1 Weerstation

Inleiding Ons weerstation werkt en geeft ons de data die we willen van het weer. Het is echter nog wel afhankelijk van een fysieke verbinding met de pc. Omdat we in sommige gevallen op afstand willen meten, kan het handig zijn om de data van het weerstation wireless naar de PC te versturen. Dit heeft een aantal voordelen. We hoeven geen kabel tussen PC en Arduino te trekken en we kunnen over grote afstand een nieuw programma zenden naar de Arduino. Het nadeel dat we wel constante voeding nodig hebben bij het station (als er geen voeding beschikbaar), zouden we kunnen ondervangen met een zonnecel en backup batterij.

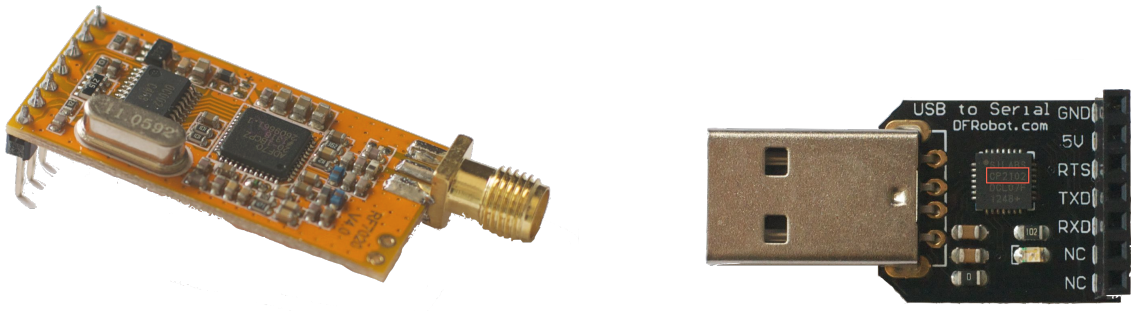
Benodigdheden Om het geheel wireless te laten werken hebben we een aantal zaken nodig. Het belangrijkste is de zend/ontvangmodule (APC220) module. Dit is een zend/ontvangmodule die makkelijk in te stellen is en makkelijk via een ander programma uit te lezen is.

- APC220 zendmodules (2x)
- Arduino software
- USB to UART adapter
- APC programma

2 APC module installeren

2.1 Zenden en ontvangen

Telemetrie is een technologie, die ervoor zorgt dat data verzonden en ontvangen wordt. De data wordt verzonden via radiogolven en door ontvangers opgevangen, bewaard en/of bewerkt. Ons weerstation moet de data van de sensoren doorsturen naar de HiSPARC computer op school, zodat deze de data kan versturen. De APC-220 module (zie Figuur 1.1a), die we gebruiken is een radio zender of ontvanger met een laag vermogen. Een van de modules gebruiken we als ontvanger, die wordt dan ook aangesloten op de pc. De andere wordt aangesloten op de Arduino en zal als zender fungeren. Het bereik van de zender is 1 km in open lucht. De zender kan door muren heen zenden, maar het verdient aanbeveling om het aantal muren tussen het meetstation en de computer beperkt te houden. De zender kan zenden tussen 430 MHz tot 440 MHz. De default waarde is echter 434 MHz. In tabel zie je mogelijke waarden die we gebruiken, de APC220 kan zenden tussen 418 MHz en 454 MHz. In Nederland mag er echter alleen tussen deze frequentiegrenzen door zendamateurs informatie verzonden worden.



(a) **(b)**

Figuur 1.1 – Een APC zend of ontvanger module (a) en zijn USB adapter (b), in het rode kader het nummer van de chip (CP210), check of dit nummer in de driver beschrijving staat bij het downloaden van de driver van SILABS.

2.2 APC-220 installeren

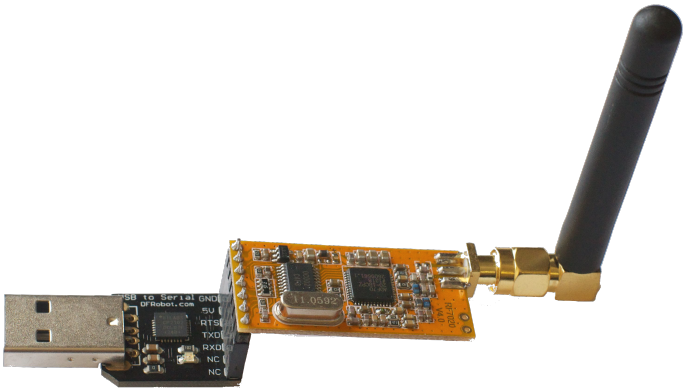
De ontvanger zal gekoppeld moeten worden aan de computer. De computer heeft een driver nodig om te werken. We hebben een VCP (Virtual Com Port) driver nodig, die we kunnen downloaden van Silabs.com: Gebruik 'google' met de zoekterm **'USB to UART Bridge VCP Drivers'** en kies de eerste optie die je krijgt. Download de juiste driver voor je besturingssysteem op de gebruikte computer en installeer deze. Zorg tijdens de installatie van de driver dat je de ontvangmodule nog niet in je pc hebt aangesloten. Na installatie sluit je de APC module Figuur 2.1 aan op een USB ingang van de PC. Als Windows hem dan niet vind, verwijder de module en stop hem er opnieuw in.

Om de APC module te configureren gebruiken we de Arduino en een programma 'apc220cfg.ino' dat we downloaden van de site van NAROM of van de universiteit van Aalborg [1] of van `https://github.com/HiSPARC/weather-arduino`.

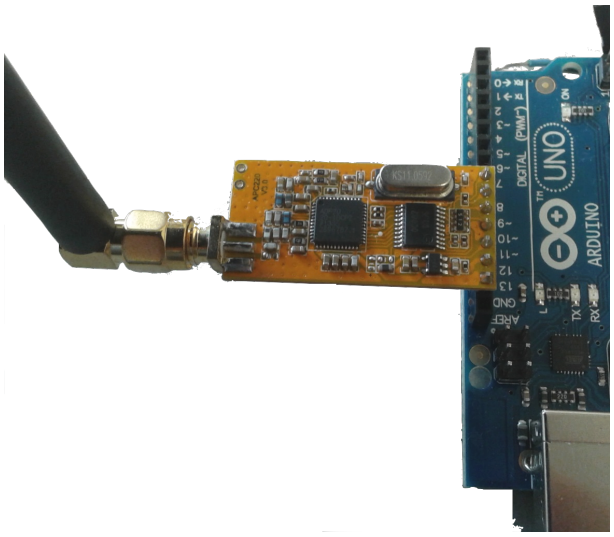
2.3 Configureren van de APC-220

Start nu Arduino software op en open het programma 'apc220cfg.ino'. We willen nu de frequentie gaan instellen waarmee de module zendt en ontvangt. Upload het programma naar de Arduino voordat je de APC module aansluit op de Arduino. Maak nu de USB kabel los en verwijder eventueel de voeding van de Arduino. Plaats nu de APC module op de Arduino zoals is aangegeven in Figuur 2.2. De APC moet zo geplaatst worden dat de pinnen van de APC in pin 8-13 en ground van de Arduino zitten. Sluit de Arduino nu opnieuw aan met behulp van de USB kabel en open de 'Serial Monitor' van Arduino. Type in de invoerbalk van de "Serial Monitor", 'm'. Dan opent het menu wat in Figuur 2.3 te zien is. Met dit menu kun je de APC-module configureren. Het invoeren van de letter 'r' geeft de huidige configuratie van de APC module weer. Bepaal met

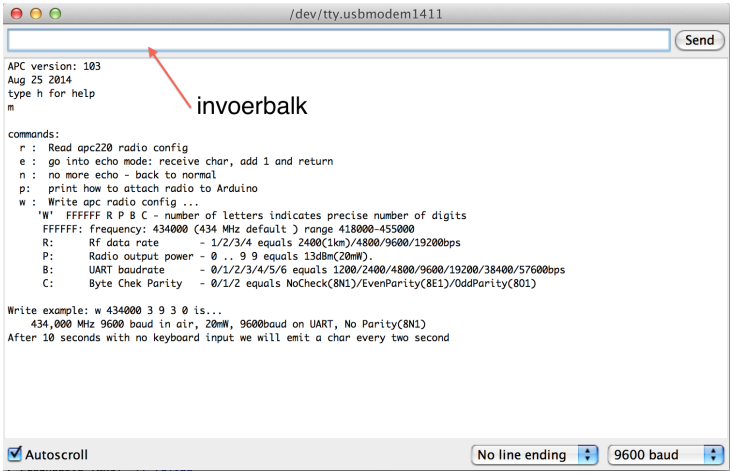
[1] NAROM is een raketbasis/organisatie in Noorwegen die participeert in CANSAT (een Europees project van de ESA). Zij hebben samen met de Aalborg universiteit op basis van Arduino een pakket ontwikkeld waarmee een CANSAT satelliet gemaakt kan worden. De CANSAT zendt draadloos data via de APC-220 module. Andere tools om de APC-module te configureren bestaan wel, maar geven veel problemen. Een voorbeeld van zo'n problematisch programma is "RF-Magic". Gebruik Google met zoekterm 'apc220cfg.ino'



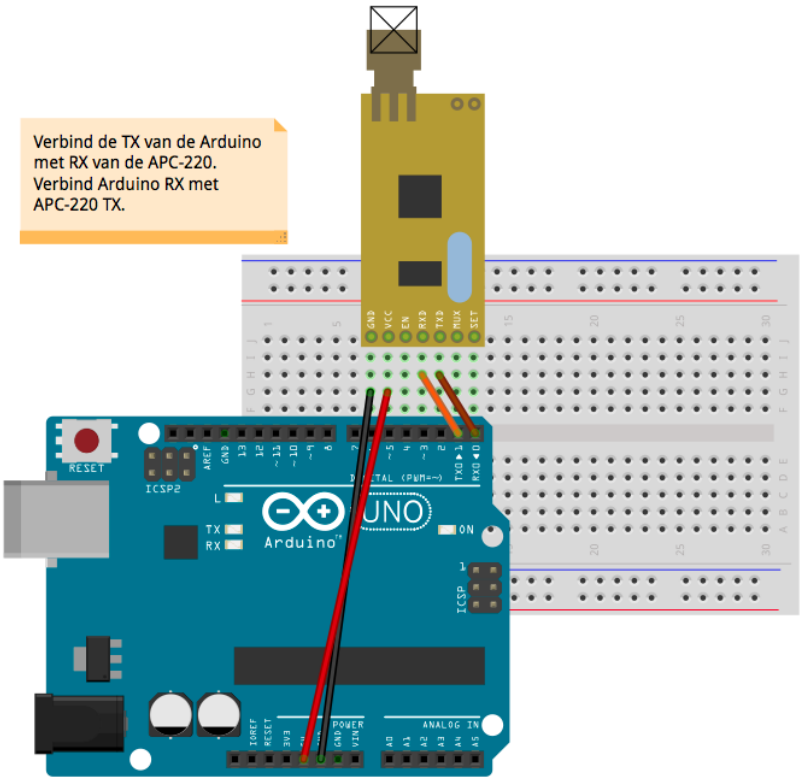
Figuur 2.1 – De APC module op de aangesloten op de USB adapter.



Figuur 2.2 – De APC module aangesloten op de Arduino om te configureren. Sluit de module aan op pin 8-13 en ground van de Arduino.



Figuur 2.3 – Screenshot van de "serial monitor", type 'm' in de invoerbalk van de serial monitor. Na het typen van 'm' wordt dit menu getoond. Type 'r' om de instellingen van de APC module uit te lezen. Type 'w' om de APC module configureren. Bepaal welke frequentie je wilt gebruiken voor transmissie en ontvangst uit tabel 1. Voorbeeld: w 434150 3 9 3 0 (denk aan de spaties) betekent: write frequentie 434 150 MHz met 9600 bps in lucht, met een zendvermogen van 20 mW ontvangen van data met 9600 bps over UART zonder parity check.



Figuur 2.4 – De APC module aangesloten op de Arduino om te zenden.

Frequentie (MHz)	Frequentie (MHz)
433050	433950
433100	434000
433150	434050
433200	434100
433250	434150
433300	434200
433350	434250
433400	434300
433450	434350
433500	434400
433550	434450
433600	434500
433650	434550
433700	434600
433750	434650
433800	434700
433850	434750
433900	434790

Tabel 1 – tabel met frequenties: Volgens [2] mag in Nederland dit bereik van radiofrequenties gebruikt worden voor telemetrie op een vermogen van 10 mW.

behulp van tabel 1 op welke frequentie je beide modules wilt laten werken. Beide modules moeten op dezelfde frequentie werken. Voer dan je keuze in. Bijvoorbeeld de invoer: w 434790 3 9 3 0 staat voor: write frequentie 434 790 MHz met 9600 bps in lucht en 20 mW 9600 bps wegschrijven over UART zonder parity check. Herhaal deze stap voor de andere APC module. Belangrijk is dat de Baudrate op 9600 staat en paritycheck 'none' is.

2.4 Zenden en ontvangen met de APC module

We hebben nu de modules zo geconfigureerd dat ze in principe klaar zijn om te zenden en ontvangen. Nu moeten we ons weerprogramma weer uploaden naar de Arduino. Sluit de USB kabel aan op de Arduino en upload het weerprogramma (Zorg dat er geen kabeltje in de Rx<-0 ingang van de Arduino zit, anders krijg je een foutmelding). Run het programma, kijk in de serial monitor van de arduino software of er data verschijnt. Sluit de serial monitor. We gaan nu wireless meten. Haal de usb kabel nu los.

Sluit een voeding aan op de Arduino. We sluiten de APC-modules aan zoals in Figuur 2.4 (let op de verbindingen RX $\longleftrightarrow$ TX) en plaatsen de APC module met UART in de USB ingang van de pc. De Arduino software zal de UART vinden en geeft ook de poort aan, waar deze zit. Ga naar *"Tools"*, *"Serial Port"*. Daar vind je de poort waar de UART zit, zorg dat je deze checkt. Het adres van deze poort is nodig om met een andere programmeertaal (in ons geval: Python) de gegevens van het weerstation uit te lezen. Op de Mac en Windows computers gaat dat net even anders, vandaar dat de onderstaande paragraven uitleg voor beide 'operating systems' bevatten.

2.4.1 Python voor Mac

Een pakket om Python in te draaien is bijvoorbeeld: Anaconda, kies Python 3. Dat is gratis te downloaden van: `https://www.anaconda.com/products/distribution#all` Download de free version en installeer deze. Je hebt dan een venster waar je kunt programmeren en een uitvoer venster.

Uitleg over hoe je moet werken met Anaconda Python 3 kun je op YouTube wel vinden.

Hieronder zie je een voorbeeld programma in Python, om data van het weerstation (voor de Mac) uit te lezen. Als een regel voorafgegaan wordt door een # dan is dat een commentaar regel en die wordt niet gecompiled en uitgevoerd.

```
# Code 1

# Code om de data van de APC220 op de Mac uit te lezen

import sys, serial
import time
from time import sleep

def main():
    strPort = '/dev/tty.SLAB_USBtoUART' # Adres van een APC-UART module voor Mac.
    ser = serial.Serial(strPort, baudrate=9600, parity=serial.PARITY_NONE,
                        bytesize=serial.EIGHTBITS, stopbits=serial.STOPBITS_ONE,
                        timeout=1.0)

    while True:
        try:
            line = ser.readline()
            print(line)
            time.sleep(10)
        except KeyboardInterrupt:
            print('exiting')
            break

if __name__ == '__main__':
    main()
```

2.4.2 Python voor Windows

Als je de wireless connectie maakt op een windowscomputer dan kun je waarschijnlijk de COM poort waar de UART zit niet vinden met de behulp van de Arduino software. Download Anaconda Python 3 voor Windows. Die is gratis te downloaden van: `https://www.anaconda.com/`

products/distribution#all Download de free version en installeer deze. Je hebt dan een venster waar je kunt programmeren en een uitvoer venster.

Uitleg over hoe je moet werken met Anaconda Python 3 kun je op YouTube wel vinden.

Eerst willen we vinden waar de COM poort van de UART zit. Gebruik het onderstaande stukje code.

```
# Code 2

# Code om COM poort van de UART op Windows te vinden.

import serial

def get_first_valid_port():
    # Windows
    found = False
    port_number = 0
    while not found and port_number < 256:
        try:
            s = serial.Serial(port_number)
            s.close()
            found = True
        except serial.SerialException:
            port_number += 1
    if found:
        return 'COM' + str(port_number + 1)
    return None

if __name__ == '__main__':
    poort = get_first_valid_port()
    print(poort)
```

Het stukje code om de informatie over de UART in te lezen (Windows): De port_id variabele heb je gevonden in het vorige stukje code (2). Dat is adres van de UART aansluiting.

```
# Code 3

# Code om de COM poort op de juiste manier te benaderen vul bij
# port_id het adres in wat in code (2) gevonden is.

import serial

port = serial.Serial(port_id, baudrate=9600, parity=serial.PARITY_NONE,
```

```
                    bytesize=serial.EIGHTBITS, stopbits=serial.STOPBITS_ONE,
                    timeout=1.0)
port.setRTS(0)
```

Het stukje code om daadwerkelijk data uit te lezen. De output leest de data in blokjes van 50 karakters en print deze als er data is. De library *msvcrt* moet je importeren met Python. `kbhit()` functie detecteert een keyboard input en de code stopt als op het keyboard de letter 's' wordt getypt. `getch()` functie haalt de waarde van de keyboard op en vergelijkt met de opgegeven 's'.

```
# Code 4

# Download data via APC-220

while True:
    output = port.read(50)
    if output:
        print(output)
    if msvcrt.kbhit() and msvcrt.getch() == 's':
        break
```

2.5 Uitlezen en wegschrijven van de data

We hebben nu een basisprogramma waarmee we de data van de Arduino kunnen uitlezen. Een aantal problemen steken nu de kop op. Het tempo waarmee de Arduino verzendt en de frequentie waarmee Python data uitleest zullen verschillen of uiteindelijk uit de pas gaan lopen. Ook is het niet interessant om meerdere keren dezelfde waarde van de temperatuur of luchtdruk te hebben. Het is interessanter als we alleen verschillen met de vorige waarden opslaan samen met een tijdstempel. Dit zullen we moeten opvangen met een stukje programmeercode. De oplossing van dit probleem kun je zelf bedenken. In deze paragraaf zullen we wel wat oplossingen aandragen. In Python kun je de data die je uitleest wegschrijven als een tekstfile. Hoe dat moet kun je hier lezen: `https://www.pythonforbeginners.com/files/reading-and-writing-files-in-python`

Voordat je wegschrijft moet je zorgen dat de nieuwe data vergelijkt met de oude data, als deze exact hetzelfde is als de vorige kun je in het programma aangeven dat je pas data wegschrijft als de data verschilt. Ook komt het wel eens voor dat de string met data leeg is. Dat wil je ook voorkomen. Daarnaast weet je dat de lengte van je data altijd hetzelfde moet zijn.

Gebruik hiervoor een 'if' statement met je voorwaarden. Bijvoorbeeld de volgende code: (code 5)

```
# Code 5

if nieuwe_data != vorige_data and len(nieuwe_data) > 32:
    with open("./Documents/Arduino/Metingen/Metingen.txt", "a") as data:
        data.write(output + "  {:%d, %b %Y} | ".format(datetime.date.today()) +
```

```
                    time.strftime("%H:%M:%S") + "\n")
    vorige_data = nieuwe_data
```

In de bovenstaande code staat in het *if* statement een controle op 3 punten. Namelijk of de `nieuwe_data` niet gelijk is aan de oude data, of `nieuwe_data` niet leeg is en het statement kijkt of de lengte van de data gelijk is aan het minimale aantal characters wat je nodig had voor bijvoorbeeld 3x een temperatuurmeting, luchtdruk en luchtvochtigheid. In ons geval is het minimale karakters in de string gelijk aan 32. Als we data naar de HiSPARC database gaan sturen zal het aantal karakters minder zijn. De file wordt elke keer geopend en gesloten als er data wordt weggeschreven, `"a"` zorgt ervoor dat er data toegevoegd wordt (**a**ppend) aan de textfile.

3 Wireless weerstation plaatsen

3.1 Behuizing en voeding Arduino weerstation

Het weerstation moet kunnen meten aan het weer, dus de behuizing moet voldoen aan de eigenschappen van een weerhuisje, zoals waterdicht zijn, wind doorlatend en niet in de volle zon zijn opgesteld. Daarbij komt dat de Arduino spanning nodig heeft. Als er buiten in de buurt van het weerstation een voeding is, dan is dit probleem zo opgelost. Anders kan een oplossing met een zonnecel en een oplaadbare batterij uitkomst bieden. Bijvoorbeeld de 'Solar MintyBoost', 'LiPo USB Charger Hookup' [2] of zelf een combinatie maken van een zonnecel en een oplaadbare batterij. Er zijn voor Arduino, wat printplaatjes te koop die een zonnecel combineren met een oplaadbare batterij zoals de 'USB LiPoly Charge'. Onze voorkeur gaat echter uit naar een kant en klare zonnecel met batterij met een USB uitgang, zodat we de Arduino via USB kunnen voeden. Let er op dat de zonnecel waterbestendig is, als dit niet zo is maak dan een constructie waardoor deze afgeschermd is van de weersinvloeden. De Arduino heeft bij (5V) een stroomsterkte van ongeveer 40 mA nodig per aangesloten pin, daarnaast 50 mA voor de 3,3 V uitgang. De zender (APC220) heeft 40 mA nodig. Als we uitgaan van metingen in de winter dan kunnen we op zonnecel ongeveer 8 uur licht verwachten. Dat betekent dat onze Arduino 24 uur moeten kunnen laten werken op een batterij die in korte tijd is opgeladen. Een snelle rekensom leert ons dat we dan een batterij met een capaciteit van rond de 2000 mA h nodig hebben. Een hogere capaciteit is wel aan te bevelen.

In het volgende document gaan we daadwerkelijk data naar de database van HiSPARC schrijven.

Referenties

[1] NAROM, 2013 *The CANSAT BOOK 2013-edition*, CANSAT project

[2] Agentschap Telecom, 2012 *Vergunningsvrije radiotoepassingen*, Ministerie van Economische zaken, Landbouw en Innovatie

[2] Deze producten kunnen gevonden door de naam te 'googlen.