей
#include "TroykaLedMatrix.h"

// создаём объекты matrix для работы с матрицами
// для каждой матрицы передаём свой адрес
// подробнее читайте на:
// http://wiki.amperka.ru/продукты:troyka-led-matrix
TroykaLedMatrix matrix1(0x60);
TroykaLedMatrix matrix2(0x61);
TroykaLedMatrix matrix3(0x62);

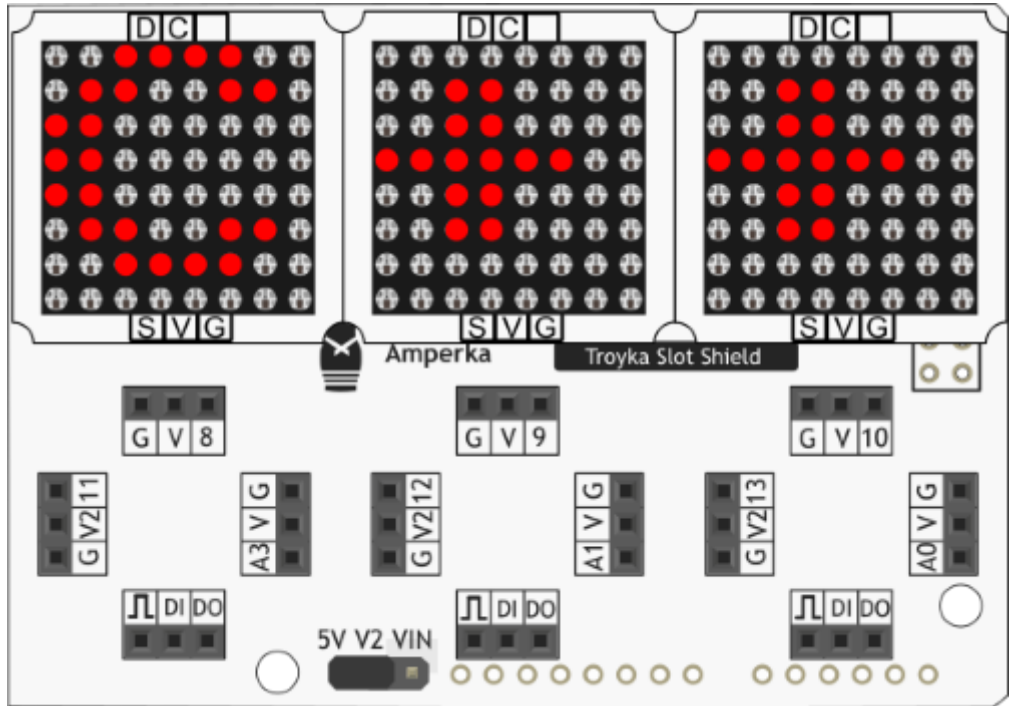
void setup()
{
  // начало работы с матрицами
  matrix1.begin();
  matrix2.begin();
  matrix3.begin();
  // очищаем матрицы
  matrix1.clear();
  matrix2.clear();
  matrix3.clear();
  // устанавливаем шрифт
  matrix1.selectFont(FONT_8X8_BASIC);
  matrix2.selectFont(FONT_8X8_BASIC);
  matrix3.selectFont(FONT_8X8_BASIC);
}
```



```

// выводим на каждую матрицу по символу из строки «C++»
matrix1.drawSymbol('C');
matrix2.drawSymbol('+');
matrix3.drawSymbol('+');
}
void loop()
{
}
}

```

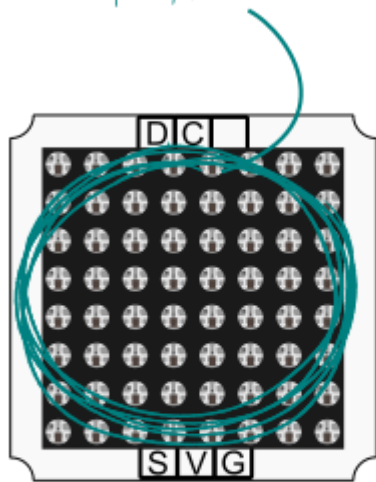


Элементы платы

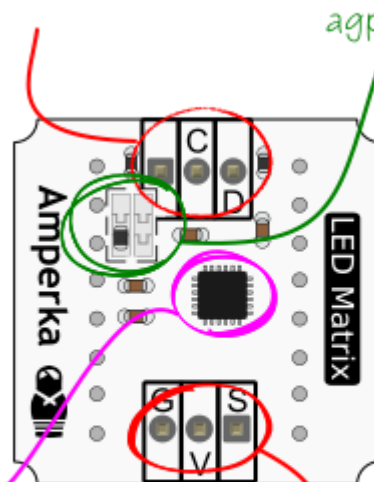
Светодиодная матрица

Тройка-контакты шины I²C

Площадки выбора адреса



Драйвер светодиодной матрицы

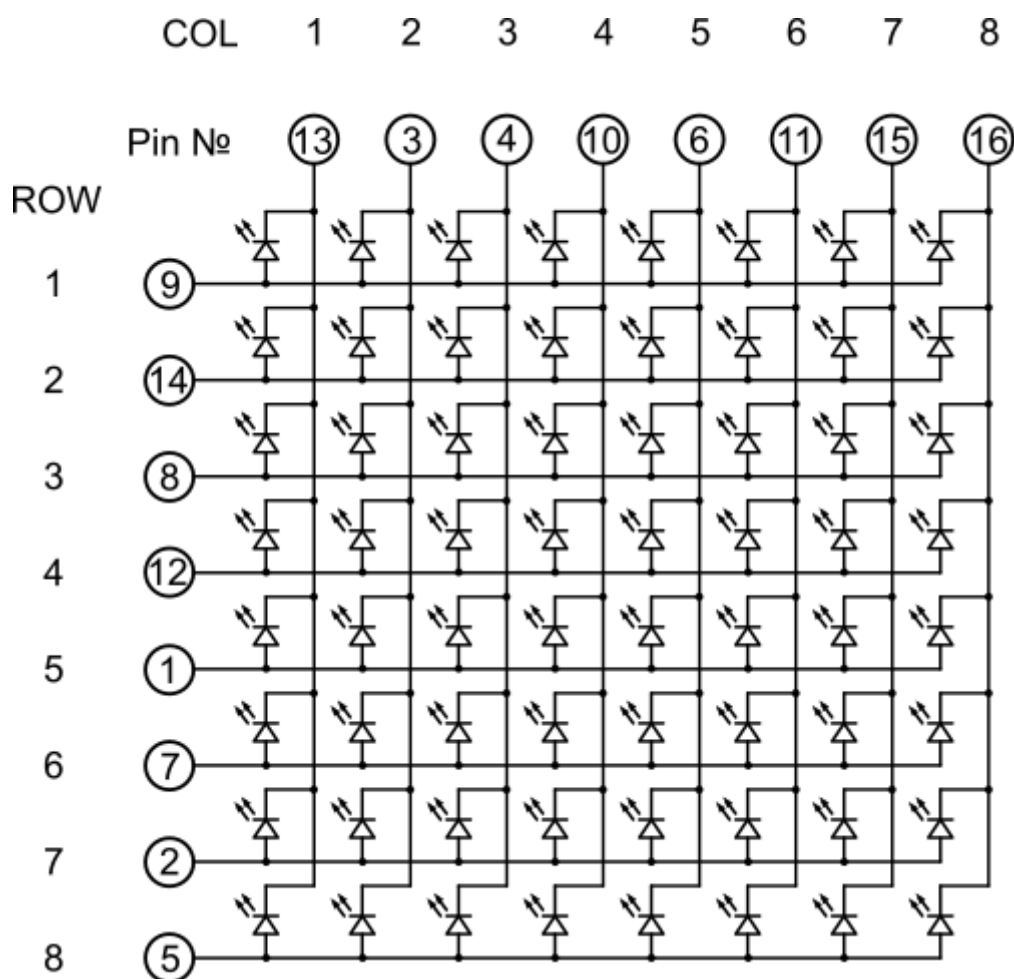


Тройка-контакты

Светодиодная матрица

Светодиодная матрица 788BS — это панель из 64 красных светодиодов.

Принципиальная схема

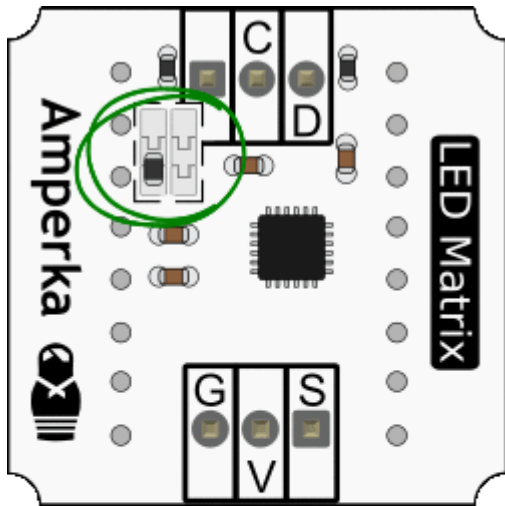


Драйвер светодиодной матрицы

Драйвер IS31FL3728 представляет из себя контроллер светодиодной матрицы 8×8. Общение с управляющей платой осуществляется по протоколу I²C. Контроллер в разы облегчает работу с матрицей и уменьшает количество занятых пинов до двух: SDA и SCL. Приятным бонусом является выбор адреса контроллера, благодаря чему есть возможность подключить до четырёх матриц к управляющей плате по двум проводам.

Площадки выбора адреса

Соедините капелькой припоя контакты специальных перемычек на модуле. Это позволит использовать до 4 матриц в одном проекте.



I²C адрес:
0x60

Контакты подключения трёхпроводных шлейфов

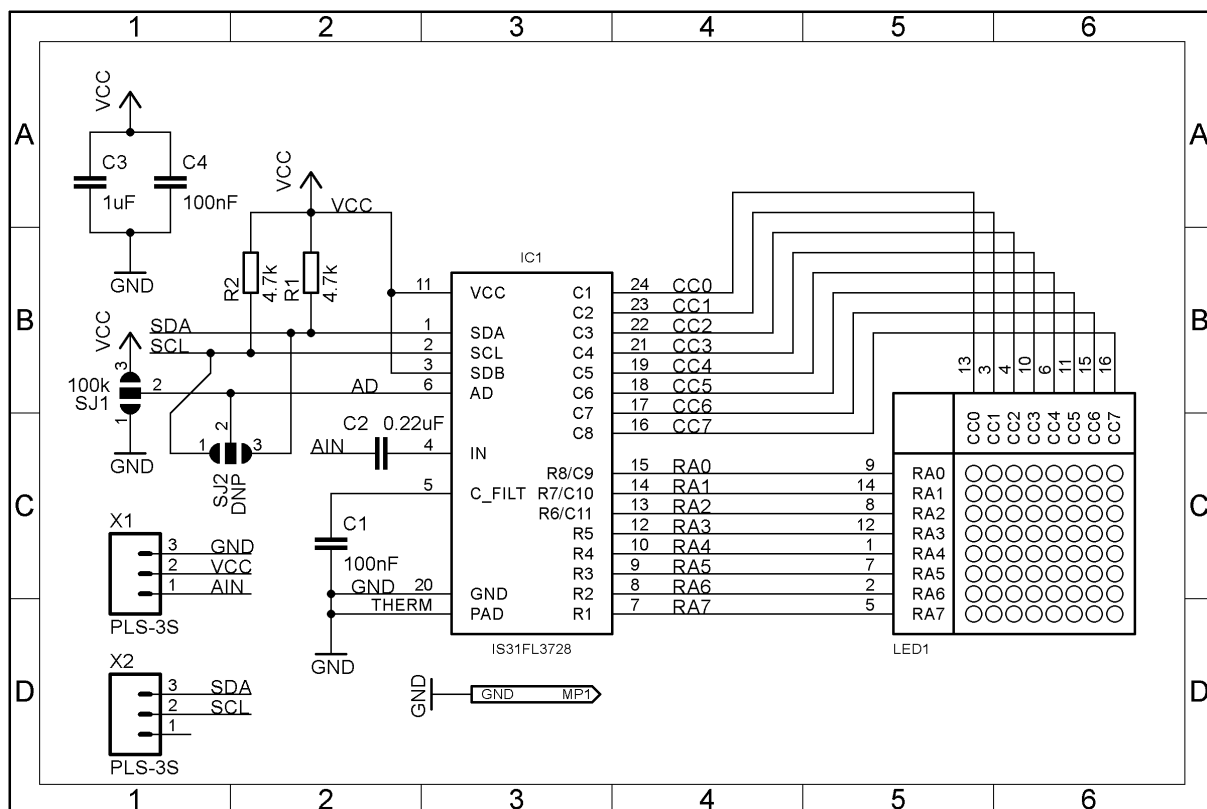
Группа питания и сигнальный

- Сигнальный пин (S) — входной пин аудио сигнала. Используется только в режиме эквалайзера. При обычной работе с модулем не используется.
- Питание (V) — соедините с питанием микроконтроллера
- Земля (G) — соедините с землёй микроконтроллера

Шина I2C

- Сигнальный (D) — подключите к пину *SDA* микроконтроллера.
- Сигнальный (C) — подключите к пину *SCL* микроконтроллера.

Принципиальная и монтажная схемы



Характеристики

- Контроллер матрицы: IS31FL3728
- Модуль матрицы: светодиодная матрица 788BS
- Количество светодиодов: 64
- Рабочее напряжение: 3,3–5 В
- Потребляемый ток: до 2 А
- Габариты: 25,4×25,4 мм

Ресурсы

- [Векторное изображение светодиодной матрицы \(Top\)](#)
- [Векторное изображение светодиодной матрицы \(Bottom\)](#)
- [Библиотека для Arduino](#)