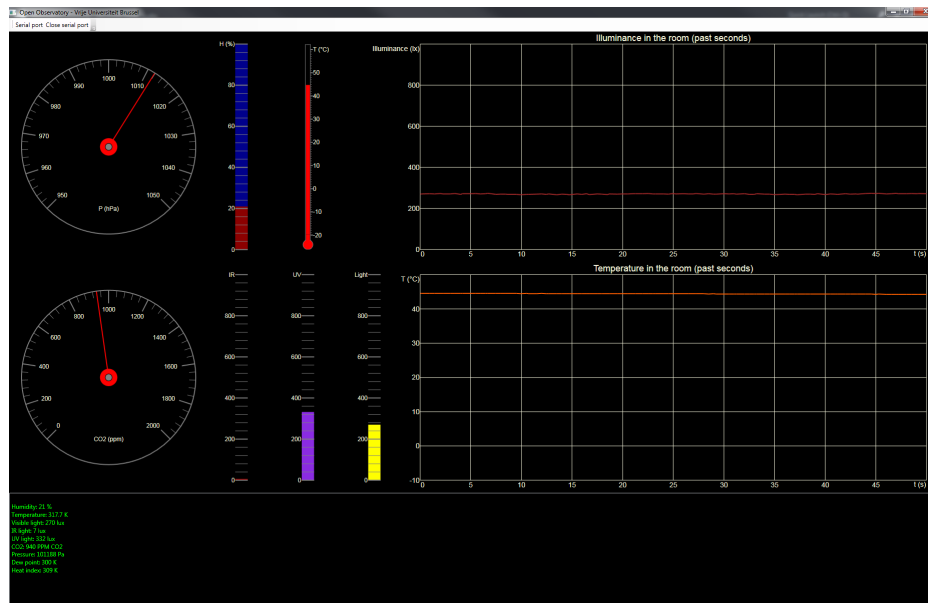


Objectgeoriënteerd Programmeren: Weerstation

1. WEERSTATION

Het klimaat is een dynamisch systeem dat continu evolueert. De planeet kent talloze verschillende weertypes met elk kenmerkende eigenschappen. Het voorspellen van het klimaat berust op het meten van een aantal grootheden zoals de luchtdruk, de vochtigheidsgraad, de temperatuur, de UV index enz. Hoewel het mogelijk is al deze parameters afzonderlijk op te meten en op te slaan, is het interessanter om de samenhang van al deze parameters samen te analyseren. Hiervoor worden de verschillende sensoren in één station ondergebracht: het weerstation. In deze opdracht zal je een interface schrijven die toelaat om de sensoren van het weerstation in te lezen, weer te geven in grafieken en op te slaan. Deze interface moet in staat zijn om de communicatie tussen de computer en het weerstation te kunnen verzorgen. De data moet op een correcte manier geïnterpreteerd worden zodat een correcte weergave mogelijk is. Een aantal grafieken en meters geven de huidige en voorbije situatie weer van de sensoren. Omdat het ook nuttig kan zijn om een tendens te bepalen uit een reeks metingen, wordt de gemeten data ook opgeslagen in bestanden. Een eenvoudig bestandstype dat hier gehanteerd kan worden is het CSV-formaat. De data terug uit een bestand kunnen inlezen is hierbij dus ook een must.



Figuur 1: Demo voorstelling van het weerstation.

2. OPENOBSERVATORY, EEN OPEN SOURCE PROJECT

De weerstations voor het project Objectgeoriënteerd Programmeren zijn de eerste 100 prototypes van een observatorium dat verder gaat dan enkel het klimaat. Op versie 3.0 zit bijvoorbeeld een sensor om de concentratie van gassen in de lucht te meten, zoals CO₂ bijvoorbeeld, maar ook een lichtsensor, een sensor voor infrarode (warmte) straling enz. OpenObservatory is een FOSS project (Free and Open Source Software), gecombineerd met open source hardware. Zowel desktop software als embedded firmware zijn beschikbaar op [de OpenObservatory GitHub repository](#). Revisie 3.0 van de hardware is beschikbaar op [CircuitMaker](#).

Alle software is beschikbaar onder [Creative Commons Attribution-ShareAlike-NonCommercial licentie](#), voor de hardware geldt de [CERN Open Hardware License](#).



Figuur 2: WeatherStation v3.0.

3. SENSORS

Documentatie voor WeatherStation v3.0 met firmware versie 3.0.4 (18 april 2016).

Revisie 3 is uitgerust met de nodige sensors om volgende omgevingsvariabelen te meten:

Omgevingsvariabele	Sensor	Bereik	Eenheid
Luchtvochtigheid	DHT11	20 - 90	RH%
Luchtdruk	BMP180	00 - 1100	hPa
UV-index	Si1145	0 - 12	
CO ₂ -concentratie	MQ135	100 - 1200	PPM
Infraroodstraling	Si1145	0 - 14000	W/m ²
Temperatuur	BMP180 ¹	-20 - 80	°C
Licht	Si1145	0 - 10000	lx

Tabel 1: Sensoroverzicht op revisie 3.0.

¹ Temperatuursensor kan gewijzigd worden.

3.1 Luchtvochtigheid

Om de luchtvochtigheid te meten is een Aosong DHT11 vochtigheidssensor voorzien (IC3). Deze sensor heeft een repeatability van $\pm 1\%$ en een fout van $\pm 5\%$ bij 25 °C. Deze sensor is behoorlijk traag, het kan tot 2s duren vooraleer wijzigingen in luchtvochtigheid opgepikt worden.

Bij 0 °C stijgt de minimaal meetbare luchtvochtigheid van 20 % naar 30 %. De DHT11 is uitgerust met een temperatuursensor voor vochtigheidscalibratie die optioneel kan geactiveerd worden via het commando

#: set TSENSOR auxiliary

De luchtvochtigheid wordt gerapporteerd als relatieve luchtvochtigheid, uitgedrukt als percentage van verzadigde lucht.

3.2 Luchtdruk

De luchtdruk wordt gemeten door een Bosch BMP180 druksensor (IC2). Deze piëzo-elektrische sensor meet de luchtdruk in een bereik van 300 hPa tot 1100 hPa met een resolutie van 6 Pa. De luchtdruk wordt gerapporteerd in Pa. De BMP180 is eveneens uitgerust met een temperatuursensor voor calibratiedoeleinden, die optioneel kan geactiveerd worden via het commando

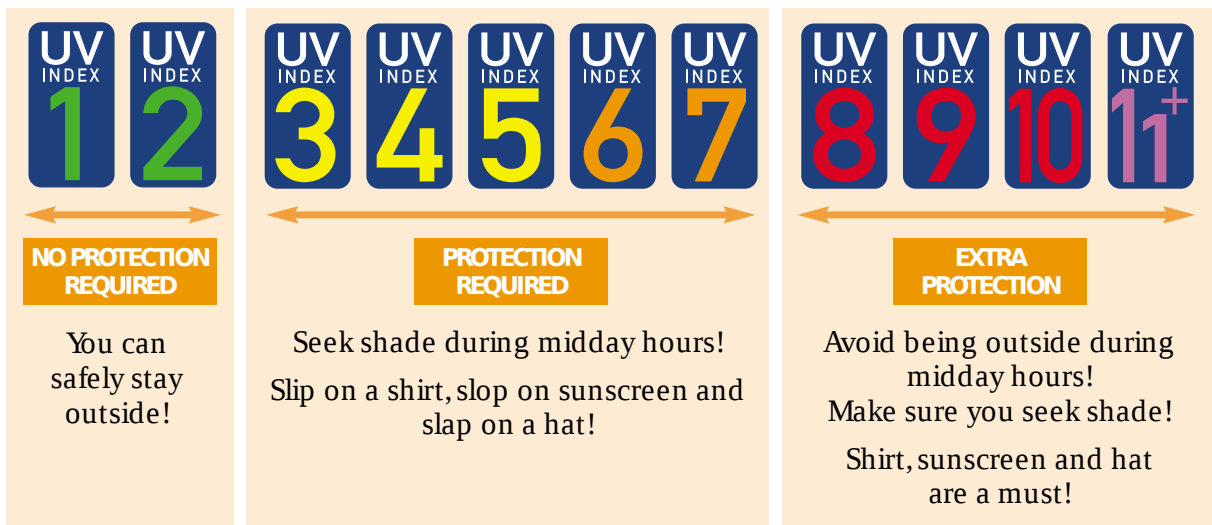
#: set TSENSOR secondary

Om correcte drukmetingen uit te voeren dient de sensor in vrij contact te staan met de buitenlucht. Het weerstation mag dus niet in een hermetisch gesloten ruimte geplaatst worden.

De sensor zelf is gevoelig voor licht. Er is een 3D-printbare cover voorzien die over de sensor kan geplaatst worden. Tekstopdruk op de bovenkant van het station wijst de positie van deze cover aan. De cover kan vastgestriipt worden met een snelbinder, gebruik makend van de 2 gefreesde slots aan weerszijde van de sensor.

3.3 UV-index

De UV-index is een genormaliseerde parameter, en wordt gerapporteerd als een dimensieloos getal tussen 0 en 12 waarbij 0 de kleinste en 12 de grootste intensiteit ultraviolet licht voorstelt. De betekenis wordt gedefinieerd volgens de richtlijnen van de WHO volgens figuur 3.



Figuur 3: Interpretatie van de UV-index.

Bron: Sloss, P., *Global Solar UV Index: A Practical Guide*, p. 8, ISBN 91 4 159007 6, World Health Organization (WHO), 2002.

3.4 CO₂-concentratie

Het station is voorzien van een socket waarin een gassensor naar keuze (IC1) kan gemonteerd worden. De standaardsensor is een MQ135 die gevoelig is voor CO₂ en waarmee dus de luchtkwaliteit kan gemeten worden. De concentratie CO₂ in de lucht wordt gerapporteerd als PPM (parts per milion). De concentratie CO₂ in buitenlucht neemt continu toe onder invloed van luchtvervuiling, en bedraagt anno 2016 ca. [400 ppm](#).

De sensorsocket maakt gebruik van *friction clamps* waardoor er geen crimpcontacten of soldeerverbindingen nodig zijn om sensors in de socket te monteren. De MQ135 CO₂-sensor is symmetrisch en kan dus op 2 manieren gemonteerd worden.

Net als alle andere sensors uit de MQ-serie is ook de MQ135 een elektrochemische sensor. De sensor moet gedurende 24 uren opgewarmd worden vooraleer de kalibratie kan gestart worden (*eng. burn-in time*).

Om een meting uit te voeren wordt de sensor intern opgewarmd met een verwarmingselement, en het is perfect normaal dat ook de buitenkant daardoor warm wordt. Als het host OS een *suspend* commando naar het station stuurt, bv. wanneer de cover van de laptop wordt gesloten, dan wordt het verwarmingselement standaard automatisch uitgeschakeld om de levensduur van de sensor te optimaliseren. Het verwarmingselement kan ook manueel bestuurd worden via het commando #: *set HEATER*:

#: *set HEATER on*

#: *set HEATER off*

#: *set HEATER auto*

In “auto” modus wordt het verwarmingselement bestuurd door het suspend signaal van het host OS.

Als het station net is ingeschakeld dan zal de gemeten CO₂-concentratie afwijken van de reële waarde omdat de sensor enige minuten tijd nodig heeft om op temperatuur te komen.

3.5 Infraroodstraling

Infraroodstraling komt overeen met infrarode deel van het elektromagnetische spectrum en is een maat voor invallende warmtestraling. De waarde wordt gerapporteerd in W/m², en correspondeert met het invallend vermogen van een infrarode stralingsbron op een oppervlak van 1 m².

Het station is uitgerust met een stralingssensor (IC4) waarmee infrarode straling kan gemeten worden. Dit laat toe om de temperatuur van objecten op afstand te meten, aangezien de uitgestraalde energie evenredig is met de temperatuur. De sensor kan dus ingezet worden om stralingsbronnen te detecteren (bv. bewegingsmelder, branddetector enz.) en om hun temperatuur te schatten.

3.6 Temperatuur

Het station beschikt over 3 onafhankelijke temperatuursensors waarvan IC4 de meest stabiele waarde geeft (primary) bij een resolutie van ±1 °C. Er is ook een secundaire sensor met hogere nauwkeurigheid beschikbaar (IC2) die echter lichtgevoelig is. Een opake cover is noodzakelijk om de hoogste nauwkeurigheid van ±0.5 °C te halen. Deze cover kan geprint worden met elke 3D-printer. Om de invloed van lichtdoorlatende materialen zoals lichtgekleurde PLA of ABS te vermijden is het adviseerbaar om met 100 % infill te printen. De printtijd bedraagt ca. 15 minuten.

Wanneer primaire en secundaire temperatuursensors sterk afwijkende metingen rapporteren, is er nog een derde hulpsensor voorzien die [een indicatie kan geven welke sensor correct is](#). Elke

sensor kan geselecteerd worden als de standaard temperatuursensor met het commando `#: set TSENSOR` gevolgd door de gewenste sensor. De 3 commando's zijn respectievelijk

`#: set TSENSOR primary`
`#: set TSENSOR secondary`
`#: set TSENSOR auxiliary`

De temperatuurmetingen worden gerapporteerd in °C, waarbij 0 K = -273.15 °C.

3.7 Licht

Het station heeft een Si1145 lichtsensoren aan boord (IC4) waarmee invallend omgevingslicht kan gemeten worden. De sensor rapporteert de invallende lichtsterkte in lux (lx), een eenheid die representatief is voor lichtsterkte zoals opgevangen door menselijke ogen. Tabel 2 geeft een overzicht van gangbare intensiteitsniveaus:

Lichtintensiteit	Bewolking
128 klx	zonneconstante
32 klx - 100 klx	direct zonlicht op de middag
10 klx - 25 klx	daglicht, geen directe zon
1 klx	bewolkt
400 lx	zonsopgang of zonsondergang, heldere hemel
100 lx	zwaarbewolkt
0.25 lx - 1.0 lx	volle maan, heldere hemel

Tabel 2: Gemeten lichtintensiteitsniveau voor diverse atmosferische omstandigheden.

Voor een correcte meting van de lichtsterkte is een loodrechte positionering ten opzichte van de lichtbron vereist, en mogen er geen lichtabsorberende materialen (plexiglas bv.) tussen zitten. De sensor is gecompenseerd voor interferentie door de LED's op het station zelf. Elke verandering aan de LED-configuratie op het station vereist dus een recalibratie van de lichtsensoren.

4. SERIËLE COMMUNICATIE

De communicatie tussen de hardware en de computer gebeurt via de seriële poort. De instellingen van deze communicatie worden weergegeven in tabel 3.

Eigenschap	Waarde
Baudrate (bitrate)	115200
Pariteit	geen
Startbits	1
Stopbits	1
Databits	8

Tabel 3: Eigenschappen van de seriële communicatie tussen de hardware en de computer.

De bit rate is een parameter die in de firmware kan aangepast worden in de header `02_Rev3.h` onder `SERIAL_SPEED`. De standaardwaarde is 115.2 kHz.

De *frame rate* is onafhankelijk van de *bit rate*, en bepaalt het aantal doorgezonden frames per seconde. Elk frame beslaat metingen van alle sensors. De frame rate kan ingesteld worden via de `set TXINT` instructie, gevolgd door een time-out in ms. Het commando

```
set TXINT 250 ms
```

levert bijvoorbeeld om de 250 ms een dataframe, wat overeenkomt met een updatefrequentie van 4 Hz. De maximale frequentie wordt begrensd door de snelheid van de seriële communicatie zelf. De maximale periode is ca. 1 minuut.

Het weerstation stuurt de gemeten waarden van de sensoren over de seriële communicatie in een tekstuele vorm. Deze tekstuele informatie wordt weergegeven in codefragment 1 (firmware 3.0.4):

Codefragment 1: Output van de seriële poort.

```

1 Humidity: 33 %
2 Temperature: 25.7 deg C
3 Visible light: 112 lx
4 Infrared radiation: 1802.5 mW/m2
5 UV index: 0.12
6 CO2: 404 ppm CO2
7 Pressure: 102126 Pa
```

Om deze tekstuele informatie te kunnen interpreteren is het dus noodzakelijk om deze waarden om te zetten in numerieke waarden. Zorg ervoor dat elk tuppel afgezonderd kan worden zodat achteraf de numerieke waarde aan de grootheid gekoppeld kan worden. Schrijf de nodige conversieroutines om bij te houden welke waarde bij welke grootheid past. Merk op dat bij elke waarde ook een tijdstip past. Hiervoor kan je gebruik maken van het *DateTime*-object in C#. **Hint:** Voorzie ook de nodige beveiligingen opdat de interface niet vastloopt wanneer er zich een probleem met de hardware voordoet (bv. uittrekken USB-kabel, resetknop indrukken e.d.).

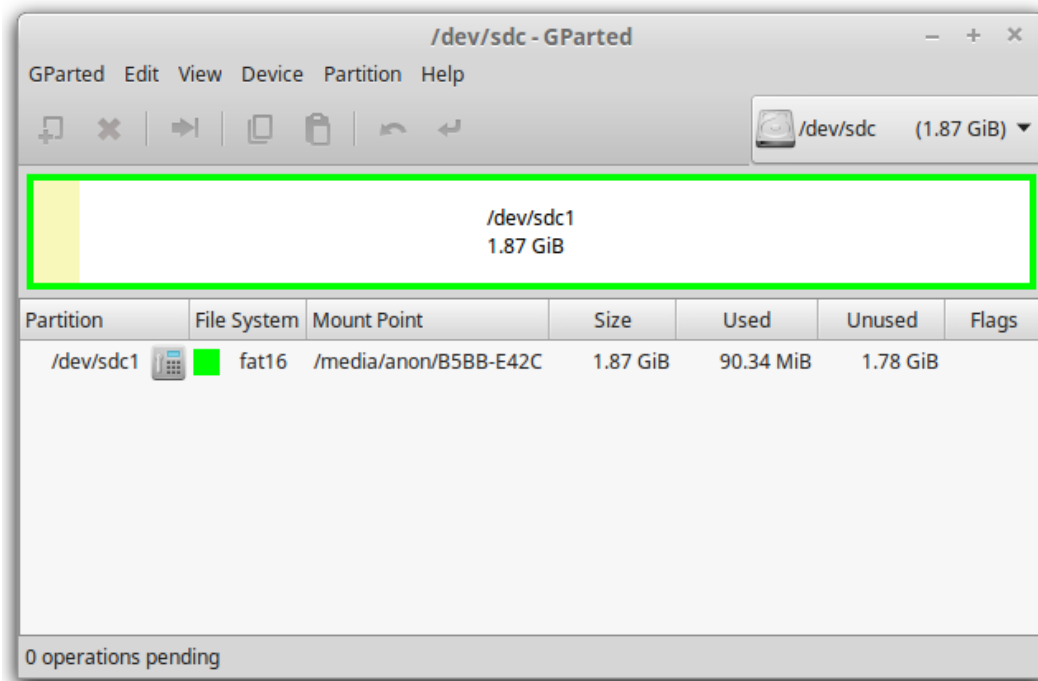
De seriële verbinding wordt ook gebruikt om commando's te sturen naar het station, om bijvoorbeeld de frame rate in te stellen, temperatuursensor te selecteren etc. De verbindingsparameters zijn identiek aan die in tabel 3. Commando's worden gestuurd in ASCII met een unix line ending. Line endings komen overeen met de karakters `'\r'` en `'\n'` (karakters 10 en 13 in

C#).

De seriële verbinding vereist van het C# programma dat data op een willekeurig moment ingelezen kan worden. Een mogelijk manier hiervoor is om het programma permanent op inkomende data te laten wachten. Hierdoor blokkeert men de voortgang van het programma waardoor het weergeven van de data op een gebruikelijke manier niet mogelijk is. Om dit te verhelpen kan men de asynchrone functies van de seriële poort binnen C# gebruiken. Hierdoor wordt de data in een aparte thread ontvangen terwijl de weergave van data mogelijk blijft (en de gebruiker met de applicatie kan verderwerken). Om data van deze thread naar de grafische user interface (GUI) te brengen moet men de data eerst bufferen. Na het bufferen kan de data via een timer naar het scherm geprojecteerd worden. Deze techniek is enkel noodzakelijk indien men data van een achtergrondthread naar de GUI wil overbrengen. In het geval dat men data onmiddellijk naar een bestand wenst weg te schrijven hoeft men deze techniek niet toe te passen. Voor meer informatie kan er gezocht worden op *cross-thread event handling*.

5. SD-KAART

Het station is uitgerust met een SD-kaartslot waarmee het zelfstandig data kan loggen. Elke SD-kaart met een primaire partitie in FAT16 bestandssysteem die kleiner is dan 4 GB kan gebruikt worden. De partitie hoeft niet leeg te zijn. Een bestaande SD-kaart kan opnieuw gepartitioneerd worden met `gparted` zoals te zien is in figuur 4. Let op: bij het partitioneren en formatteren van SD-kaarten wordt alle bestaande data gewist!



Figuur 4: Een 2 GB SD-kaart die als FAT16 bestandssysteem geformatteerd is in gparted.

Het station kan data loggen zonder met de pc verbonden te zijn. De energievoorziening verloopt in dat geval met een USB charger kabel zonder dataverbinding met een voedingsspanning van $5\text{ V} \pm 0.25\text{ V}$.

Het loggen van data kan aan- en uitgeschakeld worden via volgende seriële commando's: `logging enabled` om data logging in te schakelen en `logging disabled` om data logging weer uit te schakelen.

Voor elke sessie wordt een afzonderlijk bestand aangemaakt in de root directory van de SD-kaart met een CSV indeling (`comma separated values`).

In revisie 3.0 is FAT het enige ondersteunde bestandssysteem. De functionaliteit om data te loggen moet afzonderlijk geactiveerd worden via de preprocessor switch `#define SD_LOGGING`

6. FIRMWARE

De centrale besturingseenheid van het station is een ATmega328au microcontroller die geprogrammeerd is met een gemodificeerde Arduino-bootloader. Het station is bijgevolg 100% compatibel met de Arduino programmeeromgeving, en biedt gebruikers de mogelijkheid om aangepaste firmware te ontwikkelen en te debuggen zonder dat daarvoor een (dure) programmer nodig is. De bootloader zelf kan niet verwijderd worden.

Het downloaden van nieuwe firmware kan op 3 manieren:

- via de zwarte programmeerconnector K1 (programmer of ander station vereist);
- via de USB-verbinding met de host;
- via de uitbreidingsconnector K3.

Het station is zodanig ontworpen dat het downloaden van verkeerde firmware niet kan leiden tot onherroepelijke schade aan de hardware.

De *board definition file* `02_Rev3.h` is net zoals de laatste versie van de firmware, de softwaredrivers voor de sensors en een zelftestprogramma te vinden op [de GitHub repository](#).

Voor het flashen van firmware via USB en Arduino IDE wordt Linux Mint of Ubuntu aanbevolen. Er zijn connectiviteitsissues onder Microsoft Windows door een bug in de seriële driver van Windows.

7. CONNECTIVITEIT

Het station beschikt over 3 connectors en verschillende testpunten. De connectors zijn respectievelijk ICSP (K1), USB (K2) en UART (K3). De testpunten zijn van belang voor

kwaliteitscontrole tijdens productie, en niet bedoeld om door de eindgebruiker gemanipuleerd te worden.

7.1 ICSP

De zespolige zwarte ICSP-connector (K1) biedt rechtstreekse toegang tot de programmeerinterface van de microcontroller. De pinout is als volgt:

De ICSP-connector is compatibel met zespolige JTAG-adapters, AVR-programmers en de JTAG ICE MKII. Enkel programmers met signaalniveaus tussen 3.0 V en 3.5 V kunnen toegepast worden. Tijdens het programmeren moet de SD-kaart fysiek uit het SD-kaartslot verwijderd worden. Via de power pin kan de ICSP-connector op 3.3 V tot 250 mA voeding leveren aan aangesloten programmers. De power pin is kortsluitvast. De locatie van pin 1 in de connector is aangeduid met een witte pijl op de witte tekstopdruk.

7.2 USB

De USB-connector (K2) dient zowel voor stroomvoorziening als data streaming.

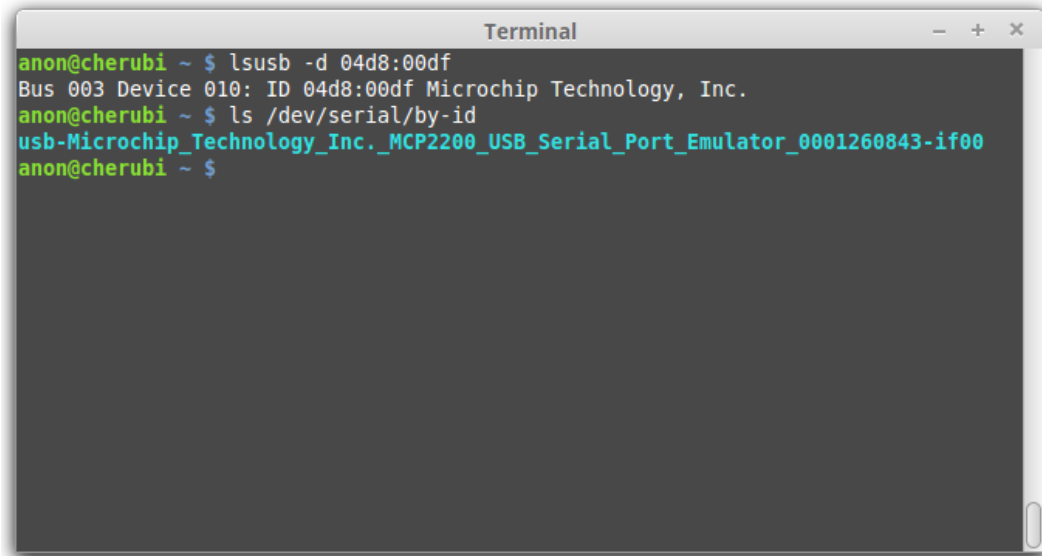
Wanneer het station verbonden is met een computer, dan kan er data uitgewisseld worden. Het station schakelt automatisch in broadcast modus, en stuurt op vaste intervallen dataframes naar de computer. Omgekeerd kunnen er ook instructies ontvangen worden. LED's D4 en D5 visualiseren data die wordt uitgewisseld tussen het station en de computer. Er wordt enkel data gestuurd als de socket aan de computerzijde geopend is, m.a.w. als er niet actief geluisterd wordt dan worden er geen dataframes gestuurd.

Het station enumereert zichzelf als een serieel device dankzij het geïntegreerde MCP2200 USB driver IC. Het IC heeft VID 04D8 (Microchip) en PID 00DF (MCP2200 USB Serial Port Emulator). De meeste moderne besturingssystemen hebben reeds een passende driver aan boord, dus het is niet nodig om extra driversoftware te downloaden vooraleer het station kan gebruikt worden. De correcte enumeratie van het station kan geverifieerd worden via de terminal aan de hand van het VID/PID paar:

Binnen Windows is het echter noodzakelijk om een driver voor de MCP2200 te installeren om ervoor te zorgen dat de MCP2200 zichtbaar wordt gemaakt als een virtuele seriële poort (COM poort). Meerdere stations kunnen tegelijk verbonden zijn zonder dat dit conflicten geeft.

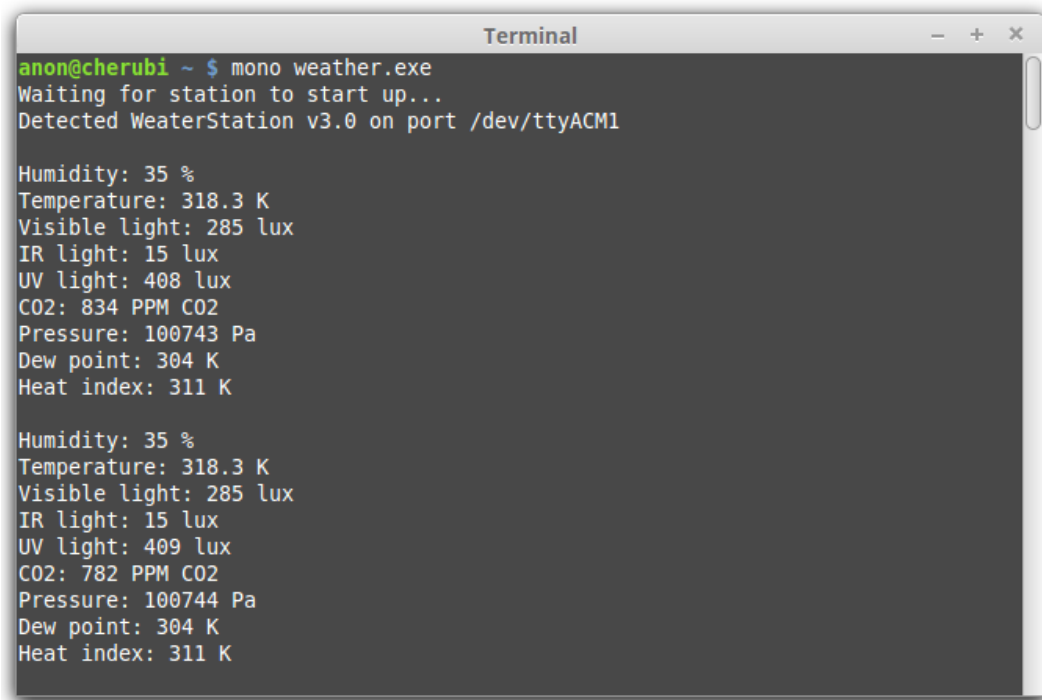
Op logisch niveau gedraagt het station zich als een seriële terminal (tty onder unix/linux) waaruit seriële data kan gelezen worden, en op dezelfde manier kan geschreven worden. Aancheftingspunten hangen af van de configuratie, maar gebruikelijk is dit `ttyACMx` met $x \geq 0$.

Er is een voorbeeldprogramma beschikbaar in C# dat het uitlezen van sensordata demonstreert. Zowel code als binary is beschikbaar op de [GitHub repository](#). Figuur 6 toont het station in actie.



```
Terminal
anon@cherubi ~ $ lsusb -d 04d8:00df
Bus 003 Device 010: ID 04d8:00df Microchip Technology, Inc.
anon@cherubi ~ $ ls /dev/serial/by-id
usb-Microchip_Technology_Inc._MCP2200_USB_Serial_Port_Emulator_0001260843-if00
anon@cherubi ~ $
```

Figuur 5: Aanhechting van de MCP2200 als serieel device in de kernel van Linux Mint 17.2 Rafaela.



```
Terminal
anon@cherubi ~ $ mono weather.exe
Waiting for station to start up...
Detected WeaterStation v3.0 on port /dev/ttyACM1

Humidity: 35 %
Temperature: 318.3 K
Visible light: 285 lux
IR light: 15 lux
UV light: 408 lux
CO2: 834 PPM CO2
Pressure: 100743 Pa
Dew point: 304 K
Heat index: 311 K

Humidity: 35 %
Temperature: 318.3 K
Visible light: 285 lux
IR light: 15 lux
UV light: 409 lux
CO2: 782 PPM CO2
Pressure: 100744 Pa
Dew point: 304 K
Heat index: 311 K
```

Figuur 6: WeatherStation v3.0, firmware 3.0.3 op seriële poort `/dev/ttyACM1` streamt sensordata naar de terminal.

Wanneer de datalijnen van de USB-kabel niet doorverbonden worden, zal het station geen data proberen te sturen. Een USB charger kabel kan dus gebruikt worden om het station te

voeden wanneer er zelfstandig data wordt gelogd op de SD-kaart.

7.3 UART

Een UART-verbinding (Universal Asynchronous Receiver and Transmitter) is beschikbaar op connector K3. K3 is een standaard 4-polige JST B4B-XH-A connector. De mating plug voor deze connector is JST XHP-4, beschikbaar op DigiKey als 455-2267-ND en via Farnell/Newark als 1516266.

De UART-verbinding gebruikt signaalniveaus van 3.3 V maar is 5 V tolerant. De volgorde van de pinnen staat bij de connector aan de onderkant van het station. Via de 3.3 V verbinding kan het station 250 mA stroom leveren aan externe hardware zoals extra sensors.

De seriële parameters voor de UART-verbinding zijn dezelfde als die voor de USB-verbinding, zie sectie *seriële communicatie*.

De UART-verbinding kan ook gebruikt worden om firmware te downloaden naar het station met behulp van de geïntegreerde bootloader. Om het reset-sigitaal te geven kan de reset-knop worden gebruikt, of het Reset testpunt naast deze knop. Reset is actief laag.

8. WEERGAVE

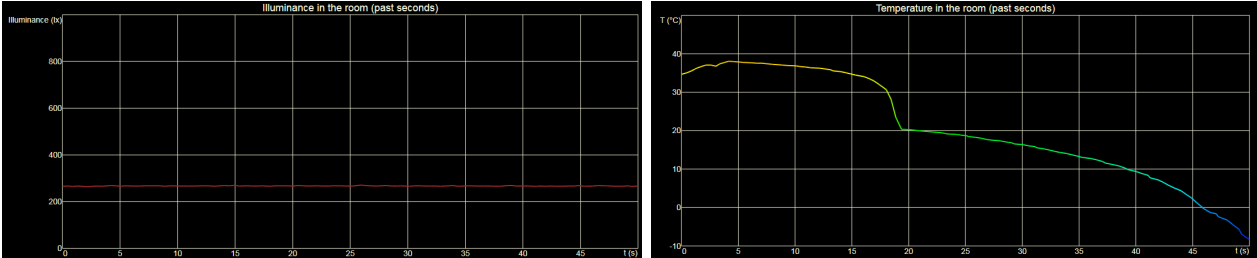
Eenmaal de interface de numerieke waarden bekomen heeft, moeten deze gegevens op een eenvoudige visuele manier aan de gebruiker voorgesteld worden. Hierbij kan gebruik gemaakt worden van grafieken en meters. Grafieken geven het verloop van de metingen in de tijd weer terwijl de meters de huidige waarden weergeven. Een aantal verschillende type grafieken en meters kunnen hierbij gehanteerd worden.

- De gewone grafiek: deze geeft het verloop van de metingen t.o.v. de tijd. De kleur van de grafiek kan bepaald worden a.d.h.v. properties (figuur 7).
- Een gradiëntgrafiek: deze verschilt t.o.v. de gewone grafiek doordat een kleurengradiënt (figuur 7) gebruikt wordt. Een voorbeeld gradiënt is de “Jet”-schakering die o.a. in Matlab gebruikt wordt. Dergelijke grafieken kunnen gebruikt worden om temperaturen weer te geven.
- Een ronder meter (figuur 8): kan gebruikt worden als barometer.
- Een thermometer (figuur 8).
- Een lineaire schaal (figuur 8): kan gebruikt worden om verhoudingen weer te geven zoals de relatieve luchtvochtigheid.

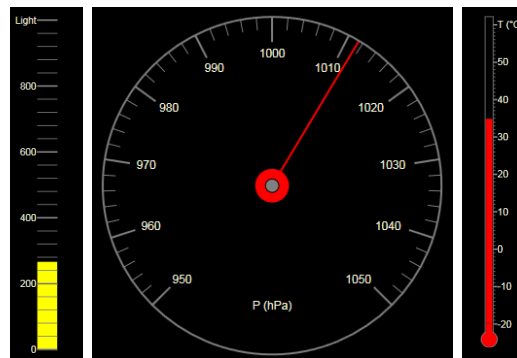
Deze lijst is zeker niet exhaustief. Deze grafieken en meters gebruiken allemaal een teken-interface waarop de nodige lijnen en oppervlaktes aangebracht. Andere gemeenschappelijke eigenschappen zijn de minimum- en maximumwaarde van de grafiek of meter. Soms kan het

nuttig zijn dat de schaal van grafiek zich automatisch aanpast aan de waarden. Zorg ervoor dat hier de nodige overerving toegepast wordt.

Hint: Gebruik een superklasse dat gemeenschappelijk is voor alle type grafieken. Een andere superklasse kan gebruikt worden voor alle meters.



Figuur 7: Grafiek met een enkele kleur (links) en met een gradiëntkleur (rechts).



Figuur 8: Enkele meters: lineaire meter (links), circulaire meter (midden) en thermometer (rechts).

9. OPSLAG

Naast het weergeven moet het programma ook in staat zijn om de gemeten data in bestanden te kunnen opslaan. Naast het opslaan moet het programma ook over een routine beschikken dat toelaat om dergelijke bestanden te openen en de metingen weer te geven. Een gemakkelijk formaat is “comma separated values” (CSV). Zorg ervoor dat elke kolom een kolomnaam heeft en dat alle waarden op de juiste plaats in het bestand komen. Uiteraard past er bij elke meting ook het tijdstip van meten!

Hint: Een bestand met metingen kan snel heel groot worden. Beperk elk bestand tot maximaal 24 uren. Nadien start je een nieuw bestand. Maak gebruik van het DateTime object om een deel van de bestandsnamen te bepalen. Een mogelijk bestandsnaamformaat kan zijn: “wheater_2016_05_12.csv”.

Hint: Voorzie ook de nodige beveiligingen opdat de interface niet vastloopt wanneer naar een bestand geschreven wordt. Hetzelfde geldt uiteraard bij het uitlezen van een bestand!

10. OPDRACHT

Schrijf de bovenvermelde interface en zorg ervoor dat deze aan alle vermelde eisen voldoet. De weergave van de gemeten waarden moet duidelijk zijn. Opslaan en lezen van gemeten waarden moet ook mogelijk zijn alsook het uitlezen van de bestanden van de SD-kaart. Eventueel kan de communicatie voorzien worden om de vermelde commando's vanuit het grafische interface naar het weerstation over te brengen. Voorzie de nodige klassen (en overervingsmechanismen), conversiemechanismen en de nodige beveiligingen zodat de interface niet stopt wanneer er problemen zich voordoen met de hardware.

De opdracht wordt op de dag van het examen object georiënteerd programmeren voorgesteld aan de assistenten elektronica-ICT en telt voor 20% mee in de eindscore van het vak Object-georiënteerd programmeren. Voor vragen m.b.t. de hardware kan je terecht bij Y. Verbelen. Voor andere vragen kan je terecht bij L. Segers.