

## Bouw je eigen weerstation: deel 4

### Inhoud deel 4

In deel 4 gaan we de weerstation gegevens die op een Arduino binnenkomen sturen naar een andere Arduino.

Het uitwisselen van digitale gegevens tussen computers heet data communicatie. Data communicatie is een heel belangrijk onderdeel van de informatica.

Enkele nieuwe begrippen zijn nodig om goed te begrijpen en daarna te programmeren hoe we data van het weerstation kunnen versturen en ontvangen.

### Bits, bytes en integers

De termen Megabyte (MB) en gigabyte (GB) worden bij een computer of smartphone gebruikt als je het over de grootte van het geheugen of de opslag capaciteit hebt. Wat betekenen ze?

Een **bit** is de kleinste eenheid van data/informatie in een computer. Een bit kan maar twee waardes hebben, namelijk 0 of 1. Deze waardes kunnen *aan/uit, ja/nee of waar/niet waar* betekenen.

Een **byte** bestaat uit 8 bits. Met één bit heb je maar twee mogelijke combinaties, namelijk 0 of 1. Met 2 bits zijn er al meer combinaties mogelijk, namelijk 4. Een byte heeft 256 mogelijke combinaties ( $2^8$ ). Je kan het aantal mogelijke combinaties van het aantal bits berekenen door 2 tot de macht het aantal bits te doen.

Een **integer** in de Arduino UNO bestaat uit 2 bytes hiermee kun je dus 2 tot de macht 16 verschillende combinaties maken en dus 65536 verschillende getallen weergeven. Omdat je ook negatieve getallen wilt kunnen gebruiken is in een Arduino UNO een integer een getal tussen -32.768 en +32.767.

### Seriële communicatie

Voor het uitwisselen van gegevens tussen computers gebruiken we vaak seriële communicatie. Bij seriële communicatie wordt telkens een bit verstuurd (of ontvangen). Bijna altijd doen we dat in pakketjes van 8 bits achter elkaar (dus 1 byte verzenden of ontvangen). Hiervoor zijn er in de computer speciale onderdelen die dit regelen namelijk de UART. De Arduino UNO heeft zo'n UART. De aansluitingen bij de UNO zijn digitaal pen 0 en pen 1. Pen 1 wordt gebruikt om data te verzenden en pen 0 wordt gebruikt om data te ontvangen. Het zenden en ontvangen gaat hier altijd met 8 bits achter elkaar dus telkens een byte. Als we twee Arduino's met elkaar willen laten communiceren moeten we dus de zend pen (ook TX genoemd) van de ene Arduino met de ontvang pen (ook RX genoemd) van de andere Arduino verbinden. En natuurlijk ook de ontvang pen van de eerste met de zend pen van de tweede verbinden. Zo kunnen deze Arduino's bytes met elkaar uitwisselen. Wel moeten we er voor zorgen dat de snelheid van de uitwisseling aan beide kanten gelijk is. De snelheid wordt uitgedrukt in bits per seconde (bit/s); een vaak gebruikte instelling is 9600 bit/s.

### Het versturen en ontvangen van data

Een integer bestaat uit 2 bytes. Om die via de seriële poort van de ene naar de andere Arduino een integer te sturen splitsen we de integer in 2 bytes; dat doen we met de commando's: `lowByte(i)` en `highByte(i)`. Hierin is `i` de integer. Aan de ontvangstkant combineren de ontvangen bytes weer tot de integer `i`. Dit doen we met de functie: `word(highByte, lowByte)`.

Om een float te versturen passen we een truckje toe. We maken van de float eerst een integer. Omdat we 1 cijfer achter de komma willen kunnen weergeven, vermenigvuldigen we de float met 10.

Daarna zetten we de float om in een integer. We versturen de integer en aan de ontvangkant delen we de integer weer door 10 en zetten we de integer weer om in een float.

De meteo data verzenden we als een pakket met alle gemeten waarden in dat pakket; dit noemen we een frame. Om het data transport op fouten te kunnen controleren zetten we voor en na het frame een kenmerk bij het zenden. In de ontvanger controleren we of de kenmerken juist zijn. Als die juist zijn dan weten we dat de ontvangen data goed is en kunnen we die in het programma gebruiken. Dat doen we hier door de meteo waarden op een display te zetten.

### Het programma voor het verzenden van de meteo data

```
/*weerstation_4a_Z_V10
* temperatuur, vochtigheid, luchtdruk en windsnelheid meten en de meetwaarden op het LCD
plaatsen
* Om het display een wat rustiger aanzien te geven wordt alleen naar het display geschreven indien
er veranderingen zijn in de          meetwaarden
* Alle meteo meetwaarden worden ook verzonden via de UART voor ontvangst op een ander
Arduino
* De meetwaarden worden in integer vorm (2 bytes per meetwaarde) omgezet indien dit nodig is.
* Het verzenden via de UART gebeurt per byte.
* Het verzenden gaat per frame met alle meetwaarden. Het frame wordt voorafgegaan met een
header (delimiter) en
* gesloten met een eindteken (framend)
*
* 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;
int tempsend, vochtsend;

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

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

void setup() {
  Serial.begin(9600);   // start de seriele monitor
  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 temperatuur, 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");
    }
}

tempsend = round (temperatuur *10);
vochtsend = round (vochtigheid *10);

Serial.write (84);
Serial.write (highByte(tempsend));
Serial.write (lowByte(tempsend));
Serial.write (highByte(vochtsend));
Serial.write (lowByte(vochtsend));
Serial.write (highByte(round(luchtdruk)));
Serial.write (lowByte(round(luchtdruk)));
Serial.write (highByte(windmeting));
Serial.write (lowByte(windmeting));
Serial.write (33);
delay(10000);

```

```
}
```

Een tweede Arduino gebruiken we nu om de ontvangen data op een display te zetten.

### Programma voor de ontvangende Arduino

```
/*weerstation_4a_O_V10
 *
 * meteodata wordt ontvangen in een frame.
 * Frame start en frame einde worden gecontroleerd op juiste code
 * De meteodata wordt als integer verstuurd en per byte ontvangen.
 * 2 ontvangen bytes worden dan gecombineerd om er weer een integer van te maken
 *
 * I2C aansluitingen van Arduino Uno: A4 (SDA), A5 (SCL)
 * GND --> GND Arduino
 * VCC --> 5V Arduino
 * SDA --> A4 Arduino
 * SCL --> A5 Arduino
 */

#include <Wire.h>          // Bibliotheek voor communicatie met I2C
#include <LiquidCrystal_I2C.h> // Bibliotheek voor het LCD scherm

byte temp1, temp2, v1, v2, l1, l2, w1, w2;
char del = 0, frameend = 0;
int luchtdruk, wind;
float temperatuur, vocht;

LiquidCrystal_I2C lcd(0x27, 20, 4); // start LCD scherm

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

void leesframe() // lees frame
{
  temp1 = Serial.read();
  temp2 = Serial.read();
  v1 = Serial.read();
  v2 = Serial.read();
}
```

```

luchthi = Serial.read();
luchtlo = Serial.read();
windhi = Serial.read();
windlo = Serial.read();
frameend = Serial.read();
}

void setup() {
  Serial.begin(9600);    // start de seriele monitor
  lcd.init();           // initialiseer het LCD scherm
  lcd.backlight();      // zet de backlight aan
  lcd.createChar(0, graden); // het symbool graden staat nu op positie 0
  pinMode (13, OUTPUT);
  digitalWrite(13, LOW);
}

void loop() {

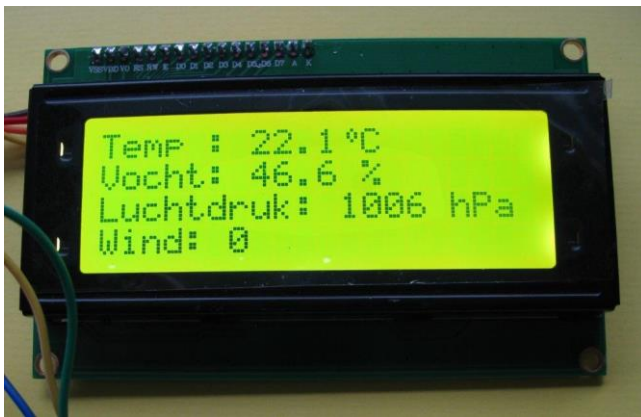
  if (Serial.available() > 12)
  {
    digitalWrite(13, HIGH);
    del = Serial.read();
    while (del != 84 ) {
      del = Serial.read();
    }; // search for 84
    del = 0;
    leesframe();
    if (frameend == 33) {
      frameend = 0;
      temperatuur = word(temphi, templo) / 10.0;
      vocht = word(vochthi, vochtlo) / 10.0;
      luchtdruk = word(luchthi, luchtlo);
      wind = word(windhi, windlo);
      lcd.clear();          // wis het scherm
      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");    // schrijf op scherm
      lcd.setCursor(0, 1);  // zet de cursor op positie 1, regel 2
      lcd.print("Vocht: "); lcd.print (vocht, 1); lcd.print (" % ");                    // schrijf op scherm
      lcd.setCursor(0, 2);  // zet de cursor op positie 1, regel 3
      lcd.print("Luchtdruk: "); lcd.print (luchtdruk); lcd.print (" hPa");            // schrijf op scherm
      lcd.setCursor(0, 3);  // zet de cursor op positie 1, regel 4
      lcd.print("Wind: "); lcd.print (wind);                                         // schrijf op scherm
      digitalWrite(13, LOW);
    }
  }
}

```

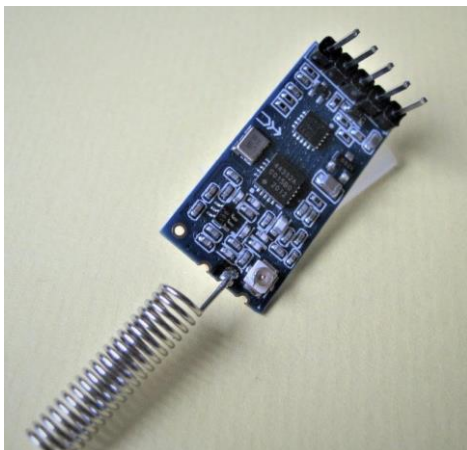
Laad het programma weerstation\_4a\_O\_V10 in een tweede Arduino. Koppel deze nu los van de computer door de USB stekker te verwijderen. Sluit het I2C LCD display aan op deze Arduino:  
LCD GND en BMP GND --> GND Arduino  
LCD VCC en BMP Vin --> 5V Arduino  
LCD SDA en BMP SDI --> A4 Arduino  
LCD SCL en BMP SCK --> A5 Arduino

Voed deze Arduino met een batterij.  
Koppel nu de zendende Arduino (Arduino\_Z) met de ontvangende Arduino (Arduino\_O) als volgt:  
GND (Arduino\_Z) --> GND (Arduino\_O)  
Digitale uitgang 1 – TX (Arduino\_Z) --> Digitale ingang 0 – RX (Arduino\_O)

Na enkele seconden zal op het display de ontvangen data te zien zijn.

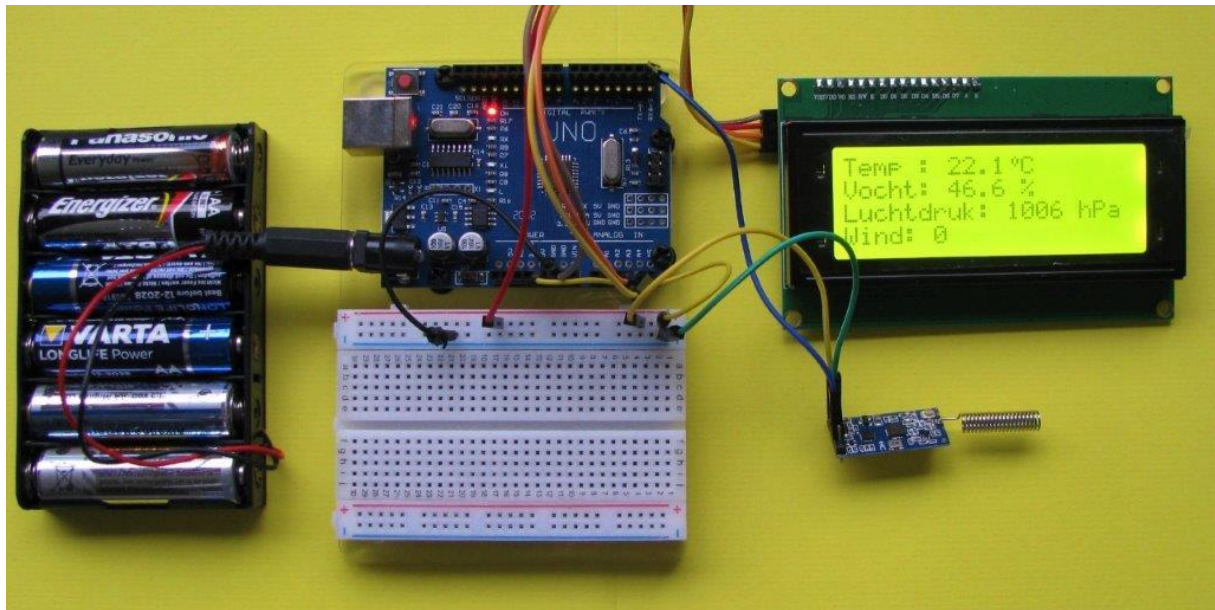


Als dat werkt gaan we nu de data via radiogolven verzenden. Verwijder de koppeling (twee draden) tussen de twee Arduino's



We nemen twee zend/ontvang modules HC-12 (zie foto). We gebruiken er één als zender door de RX pen van de HC-12 module met de TX (digitaal pen 1) van de zendende Arduino te koppelen. Uiteraard ook de voedingsspanning (5 volt) op de module aansluiten.  
Aan de ontvangst Arduino sluiten we ook een zend/ontvang module HC-12 aan maar nu de TX poort van de module naar de RX pen (digitaal pen 0) van de ontvang Arduino. Uiteraard ook de voeding van de module aansluiten GND naar GND en VCC naar de 5 volt.

Het resultaat is te zien op de volgende foto.



Tot afstanden van meer dan 10 meter kunnen we nu overbruggen.