Welkom!

Hartelijk dank voor uw aankoop van ons *AZ-Delivery 0,96 inch OLED I2C scherm*. Op de volgende pagina's wordt uitgelegd hoe je dit handige apparaat kunt gebruiken en instellen.

Veel plezier!



Toepassingsgebieden
Onderwijs en onderwijs: Gebruik op scholen, universiteiten en opleidingsinstellingen om de basisprincipes van elektronica, programmeren en ingebedde systemen te onderwijzen. Onderzoek en ontwikkeling: Gebruik in onderzoeks- en ontwikkelingsprojecten om prototypes en experimenten te creëren op het gebied van elektronica en computerwetenschappen. Prototypeontwikkeling: gebruik bij de ontwikkeling en het testen van nieuwe elektronische schakelingen en apparaten. Hobby- en makerprojecten: gebruikt door elektronicaliefhebbers en hobbyisten om doe-het-zelf-projecten te ontwikkelen en te implementeren.

Benodigde kennis en vaardigheden
Basiskennis van elektronica en elektrotechniek. Kennis van programmeren, vooral in de programmeertaal C/C++. Vermogen om schema's te lezen en eenvoudige circuits te ontwerpen. Ervaring met elektronische componenten en solderen.

Bedrijfsomstandigheden
Om schade te voorkomen mag het product alleen worden gebruikt met de spanningen die op het gegevensblad staan vermeld. Voor gebruik is een gestabiliseerde gelijkstroomvoedingsbron vereist. Bij aansluiting op andere elektronische componenten en circuits moeten de maximale stroom- en spanningslimieten in acht worden genomen om overbelasting en schade te voorkomen.

Milieu omstandigheden
Het product moet in een schone, droge omgeving worden gebruikt om schade veroorzaakt door vocht of stof te voorkomen. Bescherm het product tegen direct zonlicht (UV), omdat dit de levensduur van het display negatief kan beïnvloeden.

Beoogd gebruik
Het product is ontworpen voor gebruik in onderwijs-, onderzoeks- en ontwikkelingsomgevingen. Het wordt gebruikt voor het ontwikkelen, programmeren en prototypen van elektronische projecten en toepassingen. Het product is niet bedoeld als een afgewerkt consumentenproduct, maar eerder als een hulpmiddel voor technisch onderlegde gebruikers, waaronder ingenieurs, ontwikkelaars, onderzoekers en studenten.

Onjuist voorzienbaar gebruik
Het product is niet geschikt voor industrieel gebruik of veiligheidsrelevante toepassingen. Gebruik van het product in medische apparaten of voor lucht- en ruimtevaartdoeleinden is niet toegestaan

beschikbaarheid
Niet weggooien met het huishoudelijk afval! Uw product is volgens het Europese product Richtlijn betreffende de milieuvriendelijke verwijdering van afgedankte elektrische en elektronische apparatuur. De waardevolle grondstoffen die daarin zitten, kunnen worden gerecycled worden. De toepassing van deze richtlijn draagt bij tot de bescherming van het milieu en de gezondheid. Voor het inleveren en retourneren kunt u gebruik maken van het door uw gemeente ingestelde inzamelpunt Recycling van oude elektrische en elektronische apparaten. WEEE-reg.nr.: DE 62624346

elektrostatische ontlading
Het display is gevoelig voor elektrostatische ontlading (ESD), waardoor de elektronische componenten kunnen worden beschadigd of vernietigd. Neem de volgende veiligheidsinstructies in acht om ESD-gevaren te voorkomen: Let op: Elektrostatische ladingen op uw lichaam kunnen het display beschadigen. Opmerking: Aard uzelf door een antistatische polsband te dragen die is aangesloten op een geaard oppervlak of door een geaard metalen oppervlak aan te raken voordat u het beeldscherm aanraakt. Let op: Gebruik antistatische matten en tassen om het display te beschermen. Opmerking: Plaats het display op een antistatische werkmat en bewaar het in antistatische zakken wanneer het niet in gebruik is. Let op: Een schone en geaarde werkplek minimaliseert het risico op ESD. Actie: Houd uw werkplek schoon en vrij van materialen die elektrostatische ladingen kunnen veroorzaken. Zorg ervoor dat alle gebruikte oppervlakken geaard zijn.

veiligheidsinstructies
Hoewel het display voldoet aan de vereisten van de RoHS-richtlijn (2011/65/EU) en geen gevaarlijke stoffen bevat in hoeveelheden boven de toegestane limieten, kunnen er nog steeds chemische restgevaren bestaan. Houd rekening met de volgende veiligheidsinstructies: Let op: De achterkant van het display en de printplaat kunnen chemische resten vrijgeven tijdens de productie of tijdens het gebruik. Opmerking: Draag beschermende handschoenen als u het beeldscherm langdurig hanteert of installeert om huidirritatie te voorkomen. Let op: Elektronische componenten kunnen kleine hoeveelheden vluchtige orga-

nische stoffen (VOS) uitstoten, vooral als het display nieuw is. Opmerking: Zorg ervoor dat u in een goed geventileerde ruimte werkt om de concentratie van dampen in de lucht tot een minimum te beperken. Let op: Gebruik geen agressieve chemicaliën of oplosmiddelen om het scherm schoon te maken, aangezien deze de beschermende coating of elektronica kunnen beschadigen. Opmerking: Gebruik een antistatisch reinigingsdoekje of een speciale elektronicareiniger om het display voorzichtig schoon te maken. Hoewel het display voldoet aan de vereisten van de RoHS-richtlijn (2011/65/EU) en geen gevaarlijke stoffen bevat in hoeveelheden boven de toegestane limieten, kunnen er nog steeds chemische restgevaren bestaan. Houd rekening met de volgende veiligheidsinstructies: Let op: De achterkant van het display en de printplaat kunnen chemische resten vrijgeven tijdens de productie of tijdens het gebruik. Opmerking: Draag beschermende handschoenen als u het beeldscherm langdurig hanteert of installeert om huidirritatie te voorkomen. Let op: Elektronische componenten kunnen kleine hoeveelheden vluchtige organische stoffen (VOS) uitstoten, vooral als het display nieuw is. Opmerking: Zorg ervoor dat u in een goed geventileerde ruimte werkt om de concentratie van dampen in de lucht tot een minimum te beperken. Let op: Gebruik geen agressieve chemicaliën of oplosmiddelen om het scherm schoon te maken, aangezien deze de beschermende coating of elektronica kunnen beschadigen. Opmerking: Gebruik een antistatisch reinigingsdoekje of een speciale elektronicareiniger om het display voorzichtig schoon te maken. Het display bevat gevoelige elektronische componenten en een toplaag. Onjuiste behandeling of overmatige druk kan schade aan het beeldscherm of letsel veroorzaken. Neem de volgende veiligheidsinstructies in acht om mechanische gevaren te voorkomen: Let op: De cover van het display is kwetsbaar en kan breken als deze verkeerd wordt behandeld. Opmerking: Oefen geen sterke druk uit en buig het scherm niet. Behandel het display voorzichtig en alleen bij de printplaat om breuken te voorkomen. Let op: Vallen of stoten kan het oppervlak van het scherm doen barsten en de elektronische componenten aan de achterkant beschadigen. Let op: Laat het beeldscherm niet vallen en bescherm het tegen stoten. Gebruik bij het werken een zachte ondergrond om krassen te voorkomen. Let op: Als het display breekt, kunnen scherpe stukjes glas verwondingen veroorzaken. Opmerking: Als het display breekt, ga dan voorzichtig om met de fragmenten en draag beschermende handschoenen om snijwonden te voorkomen. Gooi de stukjes glas veilig weg. Opmerking: Onjuiste bevestiging kan leiden tot mechanische spanning en breuk van het display. Actie: Bevestig het beeldscherm stevig en zonder overmatige druk. Gebruik geschikte beugels of behuizingen om het beeldscherm stabiel te monteren. Let op: Onjuiste reinigingsmethoden kunnen het oppervlak krassen of beschadigen. Let op: Gebruik alleen zachte, antistatische doeken om het display schoon te maken. Vermijd agressieve schoonmaakmiddelen en sterke wrijving. Het display werkt met elektrische spanningen en stromen die bij onjuist gebruik elektrische schokken, kortsluiting of brand kunnen veroorzaken. Houd rekening met de volgende veiligheidsinstructies: Let op: Gebruik het product alleen met de aangegeven spanningen. Opmerking: De prestatiegrenzen van het product vindt u in het bijbehorende gegevensblad Opmerking: Onjuiste spanningsbronnen kunnen het display beschadigen of gevaarlijke situaties veroorzaken. Actie: Gebruik alleen geteste en geschikte voedingen of batterijen om uw circuits van stroom te voorzien. Zorg ervoor dat de spanningsbron voldoet aan de vereisten van het display. Let op: Vermijd kortsluiting tussen de connectoren en componenten van het product Opmerking: Zorg ervoor dat geen geleidende voorwerpen de printplaat raken of overbruggen. Gebruik geïsoleerd gereedschap en let op de plaatsing van de aansluitingen. Let op: Voer geen werkzaamheden aan het product uit als het op een stroombron is aangesloten. Opmerking: Koppel het product los van de voeding voordat u wijzigingen aan het circuit aanbrengt of componenten aansluit of verwijdert. Opmerking: Let op tekenen van elektrische schade, zoals rook, ongebruikelijke geuren of verkleuring. Actie: Als dergelijke tekenen zich voordoen, schakel dan onmiddellijk de stroom uit en inspecteer het circuit grondig op fouten. Het display kan tijdens het gebruik warmte genereren, wat bij onjuist gebruik tot oververhitting, brandwonden of brand kan leiden. Houd rekening met de volgende veiligheidsinstructies: Let op: Sommige onderdelen van het display kunnen tijdens het gebruik of bij een storing warm worden. Maatregel: Laat het display na het uitschakelen voldoende afkoelen voordat u de afzonderlijke componenten aan de achterkant direct aanraakt. Vermijd direct contact met hete componenten. Let op: Overbelasting kan overmatige verhitting van de elektronische componenten veroorzaken. Let op: Zorg ervoor dat de voeding en spanning voldoen aan de specificaties van het display en geen overbelasting veroorzaken.

Inhoudsopgave

Inleiding 3
Specificaties 4
Het I2C-adres van het OLED-scherm 5
De pin-out 6
Hoe Arduino IDE ..... 7 instellen
De Raspberry Pi en Python instellen 11
Het scherm aansluiten op de microcontroller 12
Bibliotheek voor Arduino IDE 13
Schetsvoorbeeld 14
Het scherm verbinden met Raspberry Pi 25
De I2C-interface inschakelen 26
I2C-adressen vinden 26
Python-bibliotheek 27
Python-script 28

Inleiding

OLED staat voor Organic Light Emitting Diodes (Organische lichtemitterende diodes). OLED schermen zijn arrays van LED's die in een matrix op elkaar zijn gestapeld. Het 0,96 inch OLED scherm heeft 128x64 pixels (LED's). Om deze LED's aan te sturen hebben we een drivercircuit of chip nodig. Het scherm heeft een stuurchip met de naam *SSD1306*. De driverchip heeft een I2C-interface voor communicatie met de hoofdmicrocontroller. Het I2C adres van een driver chip is voorgedefinieerd, met waarde *0x3C.*

Het OLED scherm en de SSD1306 driver chip werken in het *3,3V* bereik. Maar er is een *3,3V* spanningsregelaar ingebouwd, wat betekent dat deze schermen in het *5V* bereik kunnen werken.

De prestaties van deze schermen zijn veel beter dan die van traditionele LCD's. De eenvoudige I2C-communicatie en het lage stroomverbruik maken ze geschikter voor allerlei toepassingen.

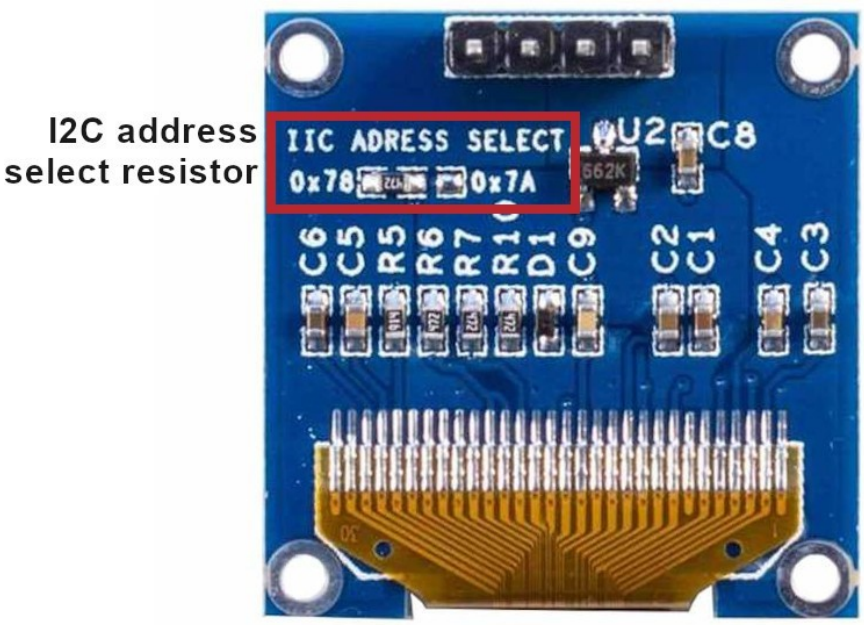
Specificaties

" Voedingsspanning:	van 3,3V tot 5V DC
" Communicatie-interface:	I2C
" Kleur pixel:	Wit
" Bedrijfstemperatuur:	-20 tot 70 °C
" Laag	stroomverbruik, minder dan 11 mA
"	Afmetingen28 x 33 x 4 mm [1,1 x 1,3 x 0,13 in].

Om de levensduur van het scherm te verlengen, is het gebruikelijk om een "Screen saver" te gebruiken. Het wordt aanbevolen om niet gedurende een lange periode constante informatie te gebruiken omdat dit de levensduur van het scherm verkort en het zogenaamde "Screen burn" effect vergroot.

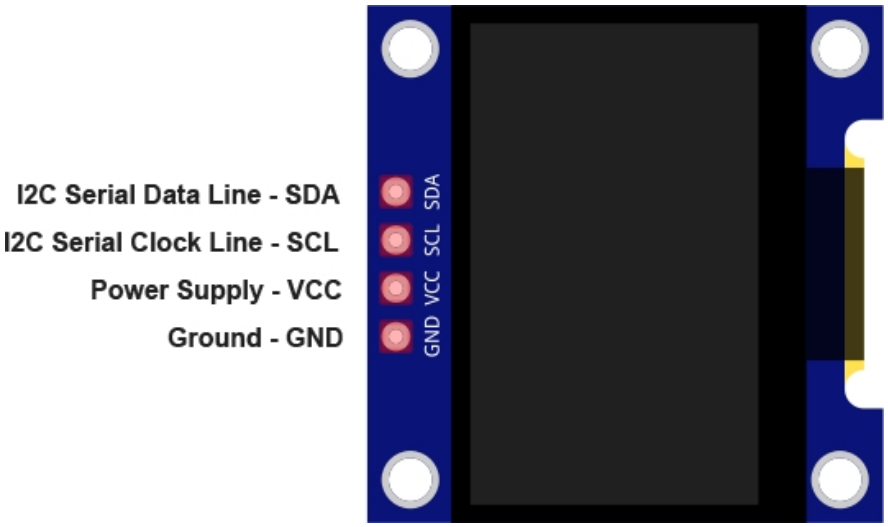
Het I2C-adres van het OLED-scherm

Alle OLED schermen die AZ-Delivery levert hebben hetzelfde seriële adres, `0x3C`. Het 0.96 inch scherm biedt de mogelijkheid om een weerstand op de achterkant van de printplaat opnieuw te solderen voor een ander adres, `0x3D`, maar dat wordt niet aanbevolen. Als er meerdere schermen met één I2C-interface moeten worden gebruikt, is het aan te raden om een I2C-multiplexer te gebruiken of gewoon een groter scherm.



De pin-out

Het 0,96 inch OLED-scherm heeft vier pinnen. De pin-out wordt op de volgende afbeelding getoond:



Het scherm heeft een *3,3V* spanningsregelaar aan boord. De pennen van het 0,96 inch OLED scherm kunnen worden aangesloten op *3,3V* of *5V* voeding zonder gevaar voor het scherm zelf.

OPMERKING: Bij gebruik van de Raspberry Pi mag de voeding alleen afkomstig zijn van

3,3V pin.

Arduino IDE instellen

Als de Arduino IDE niet is geïnstalleerd, volg dan de *link* en download het installatiebestand voor het besturingssysteem van je keuze.



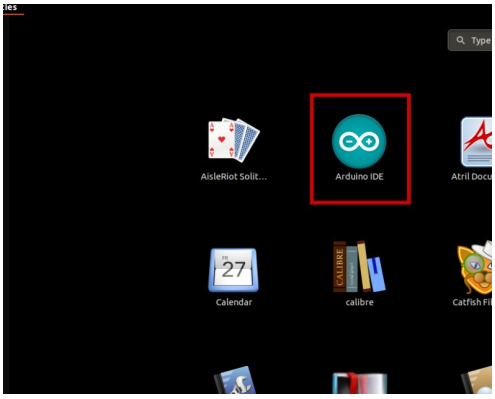
Windows-gebruikers dubbelklikken op het gedownloade exe-bestand en volgen de instructies in het installatievenster.

Voor *Linux* gebruikers, download een bestand met de extensie *.tar.xz*, dat uitgepakt moet worden. Als het is uitgepakt, ga dan naar de uitgepakte map en open de terminal in die map. Twee *.sh* scripts moeten worden uitgevoerd, het eerste heet *arduino-linux-setup.sh* en het tweede heet *install.sh*.

Om het eerste script in de terminal uit te voeren, open je de terminal in de uitgepakte map en voer je het volgende commando uit:
sh arduino-linux-setup.sh gebruiker_naam
user_name - is de naam van een superuser in het Linux besturingssysteem. Een wachtwoord voor de superuser moet worden ingevoerd wanneer het commando wordt gestart. Wacht een paar minuten tot het script alles heeft voltooid.

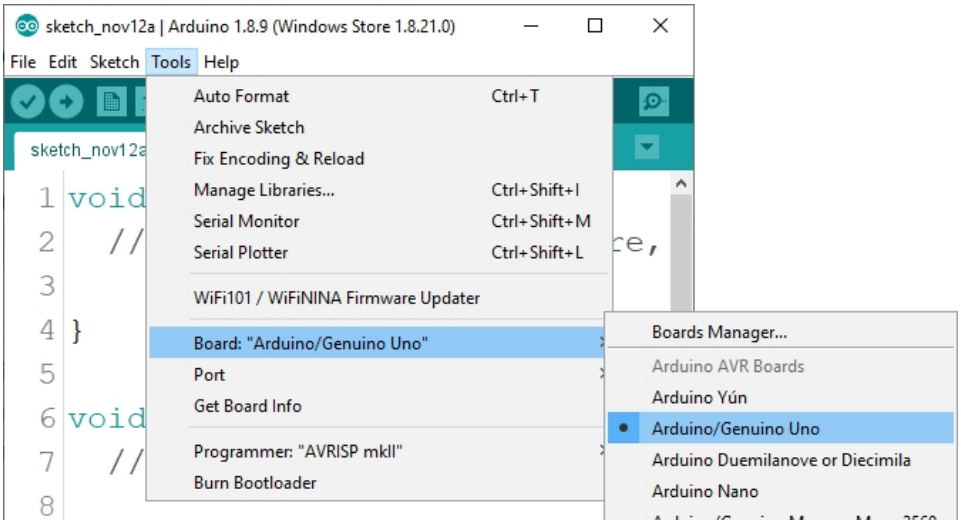
Het tweede script, *install.sh*, moet worden gebruikt na de installatie van het eerste script. Voer het volgende commando uit in de terminal (uitgepakte map): **sh install.sh**

Ga na de installatie van deze scripts naar de *All Apps*, waar de *Arduino IDE* is geïnstalleerd.



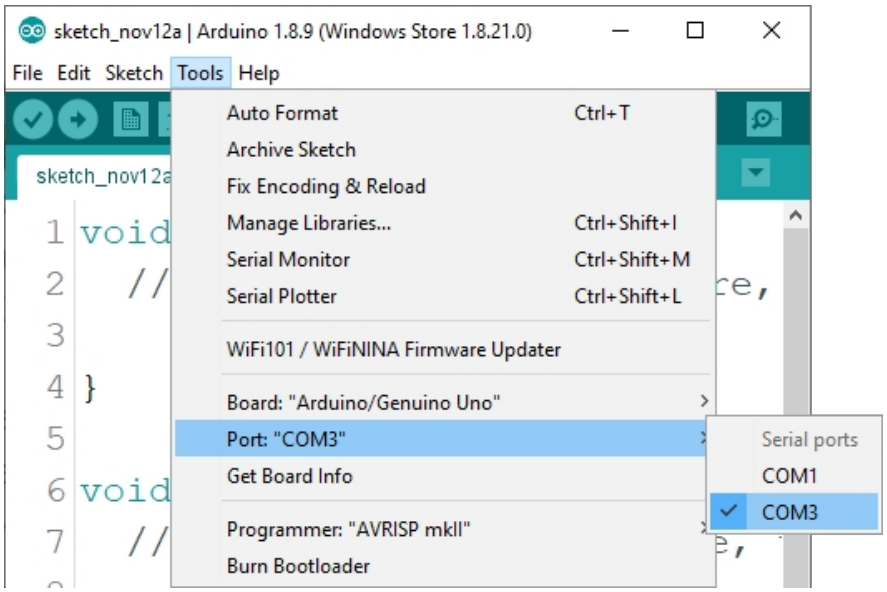
Bijna alle besturingssystemen worden geleverd met een vooraf geïnstalleerde teksteditor (*Windows* wordt bijvoorbeeld geleverd met *Kladblok*, *Linux Ubuntu* wordt geleverd met *Gedit*, *Linux Raspbian* wordt geleverd met *Leafpad*, enz.) Al deze teksteditors zijn prima geschikt voor het doel van dit eBook.

Vervolgens moet je controleren of je PC een microcontrollerbord kan detecteren. Open de pas geïnstalleerde Arduino IDE en ga naar:
Tools > Board > {uw boardnaam hier}
{uw boardnaam hier} moet de *Arduino/Genuino Uno* zijn, zoals te zien is op de volgende afbeelding:



De poort waarop het microcontrollerbord is aangesloten moet worden geselecteerd. Ga naar: *Extra > Poort > {poortnaam hoort hier}* en wanneer het microcontrollerbord is aangesloten op de USB-poort, is de poortnaam te zien in het vervolgkeuzemenu op de vorige afbeelding.

Als de Arduino IDE op Windows wordt gebruikt, zijn de poortnamen als volgt:



Voor Linux-gebruikers is de poortnaam bijvoorbeeld `/dev/ttyUSBx`, waarbij `x`

staat voor een geheel getal tussen *0* en *9*.

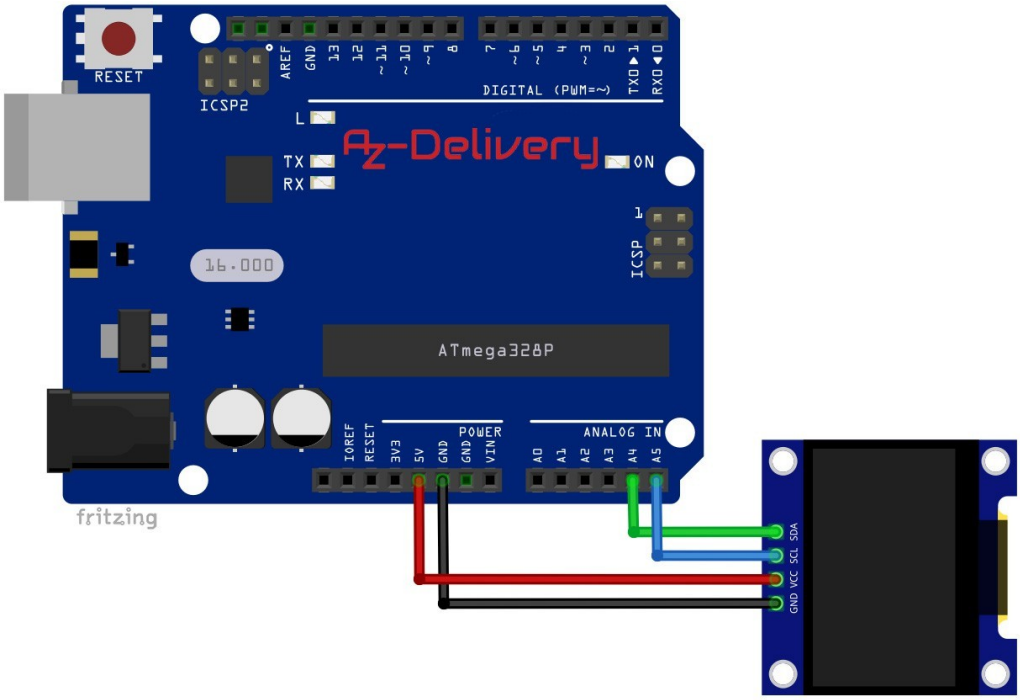
De Raspberry Pi en Python instellen

Voor de Raspberry Pi moet eerst het besturingssysteem worden geïnstalleerd, daarna moet alles zo worden ingesteld dat hij in de Headless-modus kan worden gebruikt. De Headless-modus maakt het mogelijk om op afstand verbinding te maken met de Raspberry Pi, zonder dat er een PC-schermmonitor, muis of toetsenbord nodig is. Het enige dat in deze modus gebruikt wordt zijn de Raspberry Pi zelf, de voeding en de internetverbinding. Dit wordt allemaal minutieus uitgelegd in het gratis eBook: *Raspberry Pi Snelle Opstartgids*

Het besturingssysteem *Raspbian* wordt geleverd met *Python* voorgeïnstalleerd.

Het scherm verbinden met de microcontroller

Sluit het 0,96 inch OLED-scherm aan op de microcontroller zoals aangegeven in het volgende aansluitschema:



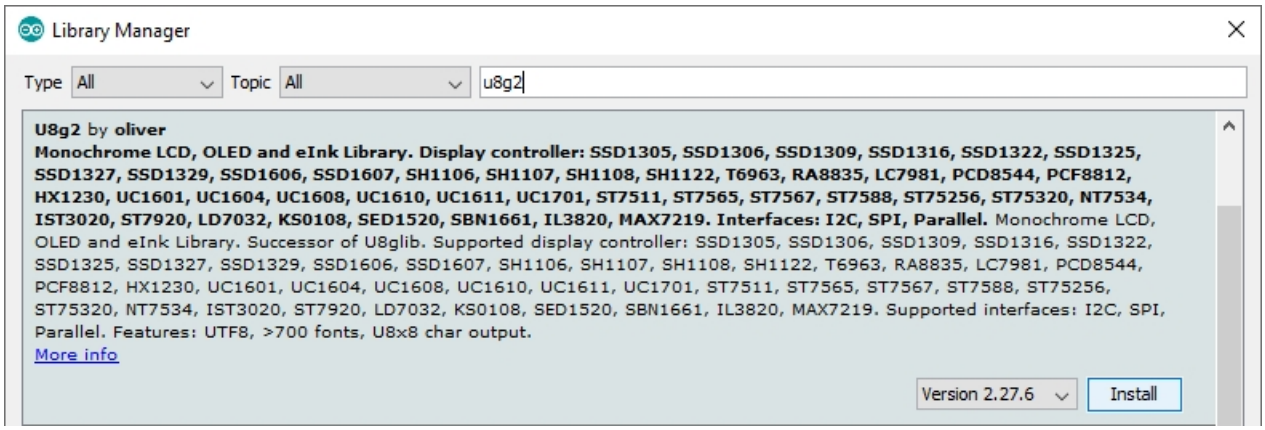
Schermpen	MC-pen	Draadkleur
SDA	A4	**Blauwe draad**
SCL	A5	**Groene draad**
VCC	5V	**Rode draad**
GND	GND	**Zwarte draad**

Bibliotheek voor Arduino IDE

Om het scherm te gebruiken met de microcontroller, is het aanbevolen om een externe bibliotheek ervoor te downloaden. De bibliotheek die in dit eBook wordt gebruikt heet de *U8g2*. Om deze te downloaden en te installeren, open je Arduino IDE en ga je naar:

Extra > Bibliotheken beheren.

Wanneer een nieuw venster wordt geopend, typt *u u8g2* in het zoekvak en installeert u de bibliotheek *U8g2* gemaakt door *oliver*, zoals weergegeven in de volgende afbeelding:



Bij de bibliotheek worden verschillende schetsvoorbeelden geleverd. Ga naar om er een te openen:

Bestand > Voorbeelden > U8g2 > full_buffer > GraphicsTest

Met dit schetsvoorbeeld kan het scherm worden getest. In dit eBook wordt de code van de schets aangepast in oreder om een beginnersvriendelijkere versie van de code te maken.

Schetsvoorbeeld

```
#include <U8g2lib.h>
#include <Wire.h>
#define time_delay 2000
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);

const char COPYRIGHT_SYMBOL[] = {0xa9, '\0'};
void u8g2_prepare() {
  u8g2.setFont(u8g2_font_6x10_tf);
  u8g2.setFontRefHeightExtendedText();
  u8g2.setDrawColor(1);
  u8g2.setFontPosTop();
  u8g2.setFontDirection(0);
}
void u8g2_box_frame() {
  u8g2.drawStr(0, 0, "drawBox");
  u8g2.drawBox(5, 10, 20, 10);
  u8g2.drawStr(60, 0, "drawFrame");
  u8g2.drawFrame(65, 10, 20, 10);
}
void u8g2_r_frame_box() {
  u8g2.drawStr(0, 0, "drawRFrame");
  u8g2.drawRFrame(5, 10, 40, 15, 3);
  u8g2.drawStr(70, 0, "drawRBox");
  u8g2.drawRBox(70, 10, 25, 15, 3);
}
void u8g2_disc_circle() {
  u8g2.drawStr(0, 0, "drawDisc");
  u8g2.drawDisc(10, 18, 9);
  u8g2.drawStr(60, 0, "drawCircle");
  u8g2.drawCircle(70, 18, 9);
}
```

```
void u8g2_string_orientation() {
  u8g2.setFontDirection(0);
  u8g2.drawStr(5, 15, "0");
  u8g2.setFontDirection(3);
  u8g2.drawStr(40, 25, "90");
  u8g2.setFontDirection(2);
  u8g2.drawStr(75, 15, "180");
  u8g2.setFontDirection(1);
  u8g2.drawStr(100, 10, "270");
}
void u8g2_line() {
  u8g2.drawStr(0, 0, "drawLine");
  u8g2.drawLine(7, 20, 77, 32);
}
void u8g2_triangle() {
  u8g2.drawStr(0, 0, "drawTriangle");
  u8g2.drawTriangle(14, 20, 45, 30, 10, 32);
}
void u8g2_unicode() {
  u8g2.drawStr(0, 0, "Unicode");
  u8g2.setFont(u8g2_font_unifont_t_symbols);
  u8g2.setFontPosTop();
  u8g2.setFontDirection(0);
  u8g2.drawUTF8(10, 20, "☀");
  u8g2.drawUTF8(30, 20, "☁");
  u8g2.drawUTF8(50, 20, "☂");
  u8g2.drawUTF8(70, 20, "☔");
  u8g2.drawUTF8(95, 20, COPYRIGHT_SYMBOL); //COPYRIGHT SIMBOL
  u8g2.drawUTF8(115, 15, "\xb0"); // GRADSYMBOOL
}
```

```
#define image_width 128
#define image_height 21
static const unsigned char image_bits[] U8X8_PROGMEM = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x06, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0xfc, 0x1f, 0x00, 0x00,
  0xfc, 0x1f, 0x00, 0x00, 0x06, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0xfe,
  0x1f, 0x00, 0xfc, 0x7f, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x07, 0x18, 0x00, 0x00, 0x0c, 0x60, 0x00, 0x00,
  0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x18, 0x00, 0x00,
  0x0c, 0xc0, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x03, 0x18, 0x00, 0x00, 0x0c, 0xc0, 0xf0, 0x1f, 0x06, 0x63, 0x80, 0xf1,
  0x1f, 0xfc, 0x33, 0xc0, 0x03, 0x18, 0x00, 0x00, 0x0c, 0xf8, 0x3f, 0x06,
  0x63, 0xc0, 0xf9, 0x3f, 0xfe, 0x33, 0xc0, 0x03, 0x18, 0x00, 0x0c, 0x0c,
  0x18, 0x30, 0x06, 0x63, 0xc0, 0x18, 0x30, 0x06, 0x30, 0xc0,
  0xff, 0xff, 0xdf, 0xff, 0x0c, 0xc0, 0x18, 0x30, 0x06, 0x63, 0xe0, 0x18,
  0x30, 0x06, 0x30, 0xc0, 0xff, 0xff, 0xdf, 0xff, 0x0c, 0xc0, 0x98, 0x3f,
  0x06, 0x63, 0x60, 0x98, 0x3f, 0x06, 0x30, 0xc0, 0x03, 0x18, 0x0c, 0x00,
  0x0c, 0xc0, 0x98, 0x1f, 0x06, 0x63, 0x70, 0x98, 0x1f, 0x06, 0x30, 0xc0,
  0x03, 0x18, 0x06, 0x00, 0x0c, 0xc0, 0x18, 0x00, 0x06, 0x63, 0x38, 0x18,
  0x00, 0x06, 0x30, 0xc0, 0x03, 0x18, 0x03, 0x00, 0x0c, 0xe0, 0x18, 0x00,
  0x06, 0x63, 0x1c, 0x18, 0x00, 0x06, 0x30, 0xc0, 0x00, 0x80, 0x01, 0x00,
  0xfc, 0x7f, 0xf8, 0x07, 0x1e, 0xe3, 0x0f, 0xf8, 0x07, 0x06, 0xf0, 0xcf,
  0x00, 0xc0, 0x00, 0xfc, 0x3f, 0xf0, 0x07, 0x1c, 0xe3, 0x07, 0xf0, 0x07,
  0x06, 0xe0, 0xcf, 0x00, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xc0, 0x00, 0x30, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xc0,
  0x00, 0x18, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0xe0, 0x00, 0xfc, 0x1f, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x00, 0xfc, 0x1f, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f };
```

```
void u8g2_bitmap() {
  u8g2.drawXBMP(0, 5, image_width, image_height, image_bits);
}
void setup(void) {
  u8g2.begin();
  u8g2_prepare();
}
float i = 0.0;
void loop(void) {
  u8g2.clearBuffer();
  u8g2_prepare();
  u8g2_box_frame();
  u8g2.sendBuffer();
  delay(time_delay);

  u8g2.clearBuffer();
  u8g2_disc_circle();
  u8g2.sendBuffer();
  delay(time_delay);

  u8g2.clearBuffer();
  u8g2_r_frame_box();
  u8g2.sendBuffer();
  delay(time_delay);

  u8g2.clearBuffer();
  u8g2_prepare();
  u8g2_string_orientation();
  u8g2.sendBuffer();
  delay(time_delay);

  u8g2.clearBuffer();
  u8g2_line();
  u8g2.sendBuffer();
  delay(time_delay);
```

```
  // een tab
  u8g2.clearBuffer();
  u8g2_triangle();
  u8g2.sendBuffer();
  delay(time_delay);

  u8g2.clearBuffer();
  u8g2_prepare();
  u8g2_unicode();
  u8g2.sendBuffer();
  delay(time_delay);

  u8g2.clearBuffer();
  u8g2_bitmap();
  u8g2.sendBuffer();
  delay(time_delay);

  u8g2.clearBuffer();
  u8g2.setCursor(0, 0);
  u8g2.print(i);
  i = i + 1,5;
  u8g2.sendBuffer();
  delay(time_delay);
}
```

Aan het begin van de sketch worden twee bibliotheken geïmporteerd: *U8g2lib* en

Draad.

Vervolgens wordt het object *u8g2* gemaakt, met de volgende regel code:

```
U8G2_SSD1306_128X64_UNIVISION_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);
```

Het gemaakte object vertegenwoordigt het scherm zelf en wordt gebruikt om het scherm te besturen. De U8g2-bibliotheek kan voor veel andere OLED-schermen worden gebruikt, dus er zijn veel constructeurs in de schetsvoorbeelden van de bibliotheek.

Daarna *wordt* de functie *u8g2_prepare()* gemaakt, die geen argumenten heeft en geen waarde teruggeeft. In deze functie worden vijf u8g2-bibliotheekfuncties gebruikt.

De eerste functie heet *setFont()*, die één argument heeft en geen waarde teruggeeft. Het argument vertegenwoordigt het lettertype *u8g2*. De lijst met beschikbare lettertypen kan worden gevonden op de volgende *link*.

De tweede functie heet *setFontRefHeightExtendedText()*, die geen argumenten heeft en geen waarde teruggeeft. Het wordt gebruikt voor het tekenen van tekens op het scherm. Meer gedetailleerde uitleg over deze functie kun je vinden op de volgende *link*.

De derde functie heet `setDrawColor()`, die één argument heeft en geen waarde teruggeeft. De waarde van het argument is een geheel getal dat een kleurindex voor alle tekenfuncties vertegenwoordigt. Tekenprocedures voor lettertypen gebruiken dit argument om de voorgrondkleur in te stellen. De standaardwaarde is *1.* Als het wordt ingesteld op *0,* dan wordt de ruimte rond het teken verlicht en het teken niet. Argumentwaarde *2* kan ook worden gebruikt, maar er is geen verschil met *0*.

De vierde functie heet `setFontPosTop()`, die geen argument heeft en geen waarde teruggeeft. Deze functie bepaalt de tekenpositie op één regel van de tekst. De functie heeft een paar versies. De eerste is `setFontPosBaseLine(), de` tweede is `setFontPosCenter().` De derde is `setFontPosBottom()` en hun doel is om de positie van de tekens op één regel te wijzigen.

De vijfde functie heet `setFontDirection()`, die één argument heeft en geen waarde teruggeeft. Het argument is een geheel getal dat de richting van de tekst weergeeft. De waarde is een geheel getal in het bereik van *0* tot *3*, waarbij *0 = 0°*, *1 = 90°*, *2 = 180°* en *3 = 270°*.

De functie `drawStr()` heeft drie argumenten en geeft geen waarde terug. Het wordt gebruikt om een constante tekenreeks op het scherm weer te geven. De eerste twee argumenten vertegenwoordigen de *X*- en Y-positie van de cursor, waar de tekst wordt weergegeven. Het derde argument vertegenwoordigt de tekst zelf, een constante tekenreekswaarde. De functies die de lay-out van de tekst instellen voordat je de functie `drawStr()` gebruikt, moeten worden gebruikt, anders gebruikt de functie `drawStr()` standaardinstellingen voor het lettertype, de grootte en de algemene lay-out van de tekst.

Om vormen weer te geven, worden specifieke functies voor elke vorm gebruikt:

De functie `drawFrame()` heeft vier argumenten en geeft geen waarde terug. Het wordt gebruikt om een frame weer te geven, een lege rechthoek. De eerste twee argumenten vertegenwoordigen de *X*- en Y-positie van de linkerbovenhoek van het frame. Het derde argument vertegenwoordigt de breedte van het frame en het vierde argument vertegenwoordigt de hoogte van het frame.

De functie `drawRFrame()` heeft vijf argumenten en geeft geen waarde terug. Het wordt gebruikt om een frame met afgeronde hoeken weer te geven. De eerste twee argumenten vertegenwoordigen de *X*- en Y-positie van de linkerbovenhoek van het frame. De tweede twee argumenten vertegenwoordigen de breedte en hoogte van het frame en het vijfde argument vertegenwoordigt de straal van de hoek.

De functie `drawBox()` heeft vier argumenten en geeft geen waarde terug. Het wordt gebruikt om een gevulde rechthoek weer te geven. De eerste twee argumenten vertegenwoordigen de *X*- en Y-positie van de linkerbovenhoek van de rechthoek. De tweede twee argumenten vertegenwoordigen respectievelijk de breedte en de hoogte van de rechthoek.

De functie `drawRBox()` heeft vijf argumenten en geeft geen waarde terug. Het wordt gebruikt om een gevulde rechthoek met afgeronde hoeken weer te geven. De eerste twee argumenten vertegenwoordigen de *X*- en Y-positie van de linkerbovenhoek van de rechthoek. De tweede twee argumenten vertegenwoordigen respectievelijk de breedte en de hoogte van de rechthoek. Het vijfde argument stelt de hoekradius voor.

De functie `drawCircle()` heeft drie argumenten en geeft geen waarde terug. Het wordt gebruikt om een cirkel weer te geven. De eerste twee argumenten vertegenwoordigen de *X*- en Y-posities van het middelpunt van de cirkel. Het derde argument vertegenwoordigt de straal van de cirkel.

De functie `drawDisc()` heeft drie argumenten en geeft geen waarde terug. Het wordt gebruikt om een schijf weer te geven. De eerste twee argumenten vertegenwoordigen de *X*- en Y-positie van het middelpunt van de schijf. Het derde argument vertegenwoordigt de straal van de schijf.

De functie `drawTriangle()` heeft zes argumenten en geeft geen waarde terug. Het wordt gebruikt om een gevulde driehoek weer te geven. De eerste twee argumenten vertegenwoordigen de *X*- en Y-positie van het eerste hoekpunt van de driehoek. De tweede twee argumenten vertegenwoordigen de *X*- en Y-positie van het tweede hoekpunt van de driehoek. De laatste twee argumenten vertegenwoordigen de X- en Y-posities van het laatste hoekpunt van de driehoek.

De functie `drawLine()` heeft vier argumenten en geeft geen waarde terug. Het wordt gebruikt om een lijn weer te geven. De eerste twee argumenten vertegenwoordigen de *X*- en Y-positie van het beginpunt van de lijn. De tweede twee argumenten vertegenwoordigen de *X*- en Y-positie van het eindpunt van de lijn.

De functie `drawUTF8()` heeft drie argumenten en geeft a waarde terug. Het wordt gebruikt om een tekst weer te geven, de stringwaarde die een teken kan bevatten dat is gecodeerd als een Unicode-teken. De eerste twee argumenten vertegenwoordigen de *X*- en Y-positie van de cursor en het derde argument vertegenwoordigt de tekst zelf. De Unicode-tekens kunnen op een paar manieren worden weergegeven. De eerste is om
kopieer en plak het bestaande teken in de schets, zoals in de volgende regel van de code: `u8g2.drawUTF8(50, 20, "☂")`
De tweede is het maken van een `char` array, die twee waarden heeft: de eerste waarde
is een hexadecimaal nummer van het `Unicode` karakter en de tweede waarde is een null karakter ("`\0`"). Dit kan worden gedaan door gebruik te

maken van de *char* array genaamd *COPYRIGHT_SYMBOL*, zoals in de volgende regels van de code:

```
const char COPYRIGHT_SYMBOL[] = {0xa9, '^0'}
u8g2.drawUTF8(95, 20, COPYRIGHT_SYMBOL); //COPYRIGHT SYMBOL
```

De derde manier om de functie te gebruiken is door een hexadecimaal getal te gebruiken voor het teken zelf, zoals in de volgende regel code:

```
u8g2.drawUTF8(115, 15, "\xb0"); // GRADSYMBOOL
```

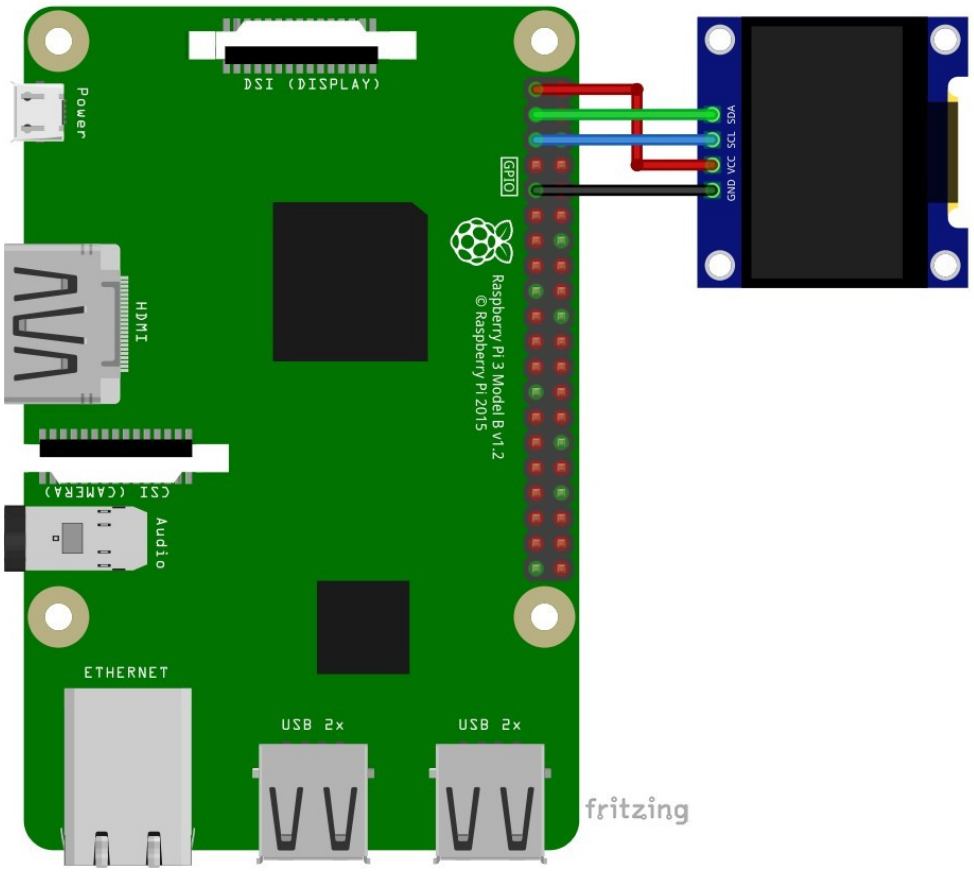
De functie retourneert een waarde, een geheel getal dat de breedte van de tekst (*string*) vertegenwoordigt.

Om iets op het scherm weer te geven, moet de gegevensbuffer van het scherm eerst worden leeggemaakt, daarna wordt een nieuwe waarde ingesteld (een afbeelding) voor de gegevensbuffer en vervolgens wordt een nieuwe waarde van de gegevensbuffer naar het scherm gestuurd. Op deze manier wordt een nieuw beeld op het scherm weergegeven. Om deze verandering te zien, moet de functie *delay()* worden gebruikt om de volgende verandering van de databuffer te verschuiven, zoals in de volgende regels code:

```
u8g2.clearBuffer();
u8g2_bitmap(); // de gegevensbuffer instellen
u8g2.sendBuffer();
vertraging(time_delay);
```

Het scherm verbinden met Raspberry Pi

Verbind het scherm met de Raspberry Pi zoals aangegeven in het volgende aansluitschema:

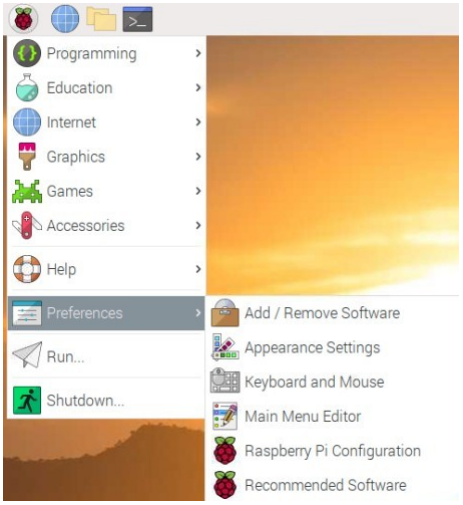


Schermpen	Raspberry Pi pin	Fysieke pin nr.	Draadkleur
SDA	GPIO2	3	**Groene draad**
SCL	GPIO3	5	**Blauwe draad**
VCC	3V3	1	**Rode draad**
GND	GND	9	**Zwarte draad**

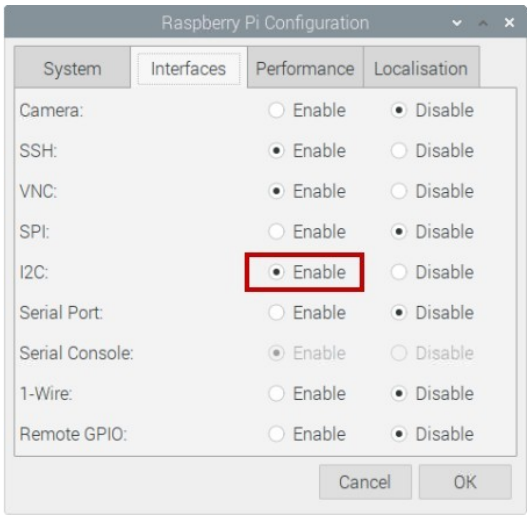
De I2C-interface inschakelen

Om het scherm met Raspberry Pi te kunnen gebruiken, moet de I2C-interface w o r d e n ingeschakeld. Open het volgende menu:

Menu Toepassing > Voorkeuren > Raspberry Pi Configuratie



Schakel in het nieuwe venster, onder de tab *Interfaces*, het keuzerondje I2C in, zoals op de volgende afbeelding:



Het adres van de OLED module vinden

Als het is ingeschakeld, gebruiken we het commando i2detect om de module op de I2C-bus te vinden: **i2cdetect -y 1**
Het resultaat zou er uit moeten zien zoals hieronder:

```
pi@rpispy:~ $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- 3c -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Ons apparaat werd herkend met het adres "0x3c". Dit is het standaard adres voor dit type apparaat.

Python-bibliotheek

Voor de weergave van vormen, tekst en afbeeldingen gebruiken we een Python-bibliotheek. Op het huidige Raspberry Pi OS zijn Python3, pip3 en git al voorgeïnstalleerd, maar als dat niet het geval is, kun je het geheel installeren met de volgende commando's:

```
sudo apt-get install python3-dev libffi-dev libssl-dev
python3-pil libjpeg-dev zlib1g-dev libfreetype6-dev
liblcms2-dev libopenjp2-7 libtiff5 -y
```

```
sudo apt-get install python3-rpi.gpio python3-pip -y
```

```
sudo apt-get install git -y
```

We gebruiken de "luma.oled" bibliotheek, die kan worden geïnstalleerd met het volgende commando:

sudo -H pip3 installeer luma.oled

Python- Script

In de map "pi" maken we nu een map genaamd "oled" en gaan naar deze map

```
sudo mkdir oled
cd oled
```

luma.oled biedt veel voorbeelden en die kunnen we downloaden met het volgende commando:

```
sudo git kloon https://github.com/rm-hull/luma.examples
```

met:

cd luma.voorbeelden/voorbeelden/

gaan we naar de map waarin de voorbeelden staan. en met:

python3 demo.py

kunnen we een van de voorbeelden starten. Als er een witte ruis op het display verschijnt, moet de juiste regelaar worden doorgegeven. dit kan worden gedaan met:

```
python3 demo.py --apparaat [controller]
```

luma.oled gebruikt standaard de SSD1306, als je bijvoorbeeld SH1106 hebt, zou het opstarten van het script er als volgt uitzien:

```
python3 demo.py --apparaat ssh1106
```

Er staan meer voorbeelden in de map.

Met het commando:

ls -a

kun je ze zien.

Probeer meer

voorbeelden. En met:

sudo nano [examplescript]

de scripts kunnen worden bewerkt.

Nu is het tijd om te leren en zelf projecten te maken. Je kunt dat doen met behulp van veel voorbeeldscripts en andere tutorials die je op internet kunt vinden.

Als je op zoek bent naar producten van hoge kwaliteit voor Raspberry Pi, dan ben je bij AZ- Delivery Vertriebs GmbH aan het juiste adres. U krijgt talloze toepassingsvoorbeelden, volledige installatiegidsen, eBooks, bibliotheken en hulp van onze technische experts.

https://az-delivery.de

Veel plezier!

Impressum

https://az-delivery.de/pages/about-us