



Workshop Arduino voor beginners deel 2

Voor deze workshop is het van belang dat je deel 1 van de workshop hebt afgerond.

In deze workshop zitten meerdere opdrachten die elkaar niet allemaal op hoeven te volgen. Maak eerst opdracht 1, daarna kun je elke opdracht afzonderlijk in willekeurige volgorde maken.

Probeer de opdrachten te verdelen over de deelnemers en wissel van opdracht als je klaar bent. Zorg dat je alle onderdelen van 1 opdracht netjes bij elkaar houdt zodat je deze als 1 pakket door kunt geven aan de volgende deelnemer.

In deze workshops zul je merken dat je zelf vaker op zoek moet naar schema's en programma code om de betreffende onderdelen aan te sturen.

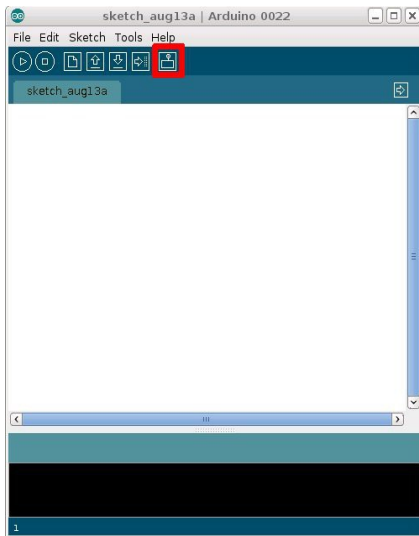
De programma code van de opdrachten zijn beschikbaar op een USB stick of op de desktop van de computer waar je op werkt, je hoeft dus niet zoals bij de eerste workshop alles over te typen. Lees de code wel goed door en probeer te begrijpen wat er gebeurt.

Opdracht 1:

In deze opdracht gaan we leren werken met de serial monitor:

In de Arduino IDE zit standaard een serial monitor verwerkt, deze monitor kun je gebruiken om input data van de arduino naar je scherm te schrijven.

De serial monitor wordt vaak gebruikt om snel te controleren of de data die je binnen krijgt van een drukknop, temperatuur sensor of ander apparaat inderdaad overeen komt met wat je verwacht had.



Sluit de arduino aan op de pc en draai deze code.

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("Hello World!");  
  delay (100);  
}
```

Klik vervolgens op de serial monitor en je zult de text Hello World! Continu in het scherm zien lopen.

Serial.begin(); zet de baudrate van de serial poort.

Serial.println(); print een text of een variabele en ga naar de volgende regel.

In het volgende voorbeeld gaan we data lezen van een analoge input poort:

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println(analogRead(0));  
  delay (100);  
}
```

Als we data gaan lezen van meer dan 1 input poort is het handig dat we weten welke poort we uitlezen.

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.print("analog port 0 ");  
  Serial.println(analogRead(0));  
  Serial.print("analog port 1 ");  
  Serial.println(analogRead(1));  
  delay(100);  
}
```

Hier zien we ook direct 2 nieuwe functies:

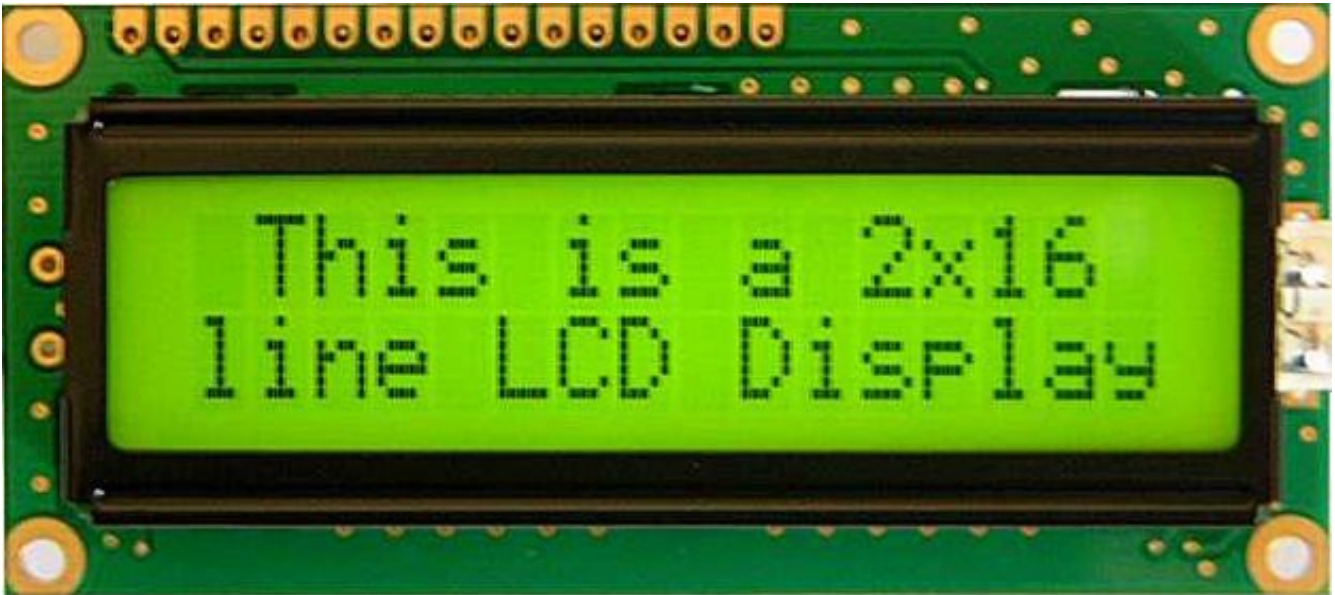
Serial.print(); print een text of een variabele en blijf op dezelfde regel.

Delay(); Een vertraging in milli seconden voor de processor verder gaat met zijn programma.

De serialprint functie zullen we in een de volgende opdrachten weer tegen komen.

Opdracht 2:

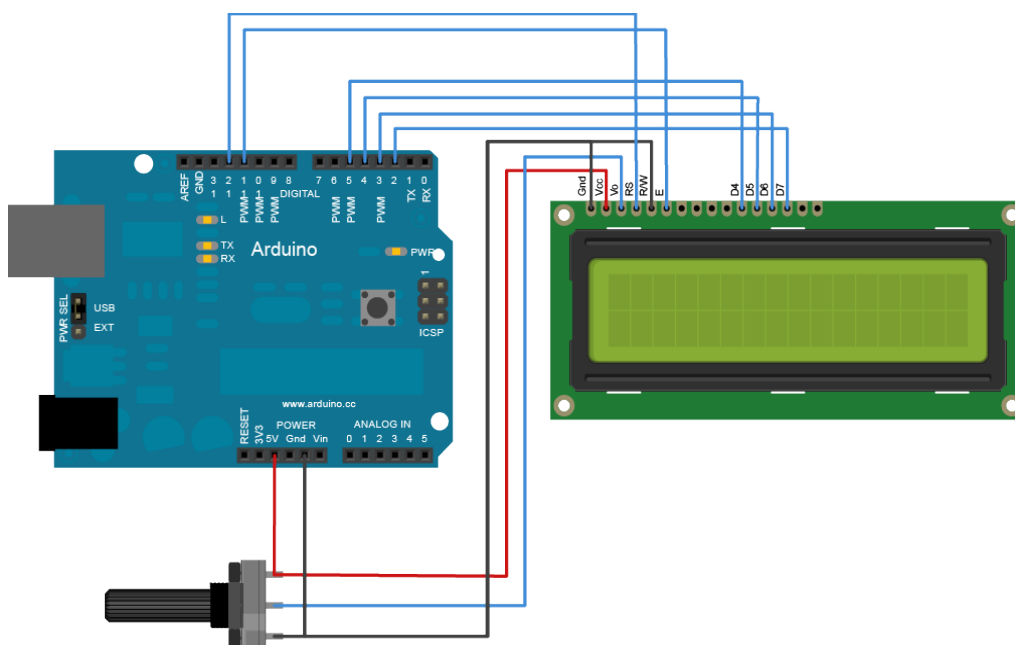
In deze opdracht gaan we leren werken met een 2x16 LCD scherm.



Voor dit type LCD scherm is er een learning pagina op arduino.cc:

<http://arduino.cc/en/Tutorial/LiquidCrystal>

Sluit het LCD scherm aan zoals op de site omschreven.



Gebruik de voorbeeld code van de arduino pagina:

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```

Als je deze code werkend hebt op je arduino draai je aan de potmeter tot dat je de letters op het scherm goed kunt lezen. Met de potmeter regel je de contrast van het display.

Hier zien we meteen een goed voorbeeld van statische text “hello, World!” en Variabele text “de tijd in seconden vanaf het opstarten van de arduino”.

Wat verder opvalt is dat de library voor de lcd module modulair geschreven is, we kunnen zelf bepalen welke pinnen van de arduino we gebruiken voor het lcd scherm: LiquidCrystal lcd(12, 11, 5, 4, 3, 2); De volgorde dient wel altijd hetzelfde te zijn: LiquidCrystal lcd(RS, E, D4, D5, D6, D7);

Sluit de display nu aan op de data pinnen (2, 3, 4, 5, 6, 7) in deze volgorde.

Pas je arduino code aan op deze setup en draai de voorbeeld code nog eens.

Het aanpassen van deze pinnen kan in de toekomst erg handig zijn als je bijvoorbeeld de SPI interface (pin 11, 12, 13) vrij wilt houden voor een ethernet module, SD kaart of als je zo veel mogelijk PWM pinnen over wilt houden.

lcd.begin(); bepaald welk type LCD scherm er gebruikt wordt.

lcd.print(); print een text of een variabele op het scherm.

lcd.setCursor(); set de cursor naar een positie op het scherm 5, 0 is het 6de teken op de eerste regel 10,1 is het 11de teken op de 2de regel.

Pas de code zo aan dat je voornaam in de rechter bovenhoek en je achternaam in de linker onderhoek van het LCD scherm te zien is.

Opdracht 3:

In deze opdracht gaan we werken met een temperatuur sensor.



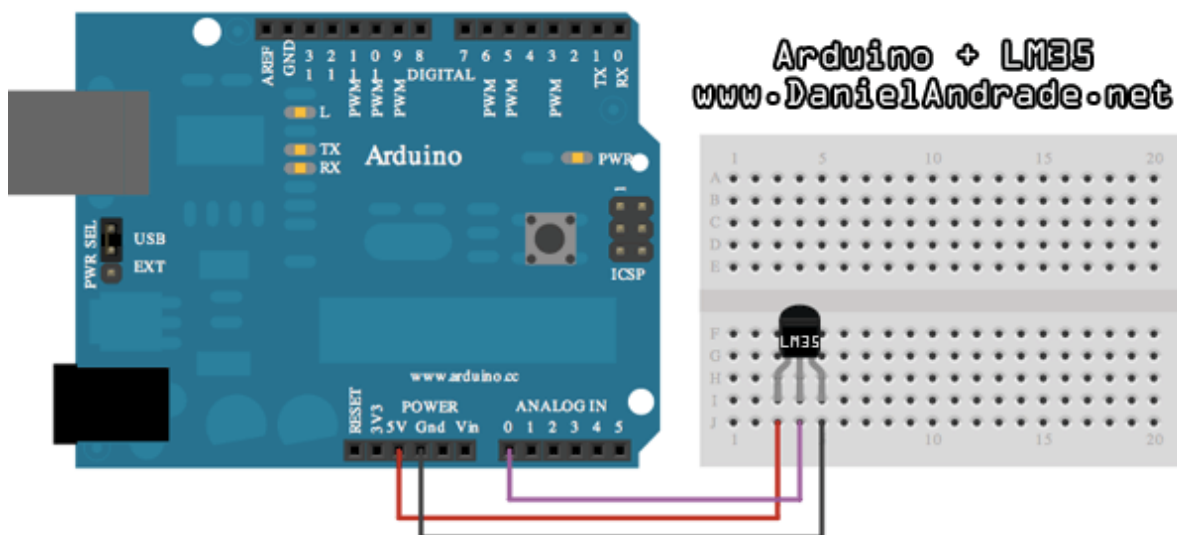
Dit is een vrij simpele variant van een temperatuur sensor die aan de hand van de voeding spanning een uitgang varieert tussen 0 en 5 volt.

Prima geschikt voor de analoge ingang op de arduino bordjes.

Ook dit project staat volledig beschreven op internet:

<http://www.danielandrade.net/2008/07/05/temperature-sensor-arduino/>

Sluit de temperatuur sensor aan zoals op de site omschreven:



De simpelste manier om de temperatuur uit te lezen en zichtbaar te maken is via de serial monitor:

Voer deze code uit en open de serial monitor.

```
void setup()
{
  Serial.begin(9600); // start serial communication
}

void loop()
{
  Serial.println (((analogRead(0)* 5.0 / 1024) - 0.5)* 100);
}
```

Je ziet nu de temperatuur van de sensor in je serial monitor.

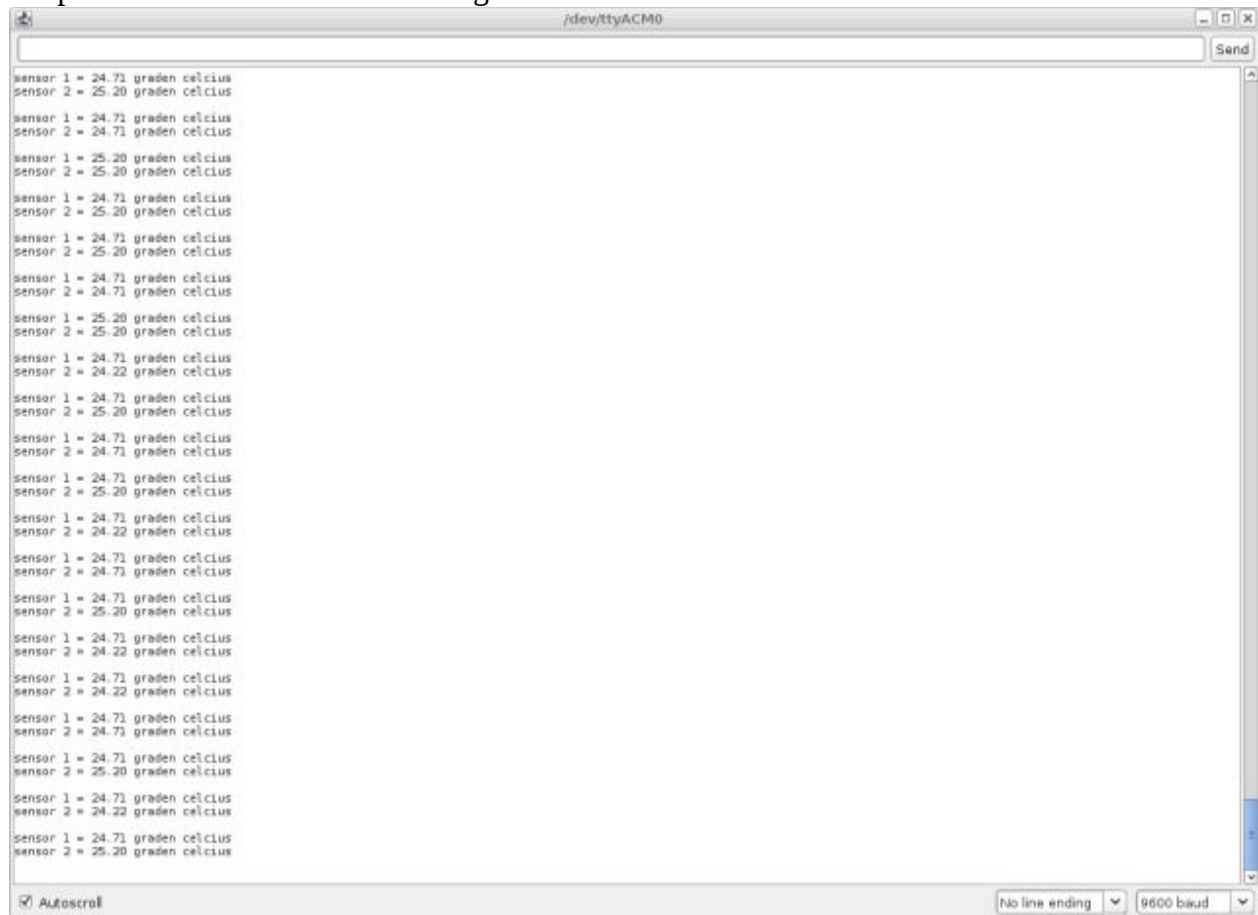
Sluit nu de 2de temperatuur sensor aan op analoge poort 1.

Pas de bovenstaande code aan zodat je 2 sensors uit kunt lezen.

Zorg er voor dat je:

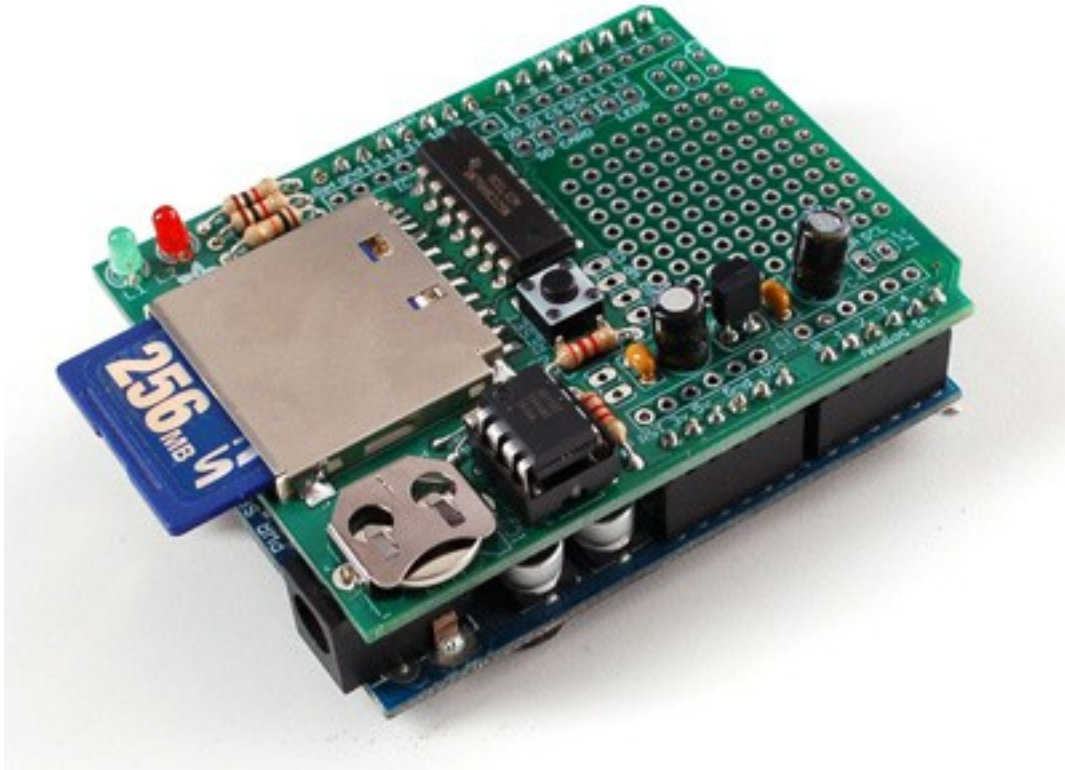
- van beide sensors kunt zien welke sensor het is.
- dat er voldoende tijd is om de data te lezen.
- Dat de eenheid in graden celsius word weergegeven achter de temperatuur.

De output in de serial monitor zal er ongeveer zo uit kunnen zien:



Opdracht 4:

In deze opdracht gaan we aan de slag met het data logger shield van Adafruit.



Deze shield voegt 2 extra functies toe aan je Arduino:

1. een real time clock
2. een SD kaart voor data opslag

de real time clock:

Een real time clock is erg handig als je gegevens op wilt slaan en later weer wilt lezen. Je kunt dan precies zien op welk tijdstip er iets gelogd is.

Als je de clock voor de eerste keer gebruikt of als je de batterij vervangen hebt moet je de clock eerst op tijd zetten.

In onze programma zetten we de tijd gelijk met de clock van de pc bij het uploaden van de code:

```
RTC.adjust(DateTime(__DATE__, __TIME__));
```

Op de website van Adafruit staat alle informatie die je voor deze shield nodig hebt:

<http://www.ladyada.net/make/logshield/lighttemp.html>

Nu gaan we de data die de clock uitstuurd zichtbaar maken in de serial monitor:


```
#include <Wire.h>
#include "RTCLib.h"

RTC_DS1307 RTC;

void setup () {
  Serial.begin(9600);
  Wire.begin();
  RTC.begin();
  if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
  }

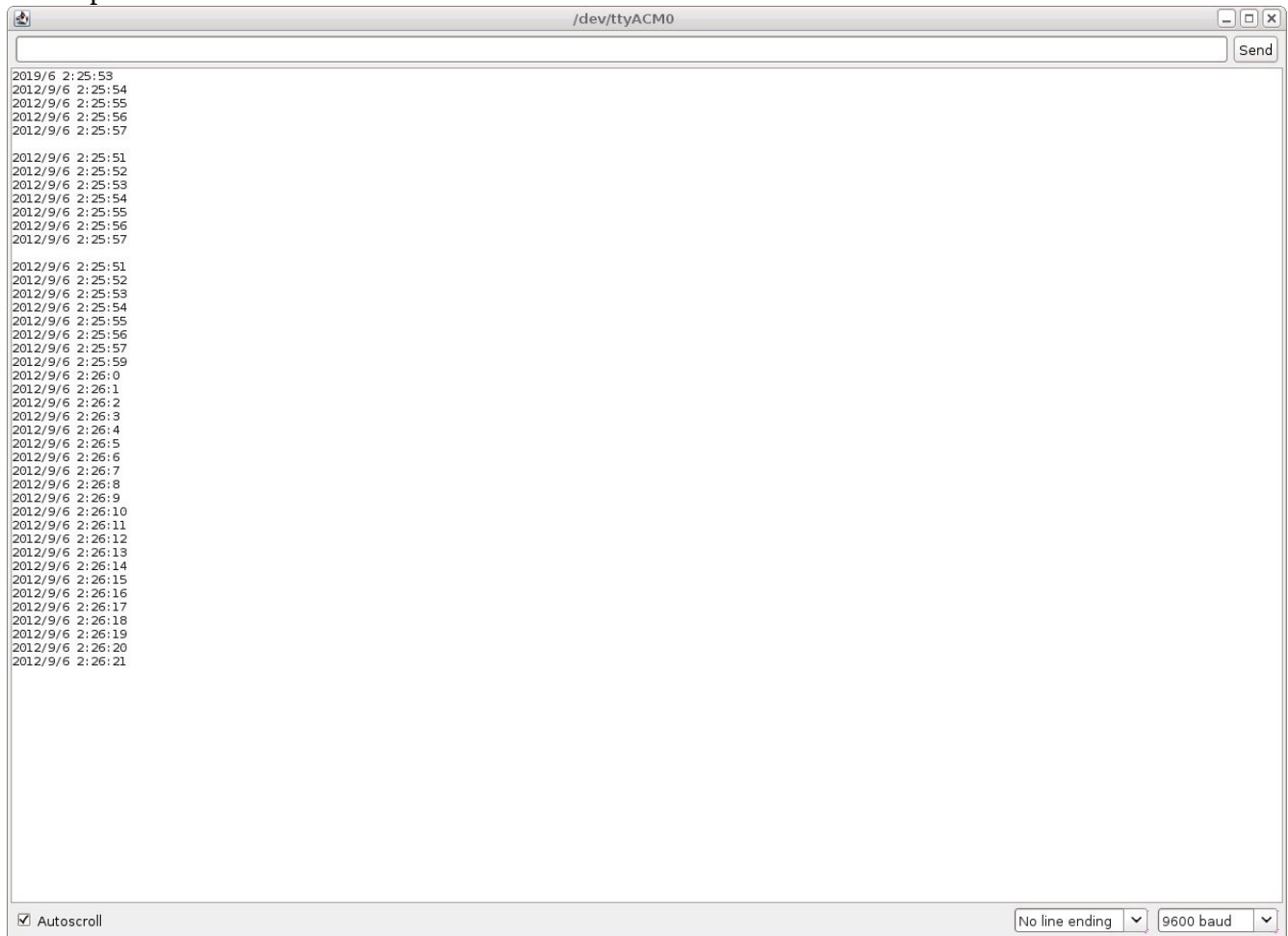
  RTC.adjust(DateTime(__DATE__, __TIME__));
}

void loop () {
  DateTime now = RTC.now();

  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
  delay(1000);
}
```

Upload deze code naar de arduino met data logger shield en start de serial monitor.

De output zou er zo uit moeten zien:



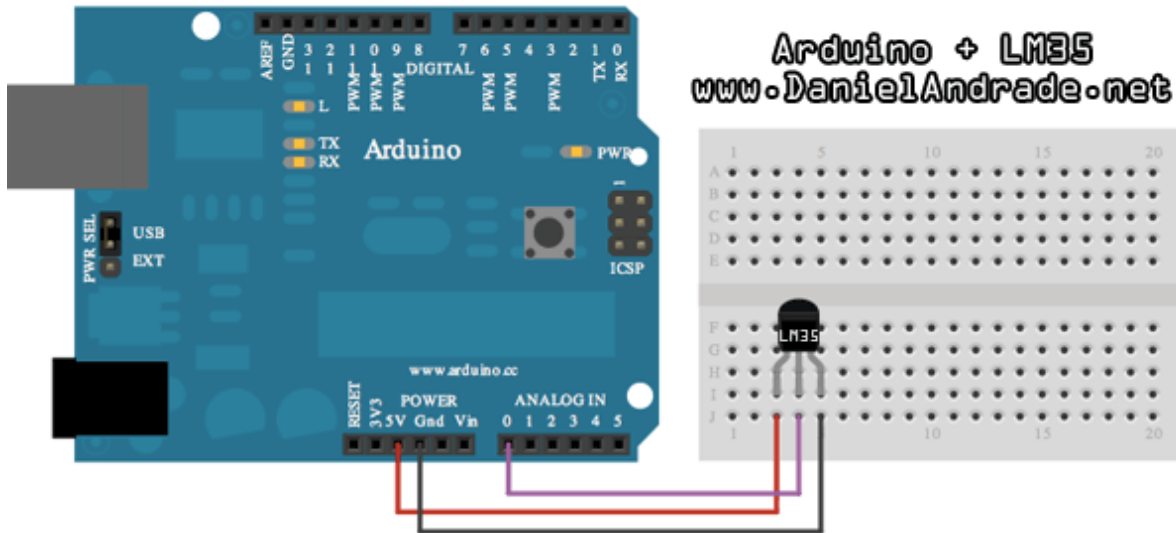
```
2012/9/6 2:25:53
2012/9/6 2:25:54
2012/9/6 2:25:55
2012/9/6 2:25:56
2012/9/6 2:25:57
2012/9/6 2:25:51
2012/9/6 2:25:52
2012/9/6 2:25:53
2012/9/6 2:25:54
2012/9/6 2:25:55
2012/9/6 2:25:56
2012/9/6 2:25:57
2012/9/6 2:25:51
2012/9/6 2:25:52
2012/9/6 2:25:53
2012/9/6 2:25:54
2012/9/6 2:25:55
2012/9/6 2:25:56
2012/9/6 2:25:57
2012/9/6 2:25:59
2012/9/6 2:26:0
2012/9/6 2:26:1
2012/9/6 2:26:2
2012/9/6 2:26:3
2012/9/6 2:26:4
2012/9/6 2:26:5
2012/9/6 2:26:6
2012/9/6 2:26:7
2012/9/6 2:26:8
2012/9/6 2:26:9
2012/9/6 2:26:10
2012/9/6 2:26:11
2012/9/6 2:26:12
2012/9/6 2:26:13
2012/9/6 2:26:14
2012/9/6 2:26:15
2012/9/6 2:26:16
2012/9/6 2:26:17
2012/9/6 2:26:18
2012/9/6 2:26:19
2012/9/6 2:26:20
2012/9/6 2:26:21
```

Nu gaan we verder met de SD kaart:

De SD kaart kunnen we gebruiken om data op te slaan en terug te lezen.

Voor deze opdracht gebruiken we net als in opdracht 3 een temperatuur sensor.

Sluit de sensor aan zoals in opdracht 3.



We gaan nu de data die we uitgelezen hebben van de temperatuur sensor opslaan op de SD kaart. Natuurlijk loggen we hierbij ook de tijd zodat we later terug kunnen lezen wat de temperatuur is geweest op een bepaalde tijd:

De code hiervoor is te groot om even over te typen dus deze staat op de USB stick (opdracht 4_2) en komt direct van de website van Adafruit:

In de serial monitor kun je zien welke data er weg geschreven wordt en wat de bestandsnaam op de SD kaart is.

Koppel eerst de USB aansluiting naar de laptop los.

Neem dan de SD kaart uit de shield en stop deze in de laptop (mocht er geen SD kaartlezer in de laptop zitten vraag dan om een kaartlezer).

Op de SD kaart staat een file die overeen komt met de filenaam die je in de serial monitor gelezen hebt. Open deze file op de laptop.

Op de site van Adafruit staat hoe je deze data kunt verwerken in bijvoorbeeld excel.

Dit wordt te specifiek voor deze workshop maar mocht je hierin geïnteresseerd zijn ga dan naar:

<http://www.ladyada.net/make/logshield/lighttemp.html>

Dit was de arduino workshop voor beginners V2.

Je hebt nu voldoende informatie om zelf je eigen projecten te kunnen maken.

Als er tijd over is zou je nog wat opdrachten kunnen combineren.

Denk bijvoorbeeld aan temperatuur en tijd weergeven op een LCD display.

Wij hebben in TKKRLAB nog veel meer hardware beschikbaar om je projecten nog verder uit te breiden.

Ook hebben wij de mogelijkheid om je project om te zetten van een prototype op een arduino bord tot een gewone print die je in je project kunt verwerken.

Dus kom gerust een keer langs om te kijken wat de mogelijkheden zijn.

www.tkkrlab.nl