

## Bouw je eigen weerstation: deel 3

### Inhoud deel 3

In deel 3 gaan we het volgende doen:

- Een speciaal teken op het LCD maken; namelijk het graden teken (code weerstation\_3a)
- Een windsnelheidsmeter op de Arduino aansluiten (code weerstation\_3b)
- Het programma van in de Arduino wijzigen zodat het aflezen op het LCD veel prettiger wordt (code weerstation\_3c)

### Een speciaal teken op het LCD

Ga naar <https://omerk.github.io/lcdchargen/>

Op deze pagina kun je met de muis klikken op de groene velden en zo kies je afbeelding die je wilt weergeven op het LCD. De output zie je veranderen bij het kiezen van pixels. Als je klaar bent kopieer je de output in het grijze venster; die heb je dadelijk nodig in je programma. Hieronder het voorbeeld voor het maken van het graden teken.

**Pixels**

**Output** COPY

```
byte customChar[8] = {  
  0b00010,  
  0b00101,  
  0b00101,  
  0b00010,  
  0b00000,  
  0b00000,  
  0b00000,  
  0b00000  
};
```

### Programma voor het testen van het speciale teken

Het graden teken gaan we nu toevoegen aan het programma dat we eerder gemaakt hebben. Aan het programma voegen we nu toe:

```
Byte graden[8] = {  
  B00010,  
  B00101,  
  B00101,  
  B00010,  
  B00000,  
  B00000  
};
```

Let op: **customChar** is vervangen door een voor ons beter te begrijpen woord namelijk: **graden**.

**B en Ob** betekenen hetzelfde en geven aan dat de tekens die volgen als binair gelezen moeten worden.

In het programma moeten we ook nog aangeven waar we het nieuwe teken willen bewaren; dit doen we in setup met: `lcd.createChar(0, graden);` // het graden teken wordt op plaats 0 bewaard.

Als we het graden teken op het LCD willen schrijven doen we dat met: `lcd.write(0);` // en dus **niet** met `print!`

We nemen nu de code van weerstation 2c en voegen het speciale teken (het graden teken dus) toe.

### De code voor weerstation 3a

```
/*weerstation_3a V10
 * speciaal teken voor graden maken
 * luchtdruk, temperatuur en vochtigheid meten en de meetwaarden op het LCD plaatsen
 * I2C voor LCD en BMP 280 sensor
 *
 * LCD GND en BMP GND --> GND Arduino
 * LCD VCC en BMP Vin --> 5V Arduino
 * LCD SDA en BMP SDI --> A4 Arduino
 * LCD SCL en BMP SCK --> A5 Arduino
 *
 * I2C aansluitingen van Arduino Uno:A4 (SDA), A5 (SCL)
 *
 * DHT22 voor temperatuur en vochtigheid
 * ARDUINO pin 9 voor data VAN DHT22
 */

#include <Wire.h>           // Bibliotheek voor communicatie met I2C
#include <LiquidCrystal_I2C.h> // Bibliotheek voor het LCD scherm
#include <Adafruit_BMP280.h> // Bibliotheek voor de BMP 280 sensor
#include <DHT.h>           // DHT sensor bibliotheek toevoegen

LiquidCrystal_I2C lcd(0x27,20,4); // start LCD scherm en lcd is gebruiksnaam
Adafruit_BMP280 bmp;           // bmp is nu de gebruiksnaam voor de druk sensor
DHT sensor_dht(9,DHT22);      // onze sensor heet nu: sensor_dht

float luchtdruk, temperatuur, vochtigheid; // variabelen

byte graden[8] = {           // graden teken
  B00010,
  B00101,
  B00101,
  B00010,
  B00000,
  B00000,
  B00000,
  B00000
};

void setup() {
  lcd.init();                // initialiseer het LCD scherm
  lcd.backlight();          // zet de backlight aan
```

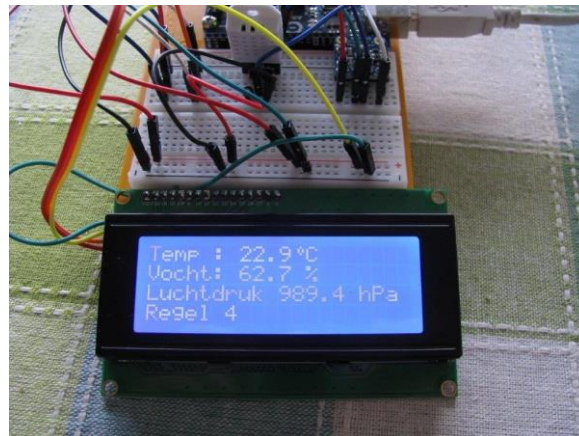
```

lcd.createChar(0,graden); // het symbool graden staat nu op positie 0
bmp.begin(); // start de bmp sensor
sensor_dht.begin(); // start de dht sensor
}

void loop() {
  delay(2000);
  lcd.clear(); // wis het scherm
  delay(500);
  //lees de temperatuur, vochtigheid en luchtdruk uit en bewaar die in de variabelen
  temperatuur = sensor_dht.readTemperature();
  vochtigheid = sensor_dht.readHumidity();
  luchtdruk = bmp.readPressure()/100.0;
  lcd.setCursor(0, 0); // zet de cursor op positie 1, regel 1
  lcd.print ("Temp : "); lcd.print(temperatuur,1); lcd.write(0); lcd.print ("C ");
  delay(500);
  lcd.setCursor(0, 1); // zet de cursor op positie 1, regel 2
  lcd.print ("Vocht: "); lcd.print(vochtigheid,1); lcd.print (" % ");
  delay(500);
  lcd.setCursor(0, 2); // zet de cursor op positie 1, regel 3
  lcd.print("Luchtdruk "); lcd.print (luchtdruk,1); lcd.print (" hPa");
  delay(500);
  lcd.setCursor(0, 3); // zet de cursor op positie 1, regel 4
  lcd.print("Regel 4"); // schrijf op scherm
  delay(1000);
}

```

Als alles goed gegaan is zie je nu op het scherm het graden teken:



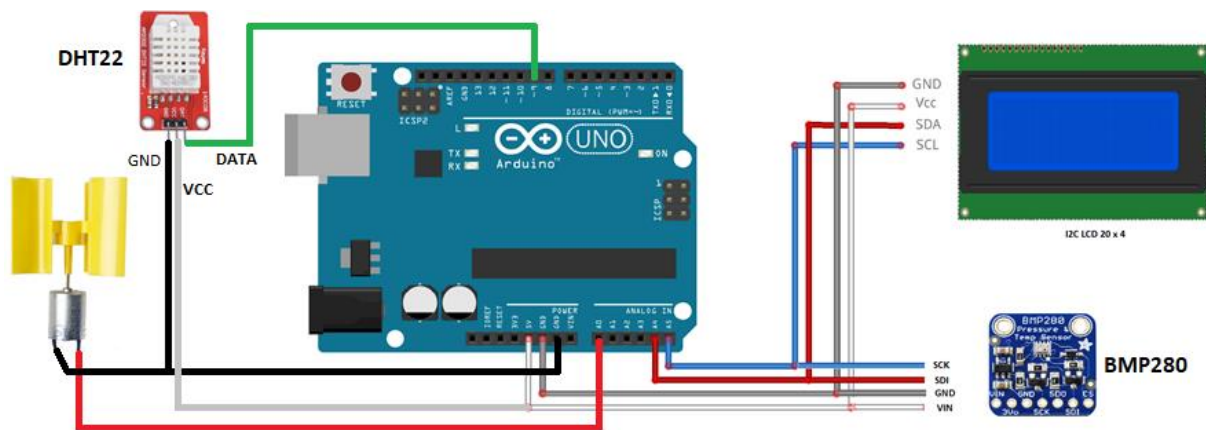
### **Windsnelheid meten**

De wind wordt gemeten op een mast met een anemometer. Een anemometer is een ronddraaiend molentje met halve bollen (cups) die met stangetjes aan een draaibare as zijn bevestigd. Het maken van een goede anemometer is niet zo gemakkelijk; voor onze toepassing nemen een goedkope oplossing die wel niet helemaal perfect is: een dynamo. Hoe harder het asje van de dynamo ronddraait hoe hoger de spanning is die uit de dynamo komt. De dynamo kun is eigenlijk een

gelijkstroom (DC) motortje, maar die kun je dus ook als dynamo gebruiken. De spanning die uit de dynamo komt meten we op een analoge ingang van de Arduino.



### Het aansluiten van de anemometer aan de Arduino



### De code voor weerstation 3b

In de code voor het weerstation lezen we voor de windsnelheid de analoge waarde van pen A0 van de Arduino. In de code is ook nog de functie **round()** gebruikt voor de waarde van de drukmeting. De drukmeting wordt door de **round()** functie afgerond naar de dichtbij gelegen gehele waarde.

```

/*weerstation_3b_V10
* temperatuur, vochtigheid, luchtdruk en windsnelheid meten.
* I2C voor LCD en BMP 280 sensor
*
* LCD GND en BMP GND --> GND Arduino
* LCD VCC en BMP Vin --> 5V Arduino
* LCD SDA en BMP SDI --> A4 Arduino
* LCD SCL en BMP SCK --> A5 Arduino
*
* I2C aansluitingen van Arduino Uno:A4 (SDA), A5 (SCL)
*
* DHT22 voor temperatuur en vochtigheid
* ARDUINO pen 9 voor data VAN DHT22

```

```

*
* windmeting op analog A0
*
*/

#include <Wire.h>           // Bibliotheek voor communicatie met I2C
#include <LiquidCrystal_I2C.h> // Bibliotheek voor het LCD scherm
#include <Adafruit_BMP280.h> // Bibliotheek voor de BMP 280 sensor
#include <DHT.h>           // DHT sensor bibliotheek toevoegen

LiquidCrystal_I2C lcd(0x27,20,4); // start LCD scherm en lcd is gebruiksnaam
Adafruit_BMP280 bmp;             // bmp is nu de gebruiksnaam voor de druk sensor
DHT sensor_dht(9, DHT22);       // onze sensor heet nu: sensor_dht

float luchtdruk, temperatuur, vochtigheid; // variabelen
int windmeting;

byte graden[8] = {           // graden teken
  B00010,
  B00101,
  B00101,
  B00010,
  B00000,
  B00000,
  B00000,
  B00000
};

void setup() {
  lcd.init();           // initialiseer het LCD scherm
  lcd.backlight();     // zet de backlight aan
  lcd.createChar(0, graden); // het symbool graden staat nu op positie 0
  bmp.begin();         // start de bmp sensor
  sensor_dht.begin(); // start de dht sensor
}

void loop() {
  delay(2000);
  lcd.clear();           // wis het scherm
  delay(500);
  // lees de temperatuur, vochtigheid, luchtdruk en windsnelheid uit en bewaar die in de variabelen
  temperatuur = sensor_dht.readTemperature();
  vochtigheid = sensor_dht.readHumidity();
  luchtdruk = bmp.readPressure()/100.0;
  windmeting = analogRead(A0);

  lcd.setCursor(0, 0); // zet de cursor op positie 1, regel 1
  lcd.print ("Temp : "); lcd.print(temperatuur,1); lcd.write(0); lcd.print ("C ");
  delay(500);
  lcd.setCursor(0, 1); // zet de cursor op positie 1, regel 2
  lcd.print ("Vocht: "); lcd.print(vochtigheid,1); lcd.print (" % ");
  delay(500);
}

```

```

lcd.setCursor(0, 2);          // zet de cursor op positie 1, regel 3
lcd.print("Luchtdruk "); lcd.print (round(luchtdruk)); lcd.print (" hPa");
delay(500);
lcd.setCursor(0, 3);          // zet de cursor op positie 1, regel 4
lcd.print("Wind: "); lcd.print (windmeting);
delay(1000);
}

```

Als alles goed gegaan is zie je op het scherm:



Als nu de rotor van de dynamo beweegt zal de 0 op de onderste regel van het scherm veranderen.

### Het programma wijzigen voor een rustigere uitlezing

Het voorgaande programma laat regel voor regel de meetwaarden zien; daarna wist het gehele scherm en begint het weer van voren af aan. Dat willen we eigenlijk niet, maar het was een eenvoudige manier om te beginnen. We hebben al eens een oefening gedaan om sterk variërende getallen op een LCD te schrijven; het bleek toen nodig om eerst het getal te wissen voordat we een nieuw getal er naar toe schrijven. Doen we dat niet dan blijft er soms een cijfer staan van de oude waarde. De oplossing is/likt dus 'eenvoudig'; voordat we een meetwaarde naar het LCD scherm schrijven wissen we eerst die regel. Dat werkt wel, maar het maakt de uitlezing erg onrustig (probeer het maar eens!). Een betere en mooiere oplossing is om wel de regel te wissen en opnieuw te schrijven maar alleen als de meetwaarde is veranderd.

Er bestaat geen commando om een regel op een LCD te wissen dus moeten we die zelf maken. Dat doen we door een subroutine of functie te maken die allemaal spaties op een lijn van het LCD plaatst.

Hoe weet ons programma of de meetwaarde is veranderd? Dat doen we met het **if** statement. Het **if** statement zijn we al tegengekomen in een van de eerste basis oefeningen. Maar het **if** statement (samen met **else** en ook met **else if**) geeft nog heel veel meer mogelijkheden. Op het internet vind je best wat informatie daarover bijvoorbeeld op:

<https://www.tweaking4all.nl/hardware/arduino/arduino-programmeer-cursus/arduino-programmeren-deel-4/#BeslissingenmakeninonsArduinoProgramma>

Tijdens de Coderdojo bijeenkomsten zal dit onderwerp nog wat uitgebreider worden toegelicht.

### De code voor weerstation 3c

```

/*weerstation_3c_V10
 * temperatuur, vochtigheid, luchtdruk en windsnelheid meten.
 * Om het display een wat rustiger aanzien te geven wordt alleen naar het display geschreven
 * indien er veranderingen zijn in de meetwaarden

```

```

*
* I2C voor LCD en BMP 280 sensor en LCD
*
* LCD GND en BMP GND --> GND Arduino
* LCD VCC en BMP Vin --> 5V Arduino
* LCD SDA en BMP SDI --> A4 Arduino
* LCD SCL en BMP SCK --> A5 Arduino
*
* I2C aansluitingen van Arduino Uno:A4 (SDA), A5 (SCL)
*
* DHT22 voor temperatuur en vochtigheid
* ARDUINO pin 9 voor data VAN DHT22
*
* windmeting op analog A0
*
*/

#include <Wire.h> // Bibliotheek voor communicatie met I2C
#include <LiquidCrystal_I2C.h> // Bibliotheek voor het LCD scherm
#include <Adafruit_BMP280.h> // Bibliotheek voor de BMP 280 sensor
#include <DHT.h> // DHT sensor bibliotheek toevoegen

LiquidCrystal_I2C lcd(0x27,20,4); // start LCD scherm en lcd is gebruiksnaam
Adafruit_BMP280 bmp; // bmp is nu de gebruiksnaam voor de druk sensor
DHT sensor_dht(9,DHT22); // onze sensor heet nu: sensor_dht

float luchtdruk, temperatuur, vochtigheid; // variabelen
int windmeting =1;

byte graden[8] = { // graden teken
  B00010,
  B00101,
  B00101,
  B00010,
  B00000,
  B00000,
  B00000,
  B00000
};

void emptyline(){ // lege lijn routine
  lcd.print (" ");
}

void setup() {
  lcd.init(); // initialiseer het LCD scherm
  lcd.backlight(); // zet de backlight aan
  lcd.createChar(0,graden); // het symbool graden staat nu op positie 0
  bmp.begin(); // start de bmp sensor
  sensor_dht.begin(); // start de dht sensor
  delay(1000);
  lcd.clear(); // wis het scherm

```

```

}

void loop() {
  //lees de temp, vochtigheid, luchtdruk en windsnelheid uit en vergelijk die met de vorige waarde

  if (temperatuur != sensor_dht.readTemperature())
  {
    temperatuur = sensor_dht.readTemperature();
    lcd.setCursor(0, 0);          // zet de cursor op positie 1, regel 1
    emptyline();
    lcd.setCursor(0, 0);          // zet de cursor op positie 1, regel 1
    lcd.print ("Temp : "); lcd.print(temperatuur,1); lcd.write(0); lcd.print ("C");
  }

  if (vochtigheid != sensor_dht.readHumidity())
  {
    vochtigheid = sensor_dht.readHumidity();
    lcd.setCursor(0, 1);          // zet de cursor op positie 1, regel 2
    emptyline();
    lcd.setCursor(0, 1);          // zet de cursor op positie 1, regel 2
    lcd.print ("Vocht: "); lcd.print(vochtigheid,1); lcd.print (" % ");
  }

  if (round(luchtdruk) != round(bmp.readPressure()/100))
  {
    luchtdruk = bmp.readPressure()/100;
    lcd.setCursor(0, 2);          // zet de cursor op positie 1, regel 3
    emptyline();
    lcd.setCursor(0, 2);          // zet de cursor op positie 1, regel 3
    lcd.print("Luchtdruk: "); lcd.print (round(luchtdruk)); lcd.print (" hPa");
  }

  if (windmeting != analogRead (A0))
  {
    windmeting = analogRead(A0);
    lcd.setCursor(0, 3);          // zet de cursor op positie 1, regel 4
    emptyline();
    lcd.setCursor(0, 3);          // zet de cursor op positie 1, regel 4
    lcd.print ("Wind: ");
    if (windmeting > 900){
      lcd.print ("storm !!");
    }
    else if (windmeting > 500){
      lcd.print ("hard");
    }
    else if (windmeting > 20){
      lcd.print ("matig");
    }
    else {
      lcd.print ("geen/weinig");
    }
  }
}

```



```
delay(1000);  
}
```

### Het eindresultaat

