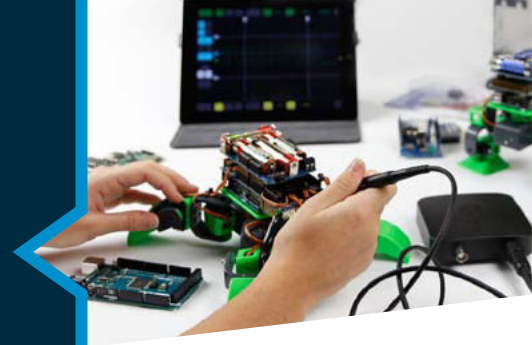




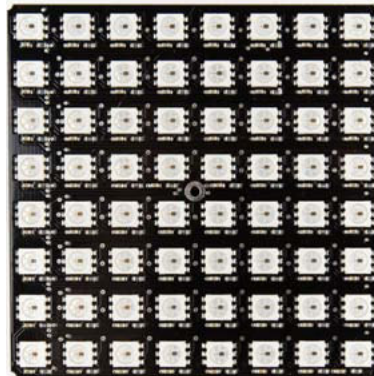
USER MANUAL

64 LED RGB MATRIX



Inhoudsopgave

De 64 LED RGB-matrix voorbereiden	3
Datalijnen	3
Voeding	6
De 64 LED RGB-matrix aansturen	8
Welke microcontroller gebruiken?	8
Een paneel op een Arduino™ Uno aansluiten	8
Adafruit™ NeoPixel-library	9
Adafruit™ NeoMatrix-library	13
De 64 LED RGB-matrix monteren	17
Montagegaten	17
3D-montagebeugels	18



Aan de slag!

De 64 LED RGB-matrix voorbereiden

Vooraleer we dieper ingaan op de werking van het paneel, leggen we eerst uit hoe u het paneel correct moet voorbereiden.

Datalijnen

Alvorens de voeding aan te sluiten, is het belangrijk om te weten hoe de WS2812-leds met elkaar verbonden zijn. Elke led functioneert als een schuifregister, deze schuift de gegevens een positie op vanaf de ingang naar de uitgang en gaat dan naar de volgende led. Dit is de manier waarop de displaygegevens via een paneel worden doorgezonden, van een led naar de volgende. In de VM207 zijn de leds verbonden in rijen. Elke rij heeft een terugkerende datalijn, zodat het beginpunt van de volgende rij verbonden is met het uiteinde van de vorige rij. Het voordeel van deze functie is dat de rijen niet zigzaggen en u dus meerdere panelen kunt koppelen.

Raadpleeg zorgvuldig de onderstaande afbeelding, om te begrijpen hoe de leds zijn aangesloten.

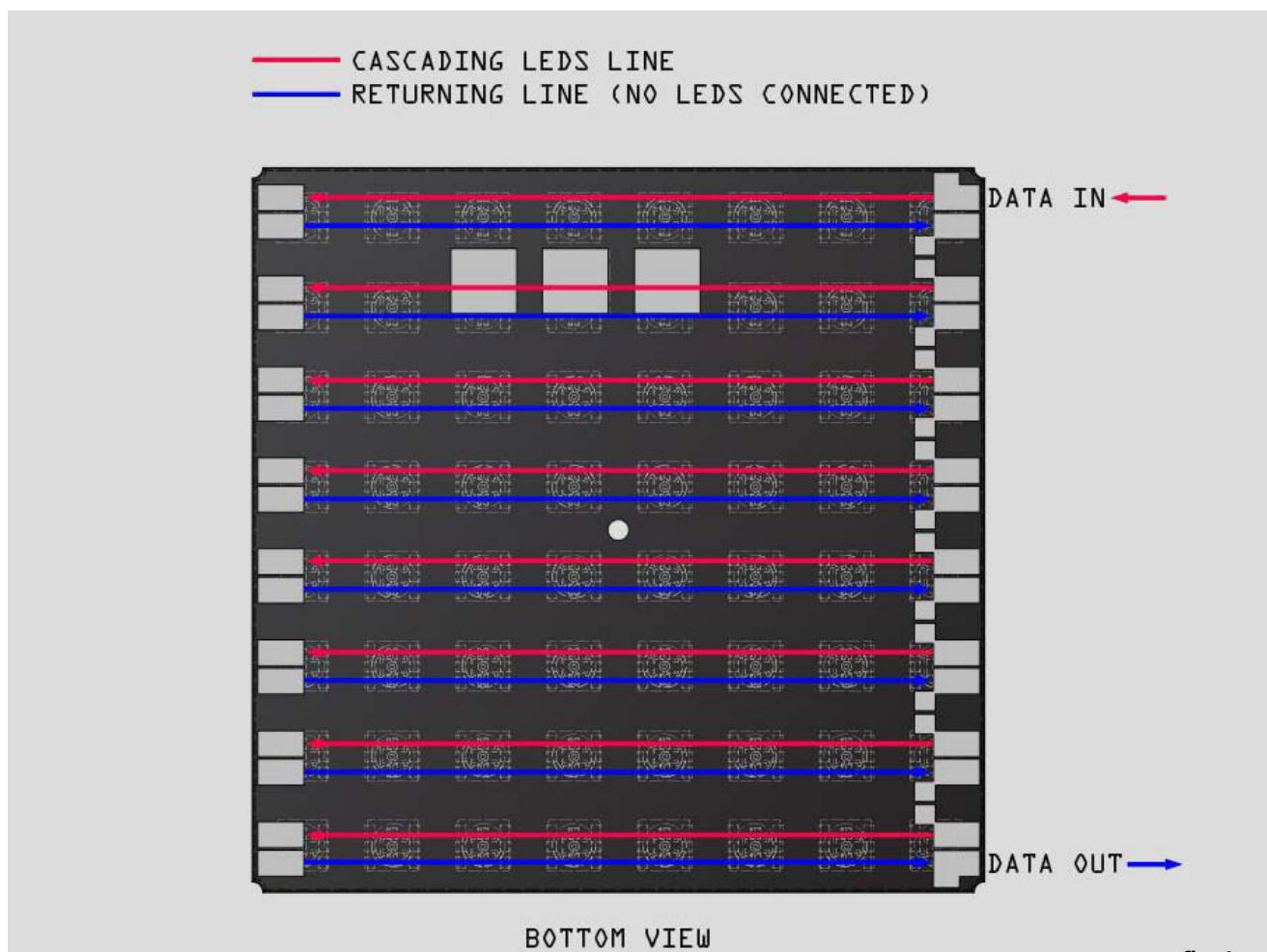


fig. 1

U ziet dat het paneel niet onmiddellijk kan functioneren omdat de eerste rij abrupt stopt en nergens op aangesloten is. De displaygegevens stoppen aan het uiteinde van de eerste rij leds en worden niet via het paneel doorgezonden. Om het paneel als 8 x 8 display te gebruiken en de displaygegevens naar elke led door te sturen, moet u de pads (gele soldeeraansluitingen) verbinden (zie afb.). Gebruik een kleine hoeveelheid soldeer om deze pads vast te maken.

Alle rijen zijn met elkaar verbonden en vormen een reeks WS2812-leds.

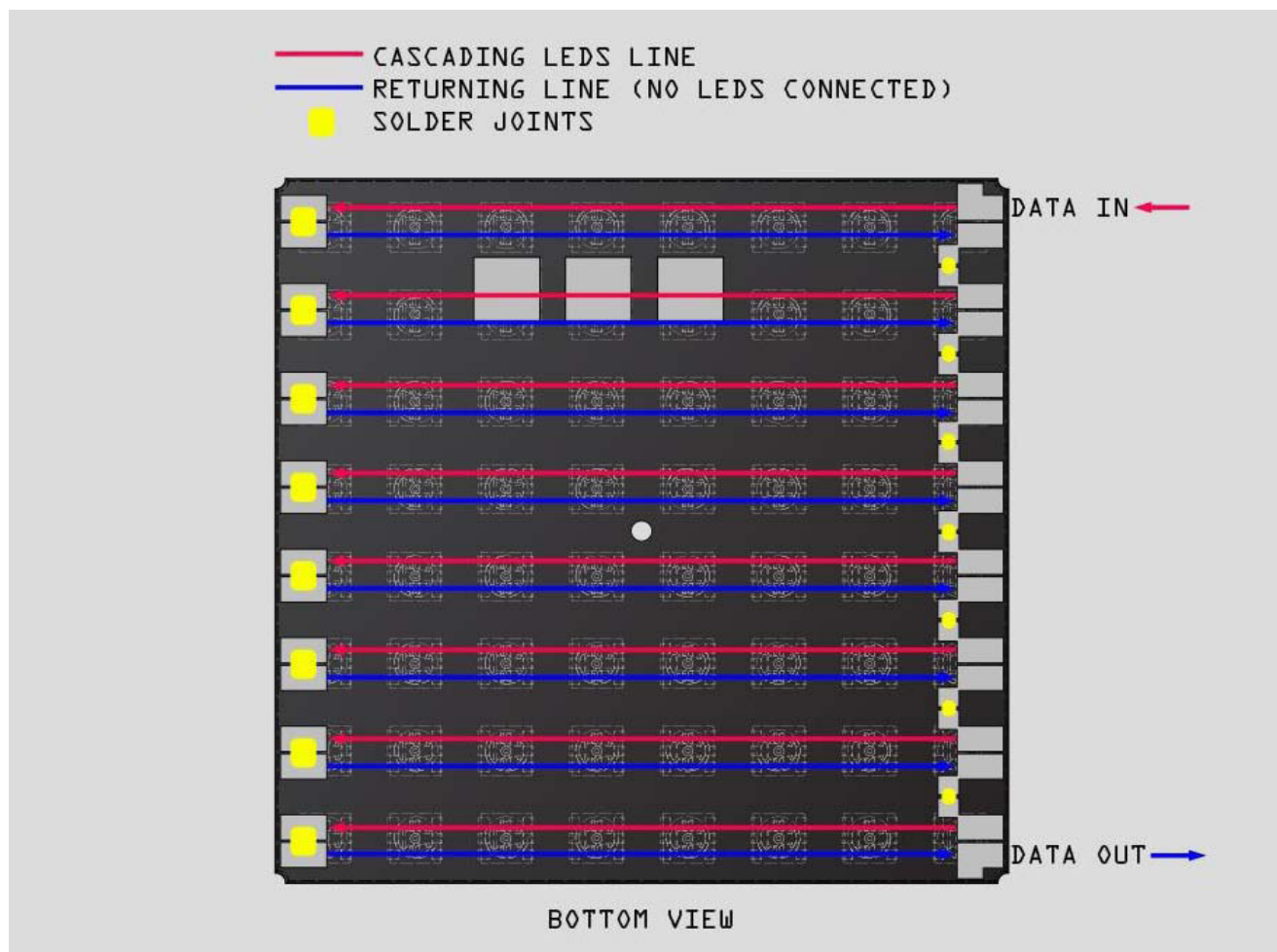


fig. 2

De onderstaande afbeelding toont aan hoe u 2 (of meer) panelen naast elkaar kunt aansluiten, door de soldeer-pads aan de zijkanten met gestrippte draad te verbinden.

Een rij bestaat nu uit 16 leds in plaats van 8 leds.

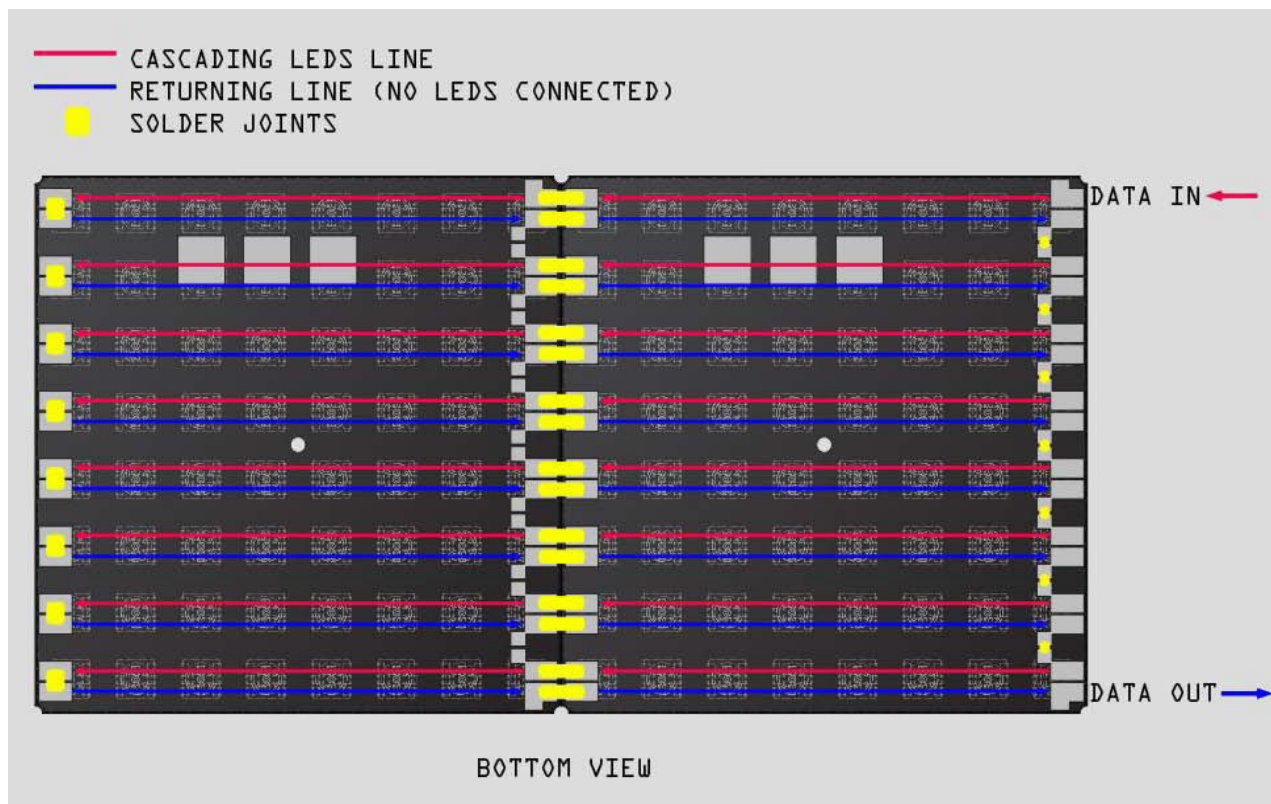


fig. 3

U kunt de panelen ook verticaal aansluiten. Sluit de DATA OUT van de vorige rij panelen aan op de DATA IN van een volgende rij panelen. Wees voorzichtig, deze aansluiting kan fragiel zijn bij grotere assemblages.

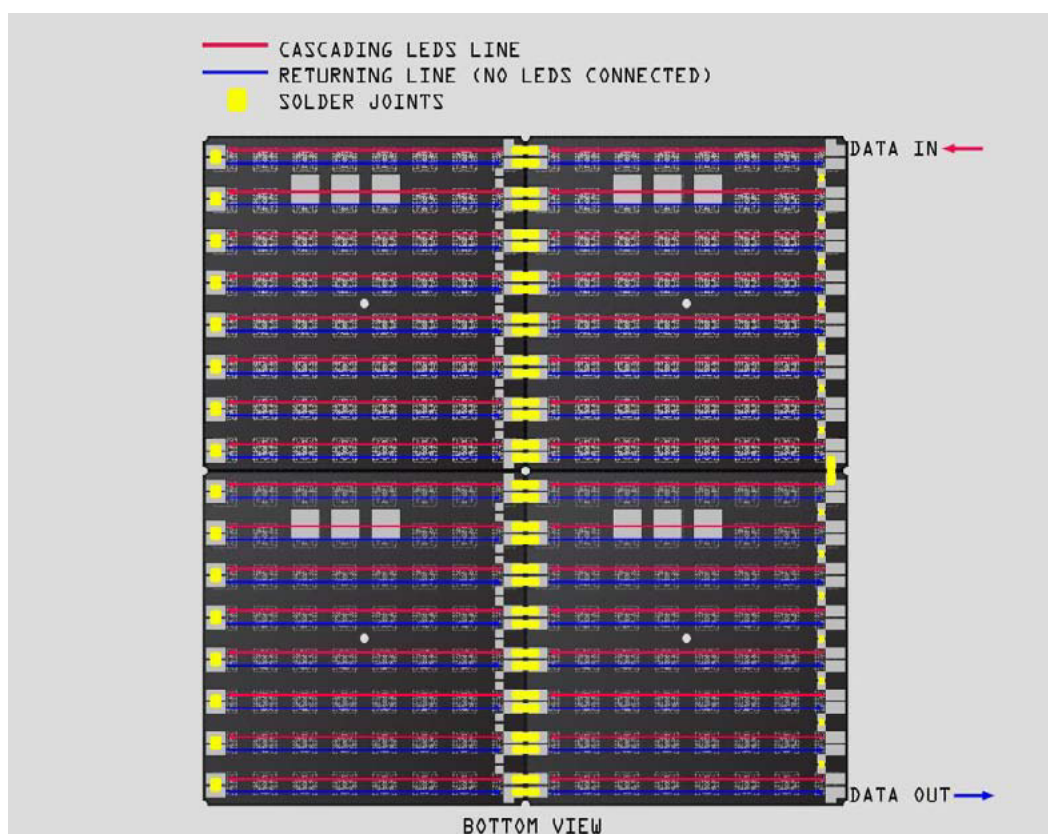


fig. 4



BELANGRIJK

- Houd de data-aansluitingen zo kort mogelijk.
- Splits de datalijn niet, dit zal niet werken. (1 led kan geen data versturen naar 2 of meer opeenvolgende leds)
- De leds functioneren met een TTL-niveau (5 V) datasignaal maar functioneren ook met een microcontroller met een uitgangsspanning van 3.3 V (5 V is beter).
- Plaats een 470 Ω -weerstand tussen de uitgangspin van de controller en de ingang van het eerste paneel. (Dit is niet altijd nodig maar het kan nuttig zijn wanneer u problemen hebt met storingsgevoelige datalijnen).

Power supply

Powering the VM207 is a bit simpler. There are 3 contacts labeled **DV+**, **LV+** and **GND**, they respectively stand for **Data Voltage +**, **LED Voltage +**, and **Ground**. The WS2812 LEDs that are mounted on the panels are the 6 pin variant and these have separate pins for the 5 V for the LED die and the 5 V for the IC die inside the package. So the DV+ pad is connected to all the IC die pins and the LD+ pin is connected to all the LED die pins. In most occasions you can just connect these two and all will be OK. If you are worried by brown-outs due to line-loss when you have a lot of panels that are operating at high brightness, you can have the IC dies on a separate 5 V supply (common ground!)

So the easiest way to power the panel is as shown in the drawing below:

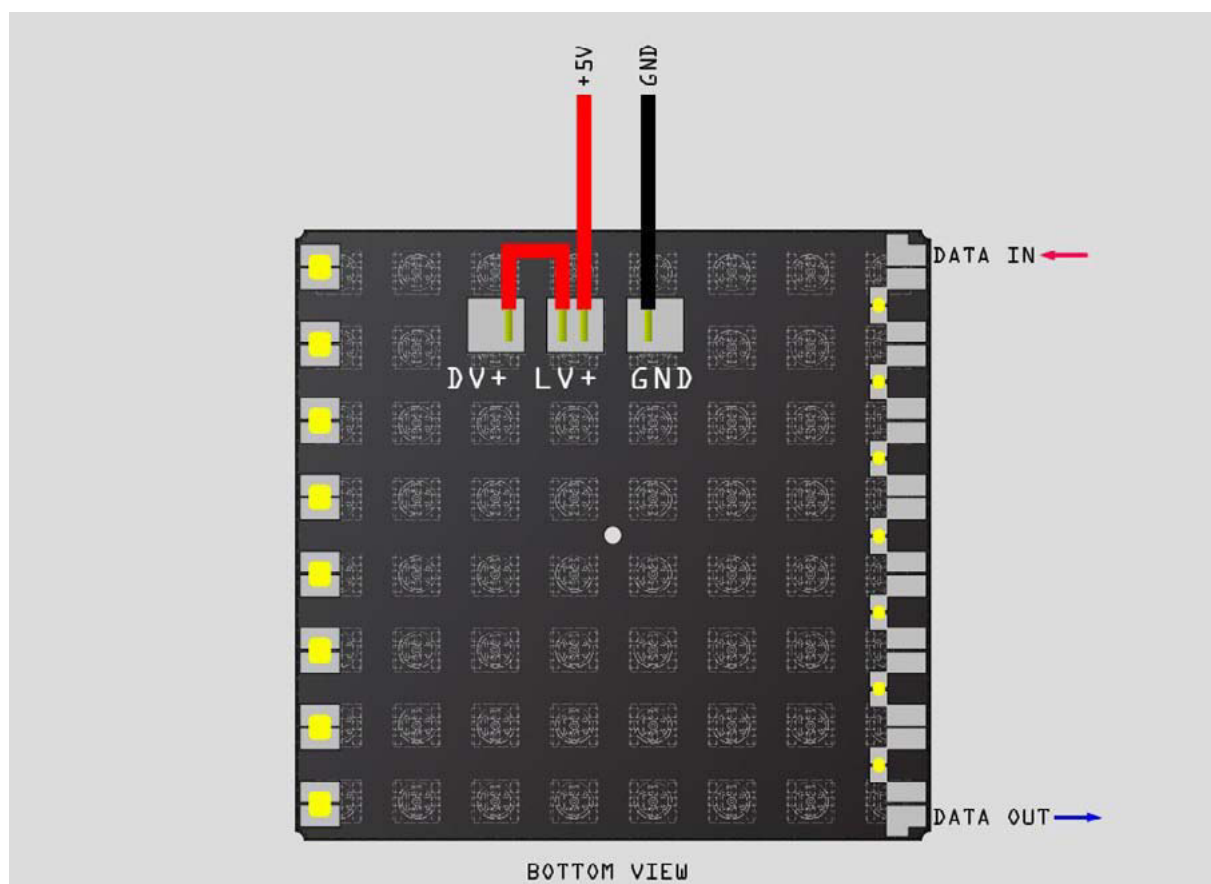


fig. 5

Wanneer u meerdere panelen gebruikt, moet u deze parallel plaatsen. **Gebruik kabels met een geschikte kabeldikte of gebruik meerdere kabels vanaf de voeding tot de panelen!**

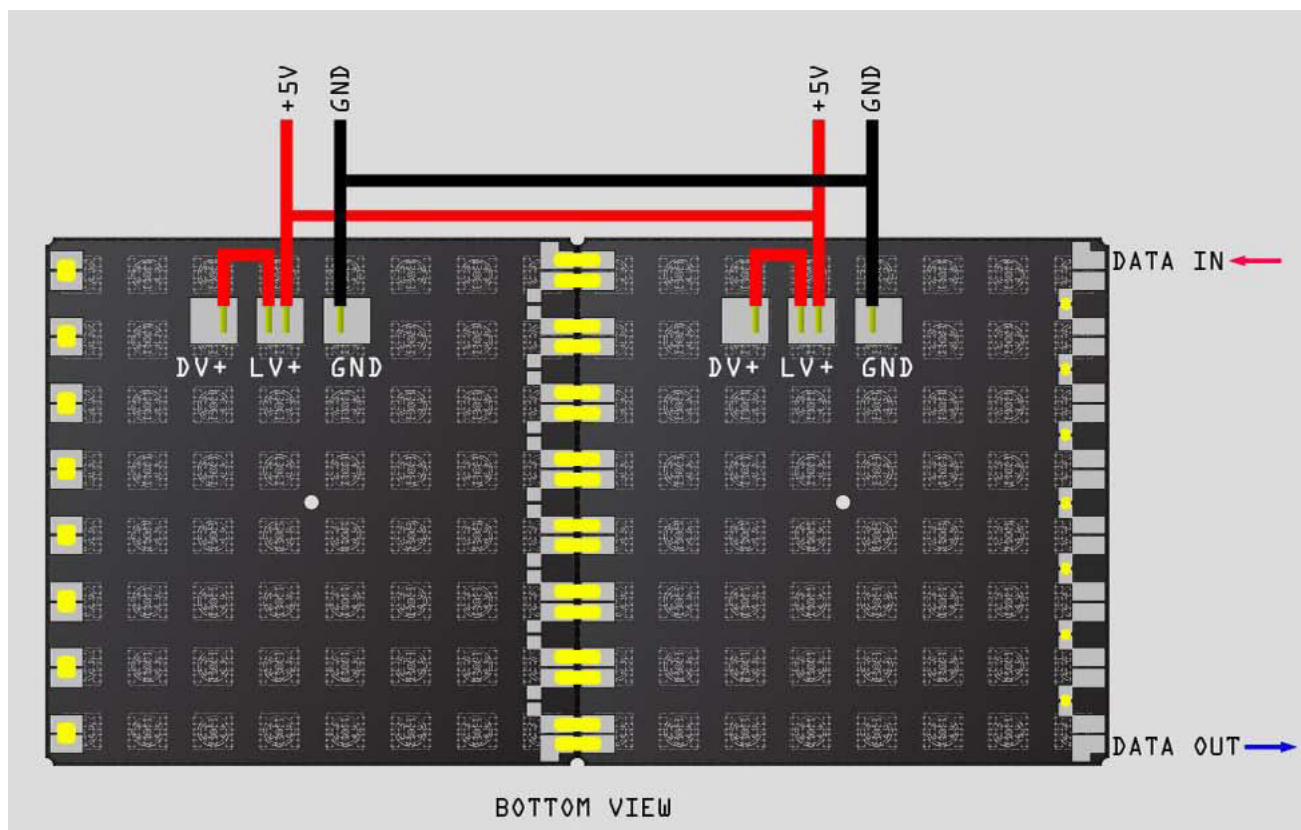


fig. 6



BELANGRIJK

- Sluit een 1000 μf -condensator (meegelev.) altijd parallel aan op de voeding.
- Schakel de voeding niet in of uit als de panelen aangesloten zijn. De in- en uitschakelpieken kunnen de leds op het paneel beschadigen.
- Sluit altijd eerst de aarddraad aan.
- Gebruik een geschikte kabel om de panelen van stroom te voorzien. Deze panelen verbruiken veel stroom (3.5 A per paneel op volle lichtsterkte). Spanningsverlies in de kabel kan een rol spelen wanneer u deze panelen gebruikt.
- Kijk niet rechtstreeks in de leds op volle lichtsterkte, dit kan een desoriënterend effect hebben.
- Het gebruik van meerdere panelen op volle lichtsterkte produceert veel warmte. Voor bepaalde toepassingen moet deze opgebouwde warmte afgevoerd worden (ventilator, koelvin, ...), om de levensduur van de panelen te garanderen.

De 64 LED RGB-matrix aansturen

Welke microcontroller gebruiken?

U kunt meerdere microcontrollers of microcontroller-platforms gebruiken, om het paneel aan te sturen. Hieronder wordt beschreven hoe u het paneel kunt gebruiken met een microcontroller-platform, dat compatibel is met de Arduino IDE (Uno, Mega, Teensy, enz.).

Als u deze niet wilt gebruiken, dan moet u de datastroom creëren om de WS2812 aan te sturen. Voor meer informatie, raadpleeg: [WS2812 DATASHEET](#).

Een paneel op een Arduino Uno aansluiten

Raadpleeg de onderstaande afbeelding om een paneel op een Arduino Uno aan te sluiten. Nu gebruiken we pin 6 als data-uitgang maar u kunt dit later in de code wijzigen. Voorzie de Arduino en de VM207 van stroom en verbind de aarddraden met elkaar.



OPGELET

Gebruik de 5 V-voeding van de Arduino niet om een of meerdere panelen van stroom te voorzien. Deze panelen verbruiken te veel stroom voor de regelaar op de Arduino. Dit zou het Arduino-board kunnen beschadigen.

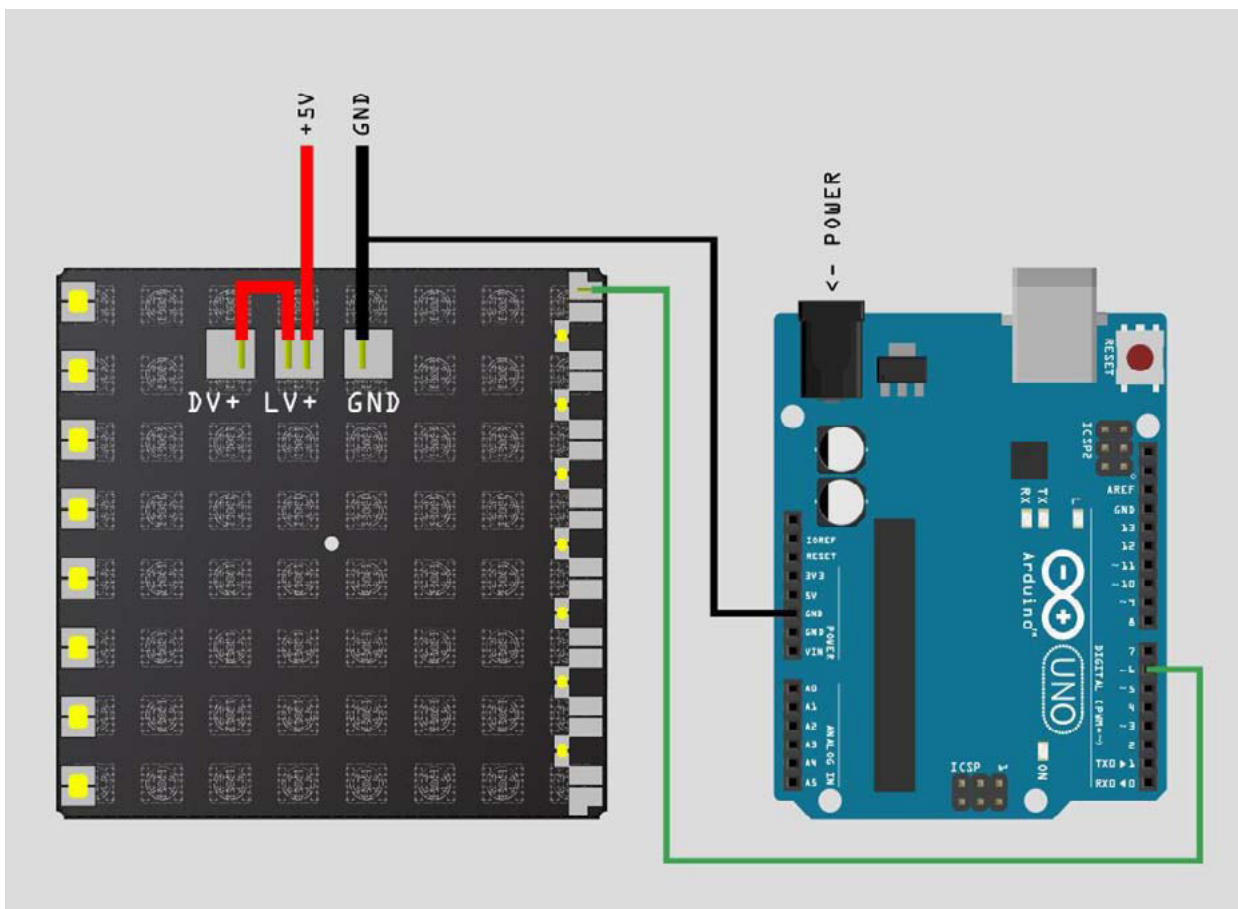


fig. 7

Adafruit NeoPixel-library

Hieronder vindt u meer uitleg over de ADAFRUIT NEOPIXEL-library. Deze library is geschreven door Adafruit en kan een reeks afzonderlijke WS2812-leds aansturen. Deze hebt u niet nodig om teksten of vormen op het paneel weer te geven maar om elke led afzonderlijk aan te sturen.

Download hier de ADAFRUIT NEOPIXEL-library: https://github.com/adafruit/Adafruit_NeoPixel (druk op de knop **Download ZIP**)

U kunt deze library in uw Arduino installeren (plaats de gedownloade en uitgepakte map in de Arduino library-map) en start de Arduino-software.

Ga naar: **File > Examples > Adafruit Neopixel > simple**, Arduino opent deze sketch.

Het bestand ziet er als volgt uit:

```
// NeoPixel Ring simple sketch (c) 2013 Shae Erisson
// released under the GPLv3 license to match the rest of the Adafruit NeoPixel library

#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
    #include <avr/power.h>
#endif

// Which pin on the Arduino is connected to the NeoPixels?
// On a Trinket or Gemma we suggest changing this to 1
#define PIN            6

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS      16

// When we setup the NeoPixel library, we tell it how many pixels, and which pin to use to send signals.
// Note that for older NeoPixel strips you might need to change the third parameter--see the strand-test
// example for more information on possible values.
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
```

Wat gebeurt er?

Het eerste deel gaat over de Adafruit-library toevoegen.

#define PIN 6 geeft aan dat de constante 'PIN' = 6 is en dat de datatstream afkomstig is van pin 6 van de Arduino. Als u deze naar 13 wilt wijzigen, dan moet u deze codelijn veranderen in: **#define PIN 13**

#define NUMPIXELS 16 geeft aan hoeveel leds zijn aangesloten om de Arduino aan te sturen. Om 1 paneel aan te sturen, ziet de lijn er als volgt uit: **#define NUMPIXELS 64**. Om 2 panelen aan te sturen, is dit 128, 3 panelen = 192, enz.

De volgende lijn ziet er als volgt uit: **Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);**

Deze lijn geeft een naam aan het paneel of verzameling panelen, hier is dit "**pixels**". U kunt deze naam gemakkelijk veranderen naar "panel", maar dan moet u dit ook doen voor alle vermeldingen van het woord "pixels" in het programma.

U maakt het programma duidelijk hoe "pixels" (ons paneel) is opgebouwd: **Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);**

- NUMPIXELS = Deze waarde is eerder in dit programma vastgelegd. Dit is het aantal pixels die moeten aangestuurd worden.
- PIN = Deze waarde is eerder in dit programma vastgelegd. Dit is de uitgangspin waar de display data-stroom zich bevindt.
- NEO_GRB = Behouden voor standaard WS2812-leds.
- NEO_KHZ800 = Behouden voor standaard WS2812-leds.

De onderstaande code is een variabele die een waarde opslaat, die als vertragingsswaarde in de hoofdfunctie wordt gebruikt. Verander deze waarde en de snelheid waarmee de 'for'-lus de 'loop'-functie doorloopt, zal veranderen.

```
int delayval = 500; // delay for half a second
```

Below that piece of code there is the setup function that looks like this:

```
void setup() {  
  // This is for Trinket 5V 16MHz, you can remove these three lines if you are not using a Trinket  
  #if defined (__AVR_ATtiny85__)  
    if (F_CPU == 16000000) clock_prescale_set(clock_div_1);  
  #endif  
  // End of trinket special code  
  
  strip.begin();  
}
```

Het belangrijkste deel van de code is de lijn **strip.begin()**. Deze activeert de leds. Vergeet deze stap niet in uw eigen code.

Vervolgens is er de hoofd 'loop'-functie:

```
void loop() {

    // For a set of NeoPixels the first NeoPixel is 0, second is 1, all the way up to the count of pixels minus one.

    for(int i=0;i<NUMPIXELS;i++){

        // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
        pixels.setPixelColor(i, pixels.Color(0,150,0)); // Moderately bright green color.

        pixels.show(); // This sends the updated pixel color to the hardware.

        delay(delayval); // Delay for a period of time (in milliseconds).

    }
}
```

Dit is het hart van het programma. De 'loop'-functie wordt telkens herhaald, en deze 'loop'-functie omvat een 'for'-lus die deze lijnen elke halve seconde uitvoert.

```
pixels.setPixelColor(i, pixels.Color(0,0,255)); // Bright blue color.

pixels.show(); // This sends the updated pixel color to the hardware.

delay(delayval); // Delay for a period of time (in milliseconds).
```

De eerste lijn geeft een groene kleur aan de pixel op de plaats waar de waarde "i" is opgeslagen, maar deze informatie wordt enkel geüpdatet wanneer de tweede lijn "pixels.show" wordt opgeroepen. De lijnen worden elke 500 ms opgeroepen en de waarde "i" blijft telkens toenemen met 1. Raadpleeg: [https://www.arduino.cc/en/reference/for about how for-loops work](https://www.arduino.cc/en/reference/for%20about%20how%20for-loops%20work)) Het paneel licht groen op wanneer u deze code op een Arduino toepast.

Test, compileer en kopieer de code naar uw Arduino. Sluit alles aan. Het paneel licht langzaam op (elke 500 ms), led per led. Eenmaal het paneel volledig groen oplicht, blijven de leds branden. Om te veranderen, wijzig de 'loop'-functie en de ingestelde vertragingstijd:

```
void loop() {

    // For a set of NeoPixels the first NeoPixel is 0, second is 1, all the way up to the count of pixels minus one.
```

```

for(int i=0;i<NUMPIXELS;i++){
    // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
    pixels.setPixelColor(i, pixels.Color(0,0,255)); // Brigh blue color.

    pixels.show(); // This sends the updated pixel color to the hardware.

    delay(delayval); // Delay for a period of time (in milliseconds).
}
for(int i=0;i<NUMPIXELS;i++){
    pixels.setPixelColor(i, pixels.Color(0,0,0)); // No color (dark).
}
pixels.show(); //Updating the panel to show nothing.
}

```

Na de eerste 'for'-lus wist deze code alle waarden m.b.v. een tweede 'for'-lus. De kleur is veranderd naar helder diepblauw, en is zeer helder. Om de helderheid van het volledige paneel te veranderen: **pixels.setBrightness(20);** deze functie accepteert een waarde tussen 0 en 255 (0 = donker en 255 = volle lichtsterkte). U kunt zien hoe dit de display beïnvloedt wanneer u de code van de 'loop'-functie verandert:

```

void loop() {
    pixels.setBrightness(20); // Setting the brightness really low.
    // For a set of NeoPixels the first NeoPixel is 0, second is 1, all the way up to the count of pixels minus one.
    for(int i=0;i<NUMPIXELS;i++){
        // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
        pixels.setPixelColor(i, pixels.Color(0,0,255)); // Bright blue color.

        pixels.show(); // This sends the updated pixel color to the hardware.

        delay(delayval); // Delay for a period of time (in milliseconds).
    }
    for(int i=0;i<NUMPIXELS;i++){
        pixels.setPixelColor(i, pixels.Color(0,0,0)); // No color (dark).
    }
    pixels.show(); // Updating the panel to show nothing.
}

```

De volledige code ziet er als volgt uit (zie onder). Experimenteer eerst voordat u verdergaat.

```

// NeoPixel Ring simple sketch (c) 2013 Shae Erisson
// released under the GPLv3 license to match the rest of the AdaFruit NeoPixel library

#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
    #include <avr/power.h>
#endif

```

```

// Which pin on the Arduino is connected to the NeoPixels?
// On a Trinket or Gemma we suggest changing this to 1
#define PIN                6

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS          64

// When we setup the NeoPixel library, we tell it how many pixels, and which pin to use to send sig-
nals.
// Note that for older NeoPixel strips you might need to change the third parameter--see the strand-
test
// example for more information on possible values.
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

int delayval = 50; // delay for half a second

void setup() {
  // This is for Trinket 5V 16MHz, you can remove these three lines if you are not using a Trinket
#ifdef __AVR_ATtiny85__
  if (F_CPU == 16000000) clock_prescale_set(clock_div_1);
#endif
  // End of trinket special code

  pixels.begin(); // This initializes the NeoPixel library.
}

```

Adafruit NeoMatrix-library

Als u teksten of vormen op het paneel wilt weergeven, dan kunt u de ADAFRUIT NEOMATRIX-library gebruiken. Om deze library correct te laten functioneren, moet u ook de ADAFRUIT GFX-library installeren. Deze zorgt voor de grafische lay-out. De NeoMatrix-library stuurt deze gegevens naar de panelen.

Download hier de ADAFRUIT NEOMATRIX-library: https://github.com/adafruit/Adafruit_NeoMatrix (druk op de knop **Download ZIP**)

Download hier de ADAFRUIT GFT-library: <https://github.com/adafruit/Adafruit-GFX-Library> (druk op de knop **Download ZIP**)

U kunt deze libraries opnieuw in de Arduino installeren. Plaats de gedownloade en uitgedrukte mappen in de library-map van de Arduino en start de Arduino-software. Zorg ervoor dat u de ADAFRUIT NEOPIXEL-library geïnstalleerd hebt.

Ga naar: **File > Examples > Adafruit Neopixel > simple** Arduino opent deze sketch.

Het geopende bestand ziet er als volgt uit:

```

#include <Adafruit_GFX.h>
#include <Adafruit_NeoMatrix.h>
#include <Adafruit_NeoPixel.h>
#ifndef PSTR
  #define PSTR // Make Arduino Due happy
#endif

#define PIN 6

// MATRIX DECLARATION:
// Parameter 1 = width of NeoPixel matrix
// Parameter 2 = height of matrix
// Parameter 3 = pin number (most are valid)
// Parameter 4 = matrix layout flags, add together as needed:
//   NEO_MATRIX_TOP, NEO_MATRIX_BOTTOM, NEO_MATRIX_LEFT, NEO_MATRIX_RIGHT:
//     Position of the FIRST LED in the matrix; pick two, e.g.
//     NEO_MATRIX_TOP + NEO_MATRIX_LEFT for the top-left corner.
//   NEO_MATRIX_ROWS, NEO_MATRIX_COLUMNS: LEDs are arranged in horizontal
//     rows or in vertical columns, respectively; pick one or the other.
//   NEO_MATRIX_PROGRESSIVE, NEO_MATRIX_ZIGZAG: all rows/columns proceed
//     in the same order, or alternate lines reverse direction; pick one.
//   See example below for these values in action.
// Parameter 5 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic v1 (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)

// Example for NeoPixel Shield.  In this application we'd like to use it
// as a 5x8 tall matrix, with the USB port positioned at the top of the
// Arduino.  When held that way, the first pixel is at the top right, and
// lines are arranged in columns, progressive order.  The shield uses
// 800 KHz (v2) pixels that expect GRB colour data.
Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(5, 8, PIN,
  NEO_MATRIX_TOP      + NEO_MATRIX_RIGHT +
  NEO_MATRIX_COLUMNS + NEO_MATRIX_PROGRESSIVE,
  NEO_GRB              + NEO_KHZ800);

```

Het eerste deel omvat de 3 libraries (NEOPIXEL, NEOMATRIX en GFX).

Vervolgens stellen we opnieuw een uitgangspin in. De werking is dezelfde als in het vorige hoofdstuk.

Hieronder wordt uitgelegd hoe u de library configureert, zodat deze weet hoe de assemblage van de panelen eruitziet. Eerst geven we een naam aan het paneel, in dit geval: "matrix". Er zijn 5 parameters:

- breedte van de matrix (aantal leds: 8, 16, 32, ...)
- hoogte van de matrix (aantal leds: 8, 16, 32, ...)
- pin-nummer van de uitgangspin.
- De 4de parameter beschrijft de lay-out van het paneel en is gevormd door parameters die worden samengeteld. Eerst moet u bepalen waar zich de eerste led in de matrix bevindt. Meestal is dit in de linkerbovenhoek: **NEO_MATRIX_TOP + NEO_MATRIX_LEFT**. Kies of de leds in rijen of in kolommen worden aangesloten. Als u de panelen horizontaal gebruikt, kies: **NEO_MATRIX_ROWS**. Kies welke rijen (of kolommen) 'progressive' zijn of zigzaggen. Met de VM207 is dit ALTIJD 'progressive', dus wordt dit: **NEO_MATRIX_PROGRESSIVE**.
- De laatste parameter betreft welke leds u met de VM207 wilt gebruiken. Dit is ALTIJD: **NEO_GRB + NEO_KHZ800**.

Om 1 paneel aan te sturen, moeten deze parameters worden samengevoegd (zie onder). Wijzig de code naar het voorbeeld hieronder, omdat dit paneel ontworpen is om een 5 x 8 paneel aan te sturen met andere eigenschappen.

```
Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(8, 8, PIN,
  NEO_MATRIX_TOP      + NEO_MATRIX_LEFT +
  NEO_MATRIX_ROWS + NEO_MATRIX_PROGRESSIVE,
  NEO_GRB              + NEO_KHZ800);
```

Als u een groter display wilt maken met VM207-panelen, pas dan de parameters overeenkomstig aan.

De onderstaande code is een reeks van 3 kleuren die we doorlopen, om het voorbeeld een specialer effect te geven:

```
const uint16_t colors[] = {
  matrix.Color(255, 0, 0), matrix.Color(0, 255, 0), matrix.Color(0, 0, 255) };
```

Het belangrijkste onderdeel van de setup-functie, is **matrix.begin()**. Deze functie start het paneel op. De daaropvolgende code zorgt voor de grafische inhoud, die op het paneel wordt weergegeven. Compileer de code en kopieer deze naar de Arduino UNO. Controleer of het werkt. Indien nodig, kopieer de onderstaande code:

```
// Adafruit_NeoMatrix example for single NeoPixel Shield.
// Scrolls •Howdy across the matrix in a portrait (vertical) orientation.

#include <Adafruit_GFX.h>
#include <Adafruit_NeoMatrix.h>
#include <Adafruit_NeoPixel.h>
#ifndef PSTR
  #define PSTR // Make Arduino Due happy
#endif

#define PIN 6
```

```

// MATRIX DECLARATION:
// Parameter 1 = width of NeoPixel matrix
// Parameter 2 = height of matrix
// Parameter 3 = pin number (most are valid)
// Parameter 4 = matrix layout flags, add together as needed:
//   NEO_MATRIX_TOP, NEO_MATRIX_BOTTOM, NEO_MATRIX_LEFT, NEO_MATRIX_RIGHT:
//     Position of the FIRST LED in the matrix; pick two, e.g.
//     NEO_MATRIX_TOP + NEO_MATRIX_LEFT for the top-left corner.
//   NEO_MATRIX_ROWS, NEO_MATRIX_COLUMNS: LEDs are arranged in horizontal
//     rows or in vertical columns, respectively; pick one or the other.
//   NEO_MATRIX_PROGRESSIVE, NEO_MATRIX_ZIGZAG: all rows/columns proceed
//     in the same order, or alternate lines reverse direction; pick one.
//   See example below for these values in action.
// Parameter 5 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic v1 (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)

// Example for NeoPixel Shield. In this application we'd like to use it
// as a 5x8 tall matrix, with the USB port positioned at the top of the
// Arduino. When held that way, the first pixel is at the top right, and
// lines are arranged in columns, progressive order. The shield uses
// 800 KHz (v2) pixels that expect GRB color data.
Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(8, 8, PIN,
  NEO_MATRIX_TOP + NEO_MATRIX_LEFT +
  NEO_MATRIX_ROWS + NEO_MATRIX_PROGRESSIVE,
  NEO_GRB + NEO_KHZ800);

const uint16_t colors[] = {
  matrix.Color(255, 0, 0), matrix.Color(0, 255, 0), matrix.Color(0, 0, 255) };

void setup() {
  matrix.begin();
  matrix.setTextWrap(false);
  matrix.setBrightness(40);
  matrix.setTextColor(colors[ 0 ] );
}

int x = matrix.width();
int pass = 0;

void loop() {
  matrix.fillScreen(0);
  matrix.setCursor(x, 0);
  matrix.print(F("Howdy"));
  if(--x < -36) {

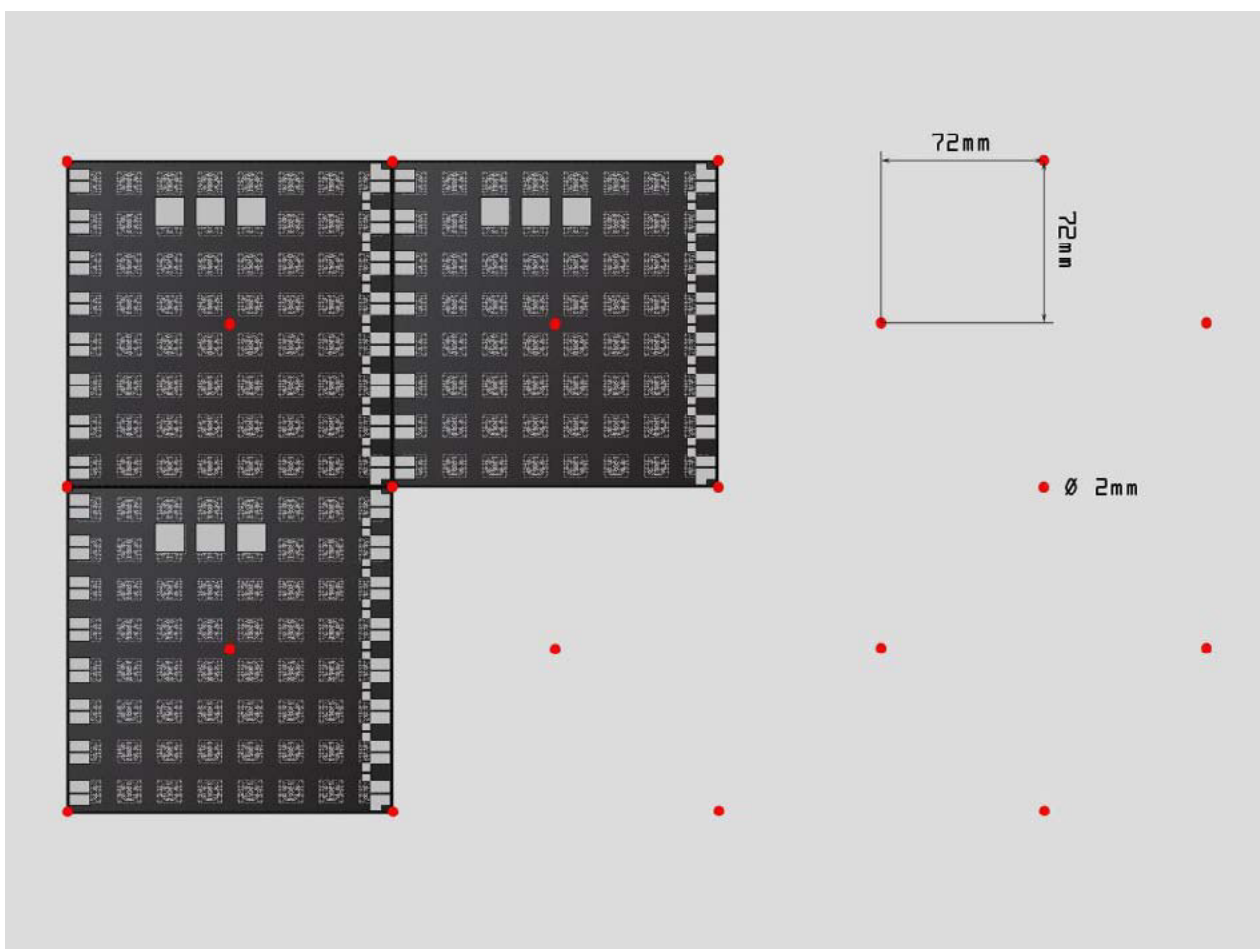
```

```
x = matrix.width();  
if(++pass >= 3) pass = 0;  
matrix.setTextColors(colors[pass]);  
}  
matrix.show();
```

De 64 LED RGB-matrix monteren

Montagegaten

De panelen zijn voorzien van montagegaten, zodat deze een patroon vormen wanneer u meerdere panelen gebruikt. De montagegaten hebben een diameter van 2 mm. Gebruik een kleine M2-schroef om de panelen te monteren. De montagegaten staan op 72 mm van elkaar en vormen een tegel. Zie afbeelding hieronder:

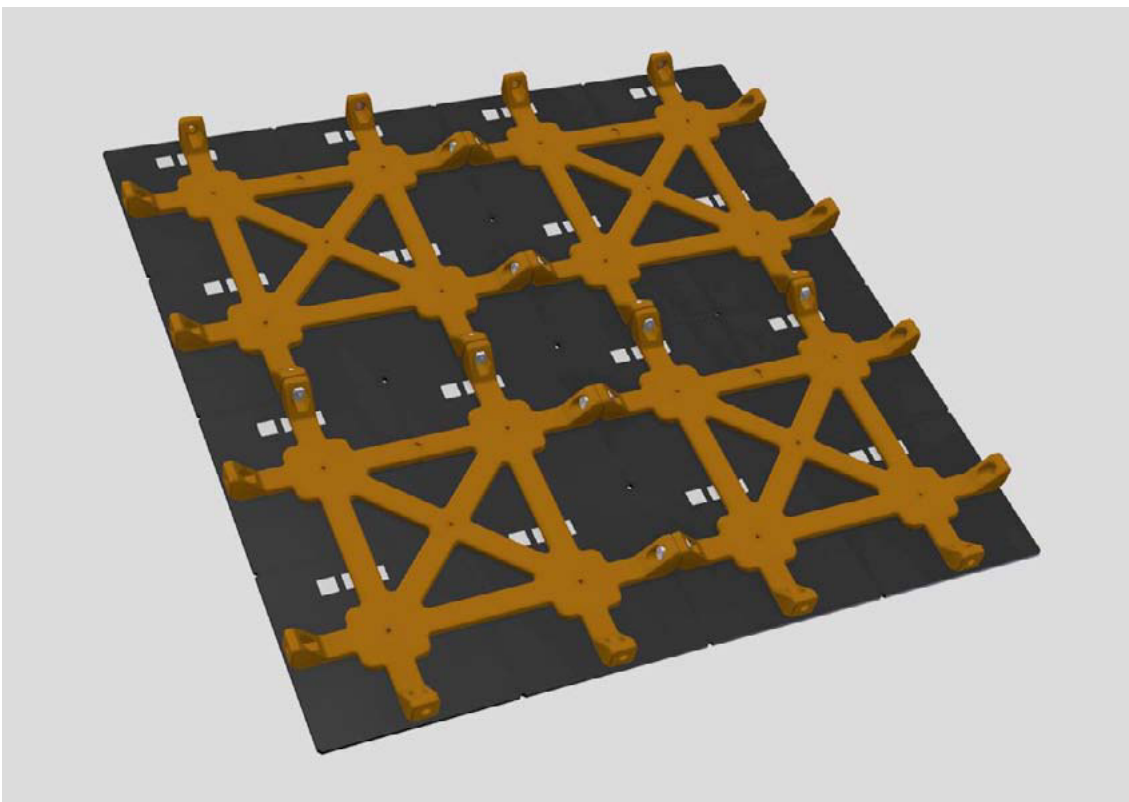
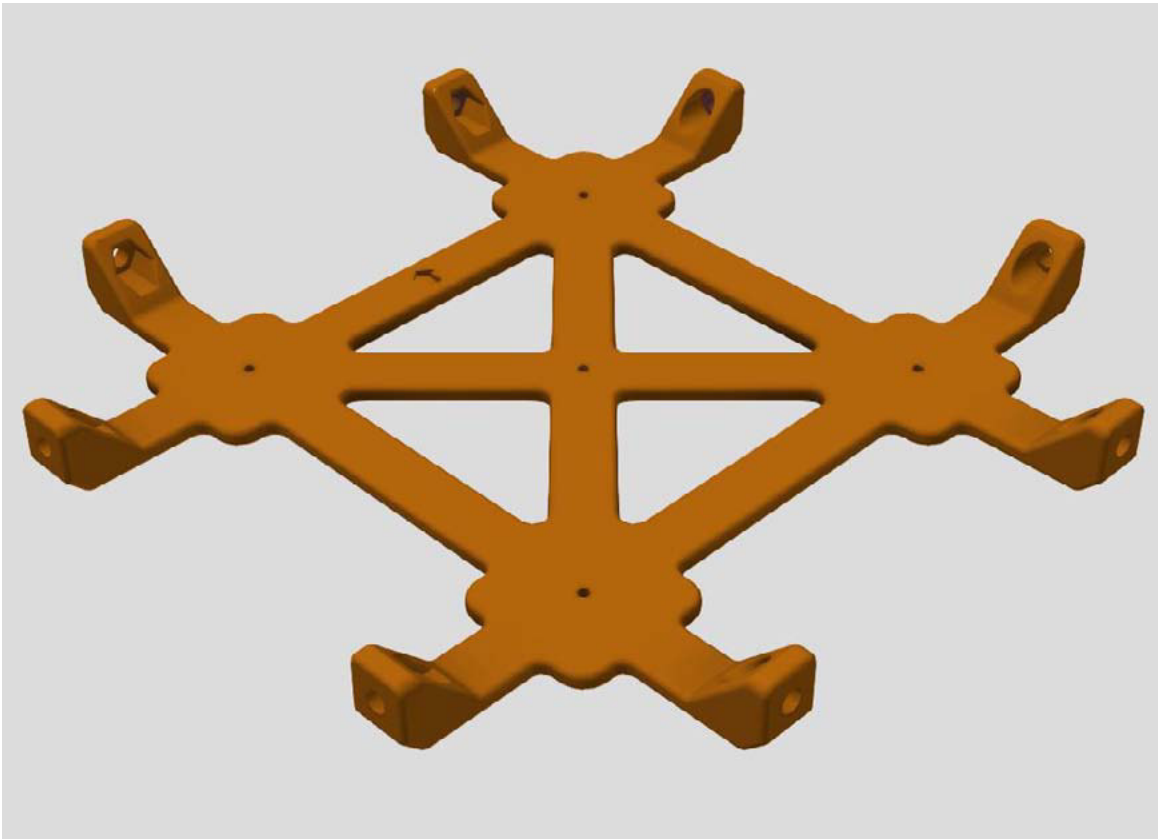


3D-montagebeugels

Als u een 3D-printer hebt, download het bestand voor de montagebeugels:

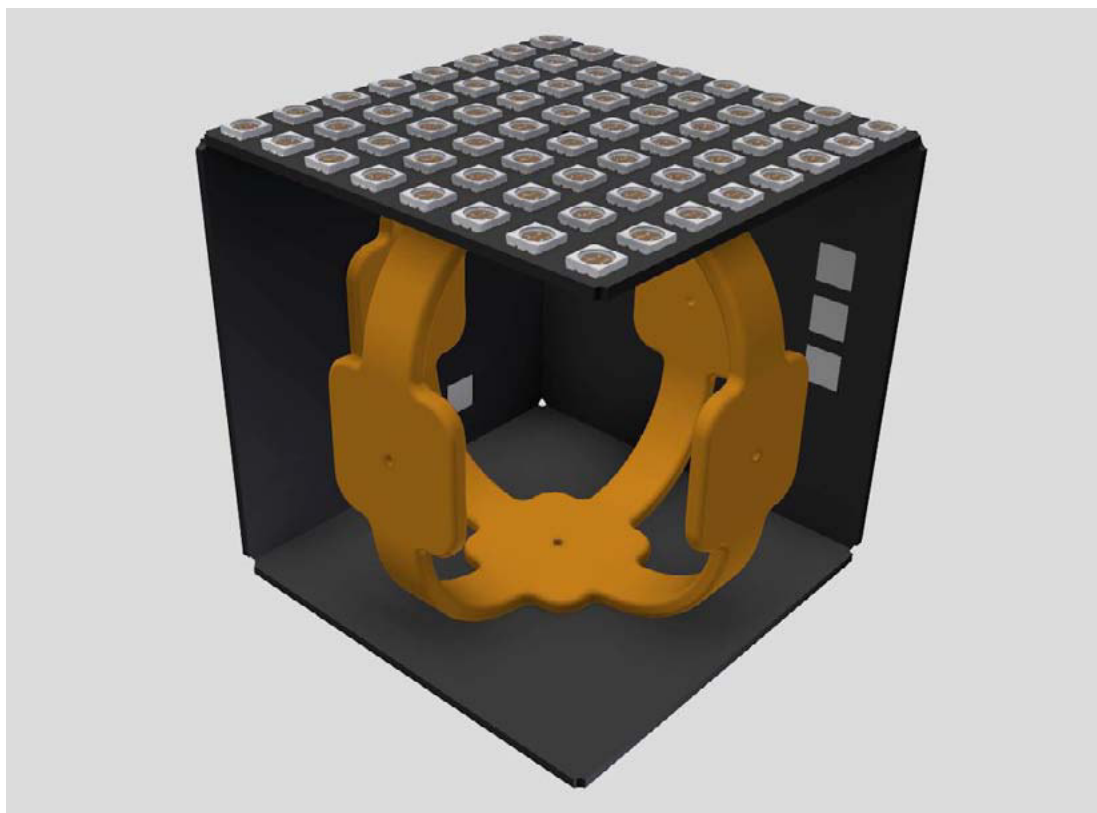
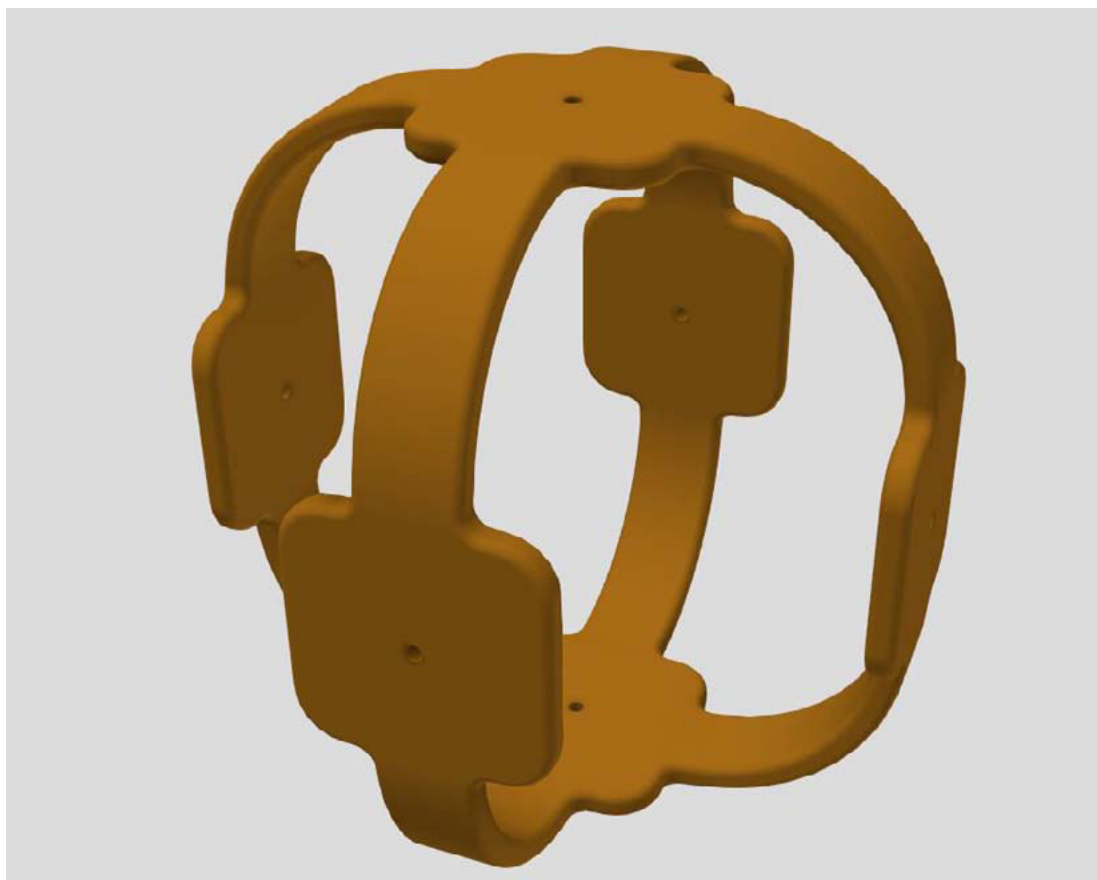
[vm207-tile_mount.stl](#)

Gebruik M3 x 10 bouten om meerdere beugels te monteren.



[vm207-cube_mount.stl](#)

Vereist 6 x VM207-paneel. Wees extra voorzichtig bij het koppelen van de panelen!



velleman®

ORDERCODE: VM207

REVISION: HVM207'1



VellemanProjects



@Velleman_RnD

VELLEMAN nv - Legen Heirweg 33, Gavere (Belgium)
vellemanprojects.com