

## 1 Weerstation

**Inleiding** Naast het meten aan kosmische straling met het HiSPARC meetstation kunnen leerlingen het HiSPARC station uitbreiden met een weerstation gemaakt met Arduino. Dit weerstation heeft als voordeel dat leerlingen dit zelf kunnen bouwen, aanpassen, programmeren, wind en water dichtmaken in een box en plaatsen bij het station. Zo kunnen zij metingen van luchtdruk, buitentemperatuur en temperatuur van de detectorplaten meten en correleren met gemeten kosmische straling. Daarnaast kan het station uitgebreid worden met andere sensoren zoals UV-index, bliksem, regen en windsensoren.

**Benodigdheden** Het weerstation kan met meer sensoren uitgerust worden dan hier genoemd. Het basis weerstation (luchtdruk, temperatuur en luchtvochtigheid) waar wij mee getest hebben heeft de volgende onderdelen nodig:

- Arduino Uno R3 (of elke andere Arduino)
- USB kabel
- DTH22 (of DTH11) luchtvochtigheid
- BMP085 luchtdruksensor
- DS18B20 digitale temperatuursensor (2 of 4x)
- arduino software

Andere sensoren zoals een bliksemdetector (AS3935) zijn tot op heden niet getest, maar geven leerlingen een extra onderzoek mogelijkheid(namelijk de correlatie tussen bliksem en kosmische straling.)

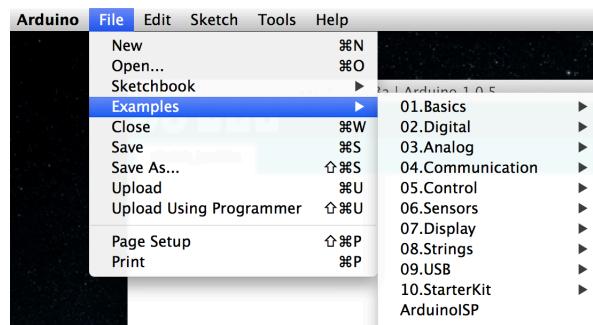
## 2 Arduino

### 2.1 Werking van Arduino

Een Arduino is een microcontroller met een ATMEGA-328 chip die bestaat uit een printplaatje met een aantal digitale en analoge ingangen, waar bijvoorbeeld sensoren op aangesloten kunnen worden, zie Figuur 1.1. Data van de sensoren kan via usb of draadloos (wat wij gaan doen) ingelezen worden in de Arduino software of met een ander programma. Deze software is vrij te downloaden van <https://www.arduino.cc>. Als u al ervaring heeft met Arduino zal het aansluiten van sensoren redelijk eenvoudig zijn. Als er nog geen ervaring is met Arduino zijn er op het internet een aantal handleidingen en tutorials beschikbaar evenals Arduino starterkits. Het



**Figuur 1.1 – Arduino Uno R3, basis voor het weerstation**



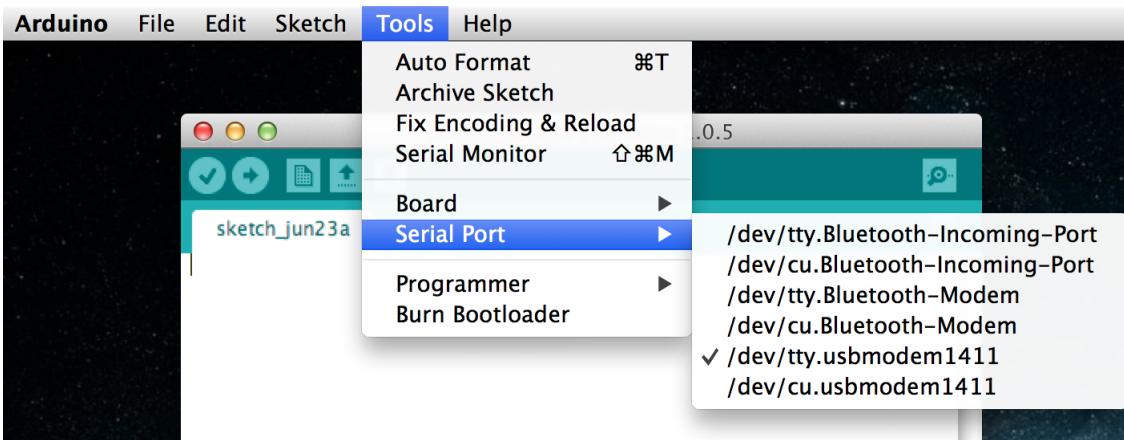
**Figuur 2.1 – Arduino Software, voorbeeld programma's**

wordt aanbevolen om eerst een aantal basisschakelingen te maken met Arduino. Voorbeelden van handleidingen zijn te vinden op <https://arduino.cc>.

Zoals in Figuur 1.1 te zien is zijn er zes analoge ingangen A0-A5 en zes digitale ingangen (pin 2 t/m 7). Voor het basis weerstation zullen we van beide twee ingangen gebruiken. Het voordeel van digitale ingangen is dat zij meerdere waarden kunnen verwerken. Er kunnen dus meerdere digitale temperatuursensoren (DS18B20) op één digitale ingang aangesloten worden.

De Arduino software heeft een aantal voorbeelden, die de gebruiker snel een aantal zaken leert. Door de voorbeelden te bestuderen kun je redelijk snel programmatuur onder de knie krijgen. Voor ons is het belangrijk dat we Arduino gebruiken om sensorwaarden uit te lezen. In de Arduino software is dan het voorbeeld: /Tutorial/AnalogReadSerial erg belangrijk. Het voorbeeldprogramma "AnalogReadSerial" bijvoorbeeld staat onder *file / tools / examples / Basics*. Zie Figuur 2.1.

Sluit de Arduino aan met de USB kabel op de pc. Controleer dat de Arduino software de Arduino vindt. Kijk hiertoe in het menu: *Tools / Serial Port*. Meestal vindt de software de usb poort vanzelf. Zie Figuur 2.2.



**Figuur 2.2 – Arduino Software, USB poort is gevonden door de software**

## 2.2 Programmeren in Arduino

Het programmeren in Arduino is relatief eenvoudig. Een programma of ‘sketch’ bevat een header en twee *void* onderdelen. In de header worden libraries aangeroepen, elke library een bepaalde functionaliteit, bijv. een library die de ijking van een sensor bevat. Daarnaast kunnen in de header datapins en variabelen gedefinieerd worden en onderdelen van libraries aangeroepen worden. In *void setup* kunnen ook variabelen gedefinieerd en kunnen libraries aangeroepen, ook wordt hier de *bitrate* communicatie met de Arduino vastgezet met het commando:

```
Serial.begin(9600);
```

Deze *bitrate* is vaak 9600. *Void loop* is het eigenlijke programma wat keer op keer herhaald wordt. Vaak wordt een delay toegevoegd om de loop enigzins te vertragen, omdat het uitlezen van de sensoren nu eenmaal wat tijd nodig heeft. Na de *void loop* kunnen er nog functies gemaakt worden, die aangeroepen worden in de *void loop*. Als een programma af is kan die met de pijl in het bovenscherf geüpload worden naar de Arduino. Om de output van de Arduino te bekijken, moet op het vergrootglas rechts bovenin geklikt worden. Hiermee komt u in het scherm van de *Serial Monitor*. In *Serial Monitor* kun je zien wat het programma doet.

## 2.3 Gebruiken van een library

Libraries zijn stukjes code die bepaalde functies bevatten die het programma in Arduino kan inladen met het *include* statement. Bij de sensoren die wij voor dit project gebruiken, gebruiken we standaard libraries van Arduino, maar downloaden we ook libraries behorend bij de sensoren die we nodig hebben voor ons weerstation. Bij elke sensor wordt aangegeven waar de betreffende library te downloaden is. Als een library gedownload is kunt U op twee manieren deze library toevoegen aan de Arduino software op uw pc. De eerste methode is om op : *Sketch / Import Library... / Add library* te klikken en dan de map aan te wijzen waar de library instaat en dan *choose* te kiezen. Dan wordt deze library toegevoegd aan de lijst met libraries en kan deze aangeroepen worden in een Arduino programma of ‘sketch’. De tweede methode is om de map

met de library handmatig te kopiëren in de map *Libraries* naar de map waar de programma's (sketches) van Arduino standaard staan. Dan zal Arduino software (na opnieuw opstarten van het programma), deze libraries herkennen.

## 3 Sensoren

### 3.1 Inleiding

Bij dit weerstation gaan we een aantal sensoren gebruiken om naast de al beschikbare data van de HiSPARC stations ook weerdata mee te nemen in onderzoek naar correlaties van kosmische straling met weergrootheden. We zijn ook geïnteresseerd in de temperatuur van de detectors. Buisman [1] heeft aangetoond dat de PMT van de detector gevoelig is voor temperatuur. Om dit verschijnsel te meten plaatsen we in elke detector ter hoogte van de PMT een digitale temperatuursensor (DS18B20). De luchtdruk- en luchtvochtigheidssensor worden bij de Arduino geplaatst in een (zelf te ontwikkelen) weerhuisje. Leerlingen kunnen dan zelf correlaties tussen luchtdruk en aantal events onderzoeken. De luchtvochtigheid biedt weer opties om bijvoorbeeld het dauwpunt uit te rekenen. Hieronder worden de verschillende sensoren kort toegelicht en uitgelegd hoe ze uit te lezen zijn met de Arduino software.

### 3.2 Temperatuursensor

Om de temperatuur van de detectoren te meten gebruiken we 2 of 4 dezelfde temperatuursensoren (DS18B20). Om de sensor te laten werken is het belangrijk om de ‘pootjes’ van de sensor op de juiste manier aan te sluiten. Deze sensor kan geademd worden op twee van de drie pootjes en wordt gevoed en uitgelezen op het pootje S (signal). Het pootje S wisselt per uitvoering. Onze versie (‘Keyes’) heeft het ‘Signal’ pootje rechts, zie Figuur 3.1a. Bij uw sensor kan dit anders zijn. Raadpleeg in dat geval de datasheet van de sensor of inspecteer de inscripties bij de pootjes van de sensor.

**Aansluiten en meten met de temperatuursensor** Om de digitale temperatuursensor te gebruiken is een schakeling nodig zoals in Figuur 3.1b is getekend. De ground en de voedingspin zijn beide verbonden met de ground van de Arduino. Het Signal of data pootje is via een  $4,7\text{ k}\Omega$  weerstand verbonden met de 5 V. Over de dezelfde draad gaat én de voeding én het signaal. Om het geheel te laten werken hebben we echter wel twee libraries nodig, om de temperatuursensor aan te sturen. De eerste library *OneWire* moet gedownload worden en zorgt ervoor dat meerdere signalen over één digitale ingang uitgelezen kunnen worden. U vindt *OneWire* hier: <https://playground.arduino.cc/Learning/OneWire>. Download ook de library: *DallasTemperature*. Die vindt U hier: [https://www.milesburton.com/Dallas\\_Temperature\\_Control\\_Library](https://www.milesburton.com/Dallas_Temperature_Control_Library)

Het verdient aanbeveling om na het aansluiten van de temperatuursensoren zoals in Figuur 3.1b, uit de map *OneWire / example* het voorbeeld programma: *DS18x20\_Temperature* uit te voeren. Het programma zoekt dan naar de adressen van de aangesloten temperatuursensoren. Als er

```

read_temp_ds18b20.ino
Run programma
Serial Monitor

#include <OneWire.h>
#include <DallasTemperature.h>
// Data wire is plugged into pin 3 on the Arduino
#define ONE_WIRE_BUS 3
// Setup a OneWire instance to communicate with any OneWire devices
// (not just Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);
// Pass our OneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire); // gebruik libraries
void setup(void) // Setup routine
{
  // start serial port
  Serial.begin(9600); // Bitrate communicatie met Arduino
  Serial.println("Dallas Temperature IC Control Library Demo");
  // Start up the library
  sensors.begin();
}
void loop() // Herhaal loop; eigenlijke programma
{
  // request to all devices on the bus
  Serial.print(" Requesting temperatures... ");
  sensors.requestTemperatures(); // Vertraging in de loop in ms; delay hier 0,5 s
  for (int deviceA = 0; deviceA < 4; deviceA++) {
    printTemp(deviceA); // For loop over 4 Temperatuur sensoren
  }
}
void printTemp(int address) {
  float TempC = sensors.getTempCByIndex(address);
  String stringone = "TempDevice ";
  stringone += address;
  Serial.print(stringone);
  Serial.print(" ");
  //Serial.print(address);
  Serial.println(TempC);
}

Done Saving.          foutmeldingen / status upload.

```

Arduino Uno on /dev/tty.usbmodem1411

**Figuur 2.3 – Arduino Software, de belangrijkste onderdelen van een programma uitgelegd.** Eerst worden libraries ingeladen met het `include` statement. Dan wordt de datapin gedefinieerd. Libraries worden aangeroepen. In `void setup` definieert u variabelen, kunt u libraries starten. De `void setup` wordt één keer uitgevoerd. `Void loop` is het eigenlijke programma wat continue herhaalt wordt. Als laatste ziet u een zelf gedefinieerde functie (`printTemp`), die binnen de `void loop` aangeroepen wordt.

geen adressen gevonden worden dan zijn de temperatuursensoren foutief aangesloten. Verdere uitleg over het voorbeeldprogramma staat op de volgende site: [https://www.pjrc.com/teensy/td\\_libs\\_OneWire.html](https://www.pjrc.com/teensy/td_libs_OneWire.html).

Als alle temperatuursensoren aangesloten zijn, de libraries geïnstalleerd en getest, dan kunnen de temperatuursensoren uitgelezen worden. Met de volgende code kunnen de sensoren uitgelezen worden:

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged into Digital pin 3 on the Arduino
#define ONE_WIRE_BUS 3

// Lib: OneWire is nodig om contact maken met ingang D3.
OneWire oneWire(ONE_WIRE_BUS);

// oneWire reference naar Lib: Dallas Temperature.
DallasTemperature sensors(&oneWire);

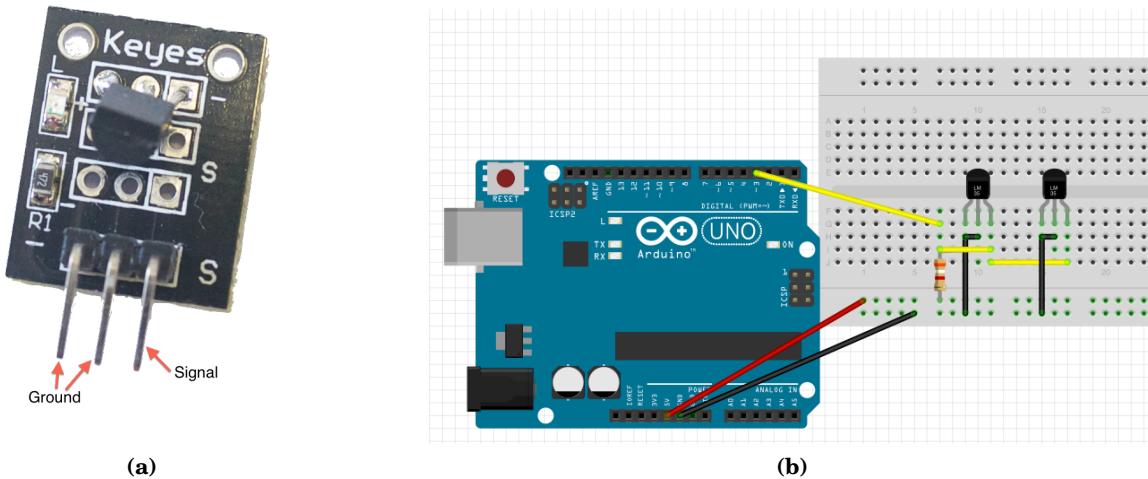
void setup(void) {
    // start serial port
    Serial.begin(9600);

    // Start up the library
    sensors.begin();
}

void loop() {
    // request to all devices on the bus
    Serial.print(" Requesting temperatures... ");
    sensors.requestTemperatures();
    Serial.println("DONE");

    // Het uitlezen van de sensoren kan in een for-loop om code te sparen.
    // Hier < 4 omdat er 4 temperatuursensoren aangesloten zijn.
    for (int deviceA = 0; deviceA < 4; deviceA++) {
        printTemp(deviceA);
    }
    delay(500);
}

void printTemp(int adress) {
```



**Figuur 3.1** – De DS18B20 temperatuursensor los en in een schakeling. De gele draad is zowel voeding voor de sensor als transmissie van het signaal. De zwarte draad zorgt voor aarding van zowel het linker als midden pootje, dus deze kan doorverbonden worden. Bij het testen van deze schakeling blijkt dat grotere lengte van de gele draad geen signaalverlies tot gevolg heeft.

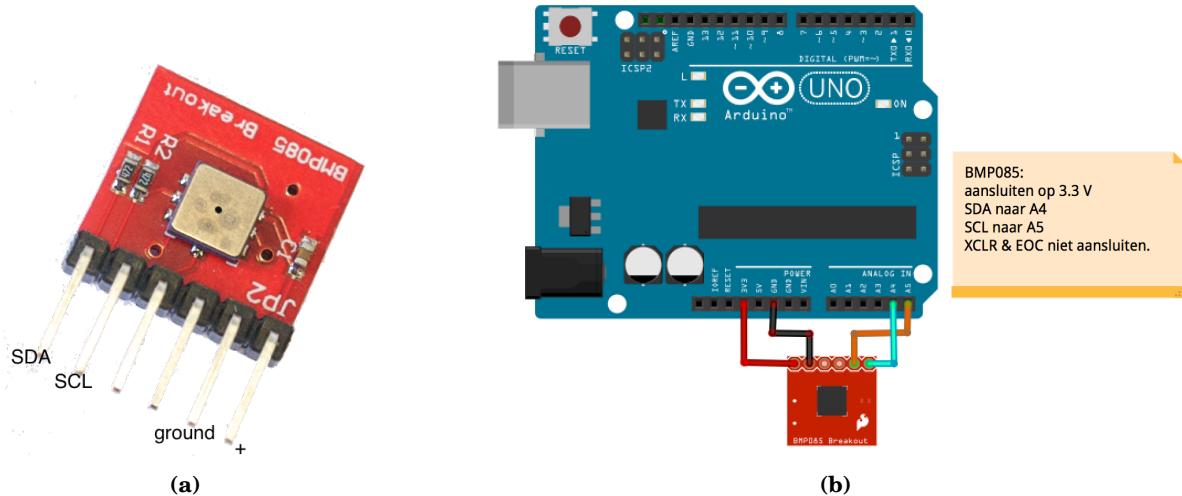
```
// Deze code kun je gebruiken om iedere temperatuursensor uit te lezen.
// Met de functie getTempCByIndex kun je de gewenste sensor uit lezen.
// Temperatuursensor 1 krijgt index waarde 0.
float TempC = sensors.getTempCByIndex(adress);
Serial.println("TempDevice", adress, ":" , TempC);
}
```

### 3.3 Luchtdruksensor

De luchtdruk gaan we met de luchtdruksensor (BMP085), die zowel luchtdruk als temperatuur kan meten. De data wordt verwerkt door de ingangen A4 en A5 van de Arduino.

**Aansluiten en meten met de luchtdruksensor** Zoals in Figuur 3.2a te zien is heeft de luchtdruksensor 6 aansluitpootjes, waarvan wij er 4 nodig hebben. De voeding is 3,3 V, de ground van de luchtdruksensor sluiten we weer aan op de ground van de Arduino. De luchtdruksensor is een zogenaamd  $I^2C$  device, dat betekent dat de twee overige aansluitingen een datalijn (SDA) en een seriële clock (SCL) zijn. Om te communiceren met de Arduino zijn een library voor de sensor nodig en een standaard library van Arduino om  $I^2C$  data te verwerken, namelijk *Wire.h*. De BMP085 library is op verschillende plekken te downloaden, van verschillende fabrikanten. Wij hebben de volgende (open source) library gebruikt <https://code.google.com/p/bmp085driver/> Pak deze library uit of installeer het zip bestand van de library in een keer in Arduino via *Sketch / Import Library... / Add library* (Arduino herkent namelijk ook *zip* - bestanden).

Als de sensor is aangesloten volgens Figuur 3.2b kan met een example programma uit de library getest worden of de sensor werkt. In de library kan met de luchtdruk waarde ook de hoogte



**Figuur 3.2** – De BMP085 los en in een schakeling. De voeding is 3,3V en de data wordt via een seriële data en clock lijn uitgelezen op A4 en A5.

worden uitgerekend, maar dan moet echter de druk op zeeniveau ook bekend zijn en meegegeven worden. Omdat het weerstation waarschijnlijk een vaste plek krijgt is het uitrekenen van de hoogte niet nodig en deze programma regels kunnen verwijderd worden.

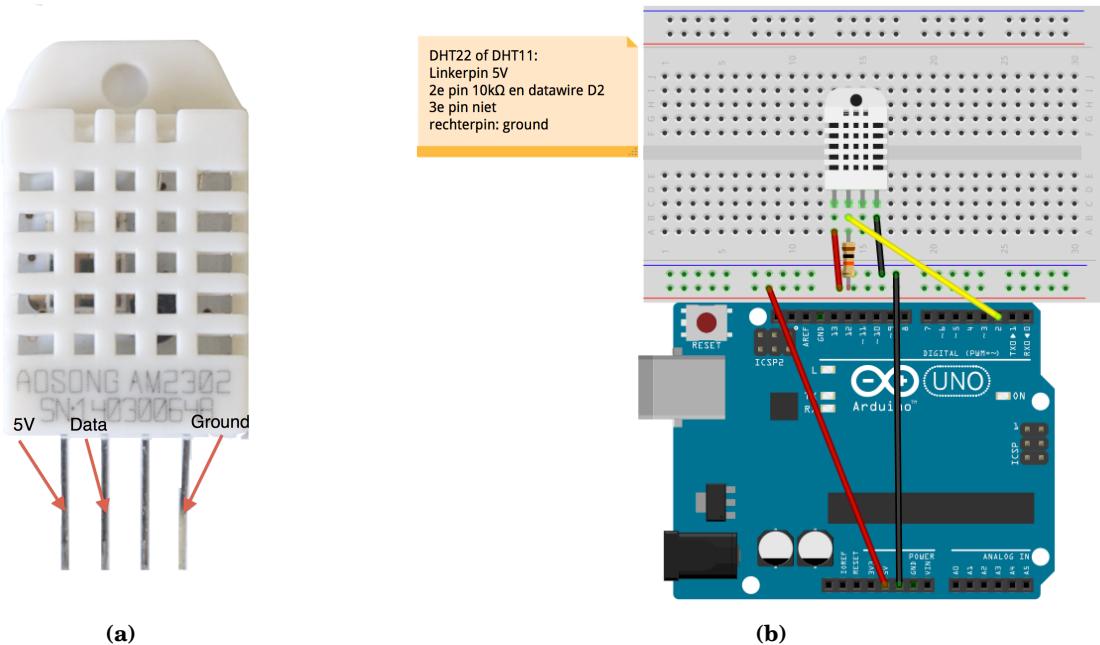
### 3.4 Luchtvochtigheidssensor

De (relatieve-) luchtvochtigheid gaan we meten met de luchtvochtigheidssensor (DHT22 of DHT11), deze sensoren kunnen ook temperatuur meten. Wij hebben gekozen voor de iets duurdere DHT22, omdat die nauwkeuriger is, een relatieve luchtvochtigheidsbereik van 0 % tot 100 % en een temperatuurbereik heeft van  $-40^{\circ}\text{C}$  tot  $125^{\circ}\text{C}$ . De DHT11 heeft een bereik van 0 % tot 80 %, temperatuurbereik van  $0^{\circ}\text{C}$  tot  $50^{\circ}\text{C}$  en een nauwkeurigheid van 5 %.

**Aansluiten en meten met de luchtvochtigheidssensor** De luchtvochtigheidssensor heeft 4 pinnen, waarvan we er 3 gebruiken. De linkerpin is de voeding 5,0V, de 2e pin van links is de datapin, die wordt via een weerstand van  $10\text{ k}\Omega$  verbonden aan digitale ingang 2 van de Arduino. De derde pin van links is niet in gebruik. De rechter pin is de ground. Zie Figuur 3.3a. We hebben wederom een library nodig om de DHT werkend te krijgen in de Arduino software. De library voor DHT22 of DHT11 is te downloaden van <https://playground.arduino.cc/Main/DHTLib>, klik daar op *latest version on github*. Als de sensor aangesloten is volgens Figuur 3.3b, kan weer een example geprobeerd worden die bij de library geleverd wordt.

### 3.5 Weerstation

Nu we alle sensoren afzonderlijk hebben getest,kunnen we de sensoren gaan combineren en ook het programma schrijven wat alle sensoren in een keer netjes uitleest en de waarden netjes print.



**Figuur 3.3** – De DHT los en in een schakeling. De voeding is 5,0 V. Naar de datapin zit een weerstand van 10 k $\Omega$ . Zowel de temperatuur als de relatieve luchtvochtigheid worden uitgelezen via een ingang.

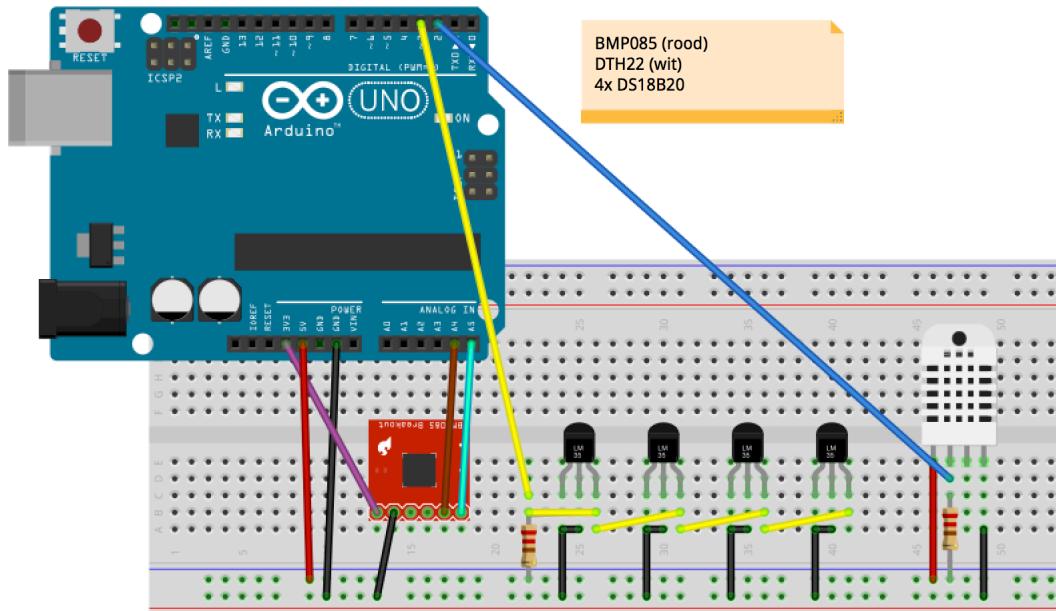
Deze waarden kunnen we dan samen met een datum en tijd functie printen of wegschrijven naar een textbox of spreadsheetprogramma.

In Figuur 3.4 zien we alle sensoren (op een breadboard) aangesloten op een Arduino. We kunnen nu meten met het weerstation, we hebben echter nog wel een vaste verbinding nodig. In de hieropvolgende handleiding gaan we het weerstation draadloos maken. In de volgende handleiding 'weerstation data' lezen we deze data uit van de Arduino en sturen we dat naar de HiSPARC database.

## 4 FAQ

Bij het werken met Arduino en de sensoren zijn er een aantal zaken die vaak mis gaan. De meest voorkomende worden hieronder opgesomd en beantwoord.

- *Arduino wil niet uploaden.* Zorg dat het juiste Arduino bordje is geselecteerd, dit stel je in, in de Arduino software bij *Tools/Board* vink daar het juiste Arduino bordje aan.
- *Foutmelding bij uploaden* De juiste usb poort moet altijd ingesteld zijn. Vink daarom de juiste usb poort aan. De Arduino software vindt deze niet altijd automatisch. Zoals te zien is in Figuur 2.2. Is de USB poort niet juist dan zie je de volgende foutmelding in het venster.  
avrdude: stk500\_recv(): programmer is not responding.
- *Onjuiste data.* Zorg dat je in je Arduino code op de juiste ingangen uitleest. Check of je de juiste pinnen van de Arduino hebt aangesloten de uitgangen van de sensoren en of deze matchen met de gebruikte waarden in de code.



**Figuur 3.4 – Alle sensoren aangesloten op de Arduino**

- *Rare meetwaarden in serial monitor* Als de meetwaarden van de sensoren niet veranderen of de hele tijd dezelfde waarden geven, dan moet de schakeling gecontroleerd worden. Het verdient aanbeveling om de schakeling helemaal af te breken en opnieuw te maken.
- *Andere foutmeldingen* Check the troubleshooter van Arduino: <https://support.arduino.cc/hc/en-us>

## Referenties

[1] H. Buisman, *HiSPARC Experiment: Lichtopbrengst en temperatuur*, LiO jaarverslag 2011