



Arduino -6- I2C: het protocol en LCD

Handleiding van Helpmij.nl

Auteur: drejansen

februari 2022

“ Dé grootste en gratis computerhelpdesk van Nederland ”



Arduino en I²C: het protocol en LCD

Er zijn heel veel sensoren en actuatoren die een I²C bus aansluiting hebben. Geen I²C bus? dan kan je het apparaat dmv een interfase toch op I²C aansluiten.



Onderstaande schrijf ik met grote dank aan Karel Dupain & Joep Suijs.

I²C bus ook wel IIC bus genoemd = Inter Integrated Circuit bus. Max transmissie snelheid 100 Kbit/sec Alleen geschikt voor kleine afstanden.

Het protocol

Een protocol is een reeks afspraken over de communicatie. Er is gebruikelijk één master en meerdere slaves op de bus aanwezig. De master roept een slave aan, die (indien aanwezig) met een ACK(noledge) signaal reageert. Een slave kan zelf ook een oproep naar de master plaatsen. Er zijn twee datalijnen, SDA (serial data, pootje A4) en SCL (serial clock, pootje A5). Er zijn geen Pull-up weerstanden (Gebruik 10 Kohm). Als een van de devices op de bus is voorzien van pull-up's, dan hoeft je er zelf geen toe te voegen. Arduino zelf heeft geen pull-up weerstanden op de bus. De reden is dat de pootjes ook als hoogohmige analoge ingang kunnen worden gebruikt.

Bij data transport wordt het meest hoogstwaardige bit (het MSB) als eerste verzonden. Na 8 ontvangen bits, reageert de geadresseerde slave met een acknowledge bit, het 9e bit.

Alle uitgangen zijn een wired AND functie, alle ingangen een wired OR. Een 'wired' component kan geen 'hoog' signaal uitsturen, daarvoor dient de pull-up weerstand. (Dit wordt vaak met een * aangegeven.)

Adressering is belangrijk, want je moet het juiste apparaat aanroepen. Een adres is 7 bits lang, waarbij het MSB als eerste wordt verzonden. Het 8e bit is lezen (1) of schrijven (0). 10 bits adressering is mogelijk, maar moeilijk. Zoals in hoofdstuk -1- is gezegd, ik hou niet van moeilijk, het is zo al lastig genoeg! Met dat 8e bit wil de master aangeven: 'ik ga jou wat sturen', of: 'ik wil van jou wat weten'. (BV een motor aanzetten, of een temperatuur uitlezen.) Bij een oproep luisteren alle slaves, de geadresseerde slave is de enige die reageert met een acknowledge signaal. De overige slaves schakelen af. Wordt er een 'onbewoond' adres aangeropen, dan reageert niemand en volgt een foutmelding. Er is nog veel meer over I2C te vertellen, maar laten we ons daar niet mee vermoeien. (Mogelijk komt dat op het einde van de reeks nog aan bod.)

LCDisplay

Een LCDisplay met I²C printje koop je [hier](#) voor € 5,- .
(Tot zover deze Ster spot)



I²C Voorbeeld: Knipperende tekst: Hello World!

Belangrijk: er kan slechts één slave tegelijk met de master communiceren.

Het display heeft 4 aansluitingen +5V, Massa, SDA (datapootje A4) en SCL (clockpootje A5) Het backlight bedien je met het brugje op de I2C print.

Klein testprogrammaatje:

helaas willen bepaalde tekens (kleiner dan en groter dan) niet meewerken, de tekst daartussen word niet weergegeven, dus het programma als plaatje ingevoegd. helaas moet je dat dan overtypen.

```
#include <Wire.h> // nodig voor het I2C protocol
#include <LiquidCrystal_I2C.h> // de LCDDisplay library (subroutine)
LiquidCrystal_I2C lcd(0x27, 16, 2); // het adres (0x27), het type display (16 karakters en
twee regels)

void setup() {
  lcd.begin(); // starten van de LCD routing
  lcd.setCursor(0, 0); // zet de cursor links bovenaan
  lcd.print("Hello World!"); // tekst naar het LCD zenden.
}

void loop() {
  lcd.noDisplay(); // Display uit
  delay(500); // korte pauze
  lcd.display(); // Display aan
  delay(500); // korte pauze
}
```

Voor veel I²C apparatuur is #include Wire.h noodzakelijk, maar hier, bij het gebruik van #include LiquidCrystal_I2C.h, is dat niet nodig, het I2C gebeuren zit al in de LCD routine verwerkt.

Slave adres

Maar hoe kom je nou achter het juiste adres van het LCD? Dat staat dat in de datasheet! Vaak kan je zelf het slave adres binnen grenzen aanpassen. Dat kan met dipswitches, brugklemmetjes of soldeerpad's. Op het printje onder de blauwe potmeter zie je drie groepjes van 2 eilandjes. zo kan je 8 verschillende display's onafhankelijk van elkaar benaderen. Met de potmeter stel je de zichtbaarheid van de tekst (eenmalig) in.

Met onderstaande schets vind je het actuele adres. (Dit kan je wel kopiëren en plakken in de Arduino IDE)

```
#include ..... // hier moet je zelf het ontbrekende 'toverwoord' met haken invullen.
// zie hierboven. Let op het gebruik van hoofdletters!

void setup() {
  Wire.begin();
  Serial.begin(19200);
  while (!Serial);
  Serial.println("\nI2C Scanner");
}

void loop() {
  int apparaten = 0;
  Serial.println("Scanning...");
  for (byte adres = 1; adres < 127; ++adres) {
```

```
// de scanner maakt gebruik van het retour signaal het einde transmissie
// commando. Als het apparaat positief reageert, is dat een geldig adres.
Wire.beginTransmission(adres);
byte error = Wire.endTransmission();
if (error == 0) {
  Serial.print("I2C apparaat gevonden op adres: 0x");
  if (adres < 16) {
    Serial.print("0");
  }
  Serial.print(adres, HEX);
  Serial.println(" !");
  ++apparaten;
} else if (error == 4) {
  Serial.print("onbekende foutmelding op adres 0x");
  if (adres < 16) {
    Serial.print("0");
  }
  Serial.println(adres, HEX);
}
}
if (apparaten == 0) {
  Serial.println("Geen I2C apparaten gevonden\n");
} else {
  Serial.println("klaar\n");
}
delay(5000); // Wait 5 seconden voor volgende scan
}
```

Tot slot wens ik u allen een inventief, creatief, maar vooral gezond 2022!