

Spaar water op de paardenrenbaan

meet continu de bodemvochtigheid en temperatuurniveau via IoT

David Cuartielles (Arduino)

Voor paardenraces moest de renbaan altijd 'met de hand' in conditie worden gehouden, waarbij deskundige zorg van bekwaam onderhoudspersoneel nodig was. Maar het bleek onmogelijk om alles overal en gelijktijdig te controleren. Anders Åkerberg, Project Manager bij de Jägersro paardenrenbaan, nam contact op met Arduino om slimmere oplossingen te vinden.

De Jägersro paardenrenbaan, waarschijnlijk de belangrijkste in het zuiden van Scandinavië, diende te bezuinigen op het waterverbruik. Anders Åkerberg nam contact met ons op om te overleggen hoe dit op een slimme manier kon worden gedaan. "Uiteraard zijn we geïnteresseerd in het zo milieuvriendelijk mogelijk maken van onze renbaan. We hebben nu een expert die de arena controleert en beslist wanneer de sproeimachine van stal moet worden gehaald, om zo de bodemvochtigheid op peil te houden. Er komen veel variabelen om de hoek kijken bij zo'n beslissing, zoals de helling van de verschillende delen van de baan, de temperatuurverschillen als gevolg van de stand van de zon, en zelfs hoe de wind over de baan waait. Er zijn simpelweg te veel externe factoren om rekening mee te houden." Wat aanvankelijk een eenvoudig automatiseringsproces leek, liep daarom uit op een zeer interessant project, waarbij het gebruik van de Arduino Portenta H7 als een on-site datacollector-tool dat ook besluiten neemt, werd onderzocht. Dit artikel beschrijft de eerste stappen die we bij dit project hebben gezet, evenals de soorten sensoren die voor het project zijn gekozen, en waarom. Alle hier beschreven ervaringen kunnen worden toegepast op veel andere IoT-projecten, van renbanen met echt gras tot akkers die zorgvuldig moeten worden geïrrigeerd voor het telen van groenten.

Achtergrond

In 2017 werd besloten om een nieuwe drafportfaciliteit te bouwen in Malmö, Zweden (**figuur 1**), ter vervanging van de bestaande renbaan, die niet voldeed aan de verwachtingen van jockeys, paarden, toeschouwers en zelfs de omwonenden, van wat een moderne sportfaciliteit zou moeten zijn. Het plan omvat een herontwerp van het hele complex, dat in 2025 klaar moet zijn.

Draverij is een veel beoefende sport in Zweden en Ierland. Er zijn verschillende vormen; jockeys kunnen de paarden direct berijden, of ze kunnen in een 'sulky' zitten die door een paard wordt getrokken in wat bekend staat als *harness racing*. Ongeacht de sport, de arena is hetzelfde. Deze bestaat uit een 1000 meter lange renbaan van natuurlijke materialen zoals zand en gravel, en steentjes in verschillende maten. De kwaliteit van de baan wordt gecontroleerd door speciaal aangesteld personeel, dat de baanconditie voortdurend bewaakt en in stand houdt.

Een van de belangrijkste zaken daarbij is de vochtigheid van de drafbaan. Om de impact op de hoeven en benen van de paarden te minimaliseren mag deze niet te nat of te droog zijn. Het effect van goed bewaterd gravel wordt *damping* genoemd die, mede door het gewicht van de paarden, het vochtige gravel doet veranderen in een soort mengeling van veren en schokdempers. De kwaliteit van de drafbaan

is continu een bron van controverse tussen de gebruikers en het onderhoudspersoneel van de baan. Tot nu toe worden draagbare meetapparaten gebruikt die, samen met de observaties van de expert, het optimale vochtgehalte van de baan bepalen. Irrigatie gebeurt met behulp van een speciale tractor met watersproei-ers. Een dergelijke situatie biedt goede uitgangspunten voor mogelijke computerisering, om daarmee de baanexpert te helpen sneller en betere informatie te krijgen, en een uniforme en voorspelbare baankwaliteit te garanderen. Theoretisch moet dit ook leiden tot een vermindering van het waterverbruik van de drafbaan, door een nauwkeurigere bepaling van de sproeilocaties en de benodigde hoeveelheid water.

Het probleem

De echte uitdaging ligt in het nauwkeurig bepalen van de juiste hoeveelheid water die voor elk deel van de baan moet worden gebruikt. De aard van een drafbaan is hier debet aan, want:

- het binnenste spoor en de bochten hebben verschillende hellingen (2 graden versus 18);
- het is onmogelijk om de gravellagen op het oppervlak van de baan een exact identieke dikte en samenstelling te geven;
- en zelfs als dat zou kunnen, zouden de paarden de baan kapot maken zodra ze hun eerste race liepen;
- de oriëntatie van de baan zorgt voor niet-uniforme zoninval (met verschil in verdamping);
- de wind beïnvloedt de omstandigheden op onvoorspelbare wijze;
- periodiek, of in opdracht van de baanbeheerder, wordt het oppervlak 'geschaafd' en wordt een paar centimeter gravel verwijderd en vervangen.

De oplossing

Zoals je je kunt voorstellen zijn de variabelen, die van invloed zijn op de kwaliteit van de renbaan, veel te talrijk om de vraag te beantwoorden waar de baan moet worden bewaterd, en wanneer technieken zoals computer vision moeten worden gebruikt. De betrouwbaarste manier om erachter te komen waar de renbaan besproeid moet worden vereist de installatie van een permanent sensornetwerk onder de dunne gravel-toplaag van de baan. Na het ophalen van de data moet het mogelijk zijn om de drafbaan te modelleren en het vochtgehalte te bepalen. Dit project kijkt naar die eerste fase. Ons doel is om een 'connected' drafbaan te creëren waarbij in real time de vochtigheidsgraad op meerdere plekken tegelijk wordt bepaald.

Ons plan is het bouwen van een systeem dat zal worden gebruikt om de kwantitatieve gegevens van de renbaan (vochtigheid en temperatuur) te monitoren en te correleren met waarnemingen door de baanbe-

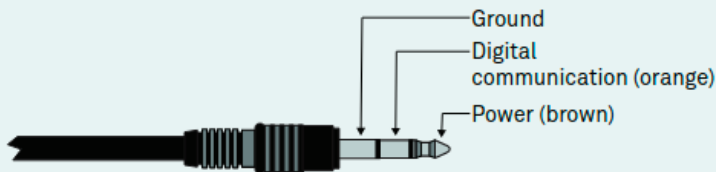


heerders. Zoals hierboven al aangegeven, kan worden verwacht dat vergelijkbare sensorinformatie, afkomstig van verschillende punten op de baan, zal resulteren in verschillende kwaliteitsindicaties. Het is dus belangrijk om erachter te komen hoe je een systeem kunt trainen dat in staat is om de baanbeheerder te voorzien van gegevens op een manier waar hij iets aan heeft, en niet domweg van de ruwe sensorwaarden. Idealiter zou het systeem in staat moeten zijn om de tijd te verkorten die voorheen nodig was om over de baan te lopen om gegevens te verzamelen. En dat beslissingen nu kunnen worden genomen door simpelweg naar een visuele weergave van de baan te kijken.

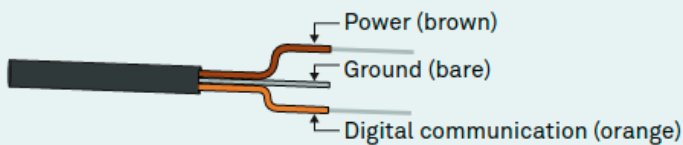
Voor Anders was dit de eerste stap naar de automatisering van het bewateringssysteem en de vermindering van het waterverbruik van de arena, terwijl de conditie van de drafbaan werd verbeterd door deze zo uniform mogelijk te houden. Voor mij is dit project de kans om een aantal complexe technologieën uit te proberen. En dan heb ik het over niet-triviale, dure sensoren (kosten tussen de € 150 en € 350 per stuk). Ten behoeve van dit artikel heb ik een sensorfamilie onderzocht die al een tijdje bestaat, maar die doorgaans voor de meesten van ons niet haalbaar is. Hieronder beschrijf ik de eerste stappen van dit project als voorbeeld hoe anderen een vergelijkbare aanpak zouden kunnen kiezen. Dezelfde technologie zou namelijk kunnen worden toegepast bij smart agriculture, het onderhoud van grasvelden voor sport, detectie van lekkages van vloeistoffen... in wezen bij elk soort project dat te maken heeft met water en bodem van welke aard dan ook.

De ultieme oplossing, eentje die de hele baan bestrijkt, zal een nog nader te bepalen aantal sensoren en Arduino Portenta H7-boards bevatten. Een deel van het onderzoek dat wordt gedaan betreft inzicht in hoeveel meter afstand er moet zijn tussen de sensorpunten. Het is ook onduidelijk of we op verschillende punten op de baan moeten meten of, omdat deze naar de binnenzijde afloopt, het voldoende kan zijn om een enkel meetpunt te gebruiken. Aan de andere kant zou het kunnen dat er sensoren nodig zijn met tot 70 meter lange kabels, wat de sensoren nog duurder maakt.

▲
Figuur 1. Het terrein voor de nieuwe Jägersrorenbaan (bron: Jägersro Hästcenter).



Figuur 2a. Jack-connector (bron: METER Group, TEROS 11 Integrator Guide).



Figuur 2b. Kale vertinde aansluitdraden (bron: METER Group, TEROS 11 Integrator Guide).

Tabel 1. Vergelijking van de belangrijkste sensor-eigenschappen.

Eigenschap	TEROS 10	TEROS 11	TEROS 21
Resolutie	0,001 m ³ /m ³	0,001 m ³ /m ³	
Temperatuur	-40 ... 60 °C	-40 ... 60 °C	-40 ... 60 °C
Type meting	vochtigheid	vochtigheid, temperatuur	waterpotentiaal, temperatuur
Analoge sensor	1 ... 2.5 V	1 ... 2.5 V	
Communicatie protocol	analoog uit	DDI of SDI-12 serieel	DDI of SDI-12 serieel
Volume influence	430 ml	1010 ml	
Frequentie	70 MHz	70 MHz	70 MHz
Voeding	3 ... 15 VDC	3 ... 15 VDC	3,6 ... 15 VDC

Want die moeten in de fabriek met de hand worden gekalibreerd, en ze zijn meestal niet ontworpen voor een dergelijke lange bekabeling.

Waterpotentiaal

Eén relevant concept dat ik leerde kennen tijdens het vooronderzoek voor dit project is de 'waterpotentiaal'. Om beter te begrijpen wat dit betekent moet je denken in termen van thermodynamica. Overtollige warmte stroomt altijd op zo'n manier dat twee rakende oppervlakken een gelijke temperatuur krijgen. Waterpotentiaal kan op dezelfde manier worden gezien: water zal van gebieden met een hogere waterpotentiaal naar gebieden met een lagere waterpotentiaal stromen. Vanuit dat perspectief kunnen we de waterpotentiaal niet als een absolute grootheid meten, het moet worden geïnterpreteerd als een relatieve grootheid. Zuiver water levert de referentiewaarde voor deze potentiaal. Het voorkomen van verschillende oplosmiddelen in het water, of wanneer het water door de hele bodem verdeeld is, leidt tot een verandering in de waterpotentiaal. Wanneer je dus op het punt staat om in de datasheets van een high-end bodemvoch-

tigheidssensor te duiken, is het belangrijk om dit in gedachten te houden.

De sensoren

Voor dit artikel heb ik enkele sensoren uit de TEROS-familie van de METER Group vergeleken. Deze sensoren bestaan al een tijdje (het bedrijf heette voorheen Decagon). Het zijn digitale sensoren: ze hebben een microcontroller op de sensor die metingen uitvoert en de informatie terugstuurt naar een ander apparaat, in analoge of digitale vorm. De meeste bedrijven die dergelijke sensoren verkopen, leveren ook handeld dataloggers voor gebruik in het veld. Daarom is het mogelijk om deze sensoren met een gebruiksvriendelijke jack-connector te krijgen, of met gestripte en netjes vertinde uiteinden, zoals te zien in **figuur 2a** en **2b**.

Ik denk dat de meeste lezers bekend zijn met goedkope bodemvochtigheidssensoren, die de bodemgeleiding of bodemcapaciteit meten. Ze zijn er in de vorm van een vork met één tot drie tanden, en leveren een signaal dat eenvoudig kan worden ingelezen via een analoge poort op een microcontroller-board. Deze goedkopere sensoren zijn geweldig voor gebruik binnenshuis, en voor projecten waar ze gemakkelijk kunnen worden vervangen indien ze niet werken. Maar ze ingraven en dan hopen dat ze winterse onder-nul-temperaturen overleven, of de niet aflatende mechanische belasting als gevolg van bewegend gravel, getuigt niet van veel realiteitszin. Aan de meer professionele kant van het spectrum vinden we de sensoren die ik voor evaluatiedoeleinden heb gekozen. Er zijn twee verschillende families van sensoren, namelijk die met metalen uiteinden en die welke piëzo-elektrische technologie gebruiken.

Tabel 1 toont enkele basisspecificaties van sensoren, gewoon om je een idee te geven wat ze kunnen. Als je meer informatie wilt, kun je de datasheets via de weblinks downloaden.

De sensoren TEROS 10 (**figuur 3**) en TEROS 11 (**figuur 4**) gebruiken dezelfde technologie voor de vochtigheidsberekening, en zijn dus identiek als het gaat om wat ze kunnen. Met uitzondering van het oppervlak dat ze bestrijken (bekend als *volume influence*), die meer dan het dubbele is voor de TEROS 11. Zoals je op de foto's van de sensoren kunt zien, lijken deze eerste twee apparaten identiek, behalve dat de ene twee pinnen heeft en de andere drie. De derde pin van de TEROS 11 is een thermokoppel voor het meten van de bodemtemperatuur. Dit is een groot verschil tussen de twee sensoren, en het heeft ook invloed op de manier waarop gegevens worden teruggestuurd naar de microcontroller. De TEROS 10 stuurt een ruw analoge signaal dat rechtstreeks door een Arduino-board kan worden gelezen



met behulp van de functie `analogRead()`. Ik heb dit getest door simpelweg voeding, massa en signaal van de sensor aan te sluiten op de pinnen van mijn Portenta breakoutboard. Voeding van de sensor met 3,3 V was genoeg om hem te laten werken, ondanks de 5 m lange kabel tussen de sensor en mijn Portenta. Aan de andere kant kunnen de TEROS 11 en TEROS 21 (**figuur 5**) meer dan één soort meetwaarde leveren (vochtigheid en temperatuur, om te beginnen), dus is het noodzakelijk om een digitaal protocol te implementeren voor communicatie met de sensoren. Dit is ook een voordeel want het is dan mogelijk is om de sensoren te configureren om te communiceren met behulp van SDI-12, een specifieke bustechnologie (hierover later meer). Dit is ook de reden waarom de TEROS 11 en TEROS 21 hogere spanningen nodig hebben. In mijn tests moest ik een externe 12 V voeding gebruiken om ervoor te zorgen dat de sensoren werkten zoals verwacht (**figuur 6**).

DDI versus SDI-12

Bij gebruik van SDI-12 (Serial Digital Interface bij 1200 Baud) is het mogelijk om meerdere sensoren met 1200 bps parallel aan te sluiten, en informatie daaruit terug te sturen met behulp van speciale, leesbare AT-commando's (zie de site van SDI-12 voor meer informatie [5]). Ook hebben deze sensoren een fallback-techniek geïmplementeerd, genaamd DDI-Serial, die de sensor dwingt om een tekstreeks naar de communicatiepoort te sturen bij het herstarten. **Figuur 7** toont een screenshot van de Serial Monitor-terminal van Arduino, die informatie ontvangt tijdens meerdere voedingscycli van de sensor. In dit geval had ik net wat ijsblokjes in een bak met water gedaan, en je kunt zien dat je dan kunt meten hoe de vochtigheidswaarde consistent blijft terwijl de temperatuur daalt. De laatste tekens aan het einde van elke regel zijn de sensorspecifieke identifier, checksum enzovoort. Daarmee is één mogelijke manier om dit project te realiseren met een TEROS 11 of TEROS 21 al opgelost door gebruik te maken van de DDI-gegevensstromen na elke sensor-voedingscyclus. Ik hoef alleen maar een board te configureren waar de verschillende ontvangende pinnen aan elkaar kunnen worden gekoppeld, en met verschillende MOSFET's de sensoren afzonderlijk van stroom worden voorzien. Ik zou ook de impedantie kunnen meten en berekenen hoeveel sensoren ik tegelijkertijd kan aansluiten, en of ik dan een soort driver nodig zou hebben. Zowel DDI als SDI-12 zijn ééndraads 7N1 seriële communicatieprotocollen. Er is een goed gedocumenteerde en onderhouden uitgebreide seriële bibliotheek te vinden voor AVR-gebaseerde Arduino-boards [6]. Maar deze maakt gebruik van Arduino's SoftwareSerial library plus hardware interrupts om signalen goed te ontvangen.



Figuur 3. METER TEROS 10 (bron: Kristoffer Engdahl / Arduino).



Figuur 4. METER TEROS 11 (bron: Kristoffer Engdahl / Arduino).



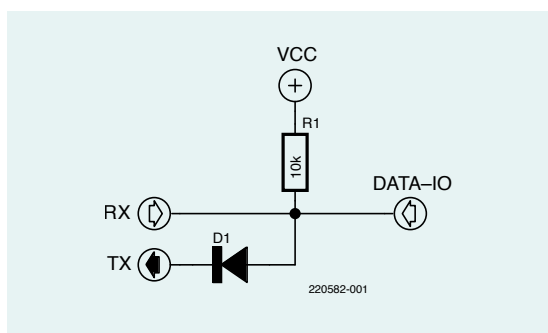
Figuur 5. METER TEROS 21 (bron: Kristoffer Engdahl / Arduino).



Figuur 6. Testopstelling (bron: Kristoffer Engdahl / Arduino).



▲
Figuur 7. Arduino's
Serial Monitor met
DDI-berichten van de
TEROS 11.



Figuur 8. Tweedraads-naar-eendraads seriële converter.

De library vereist wat minimale hardware om verbinding te maken met de sensor. En omdat SoftwareSerial half-duplex is, kun je de datakabel van de sensor rechtstreeks op een pin aansluiten en werkt deze (zolang je een 3V3-board hebt, anders heb je spanningsconversie nodig om de sensor te beschermen). Portenta heeft geen probleem met hardware seriële poorten, dus we zullen geen SoftwareSerial-implementatie gebruiken (en ook omdat we compileren met mbedOS, dat niet goed samenwerkt met interrupt-gestuurde firmware). In mijn geval koos ik voor de truc van het gebruiken van een hardware seriële poort (pinnen 13/14 op de Portenta H7), met TX aangesloten via een diode naar een omhooggetrokken RX-pin. Dit is een bekende truc die ook in dit geval zou moeten werken (waarschuwing: dat doet het inderdaad, maar alleen voor ontvangen, niet voor verzenden). **Figuur 8** toont het schema voor de tweedraads-naar-eendraads conversie.

Nu ik DDI-communicatie heb, is mijn volgende stap het implementeren van een hardware-driver om het verzenden van datarequests naar de sensoren mogelijk te maken, om zo de SDI-12 te ondersteunen. Dit geeft me een alternatieve manier om informatie binnen te halen.

Voedingscycli voor de sensoren

Iets wat in alle datasheets voor deze sensoren wordt uitgelegd is de vereiste om ze kort in te schakelen, om zo alleen de gegevens uit te lezen, en ze vervolgens weer uit te schakelen. De redenen hiervoor zijn zowel het verminderen van het stroomverbruik als het verlengen van de levensduur van de betreffende elektronica. De metingen kunnen zelfs in gevaar komen als de sensoren gedurende lange tijd 'heet' blijven. Dit geldt voor alle drie de geteste sensoren. Dit concept van voedingscycli ('power cycling') betekent dat ik MOSFET's zal moeten opnemen om elk van de sensoren te activeren en te deactiveren als een methode om informatie via DDI te binnen te halen, iets wat ik toch al van plan was. Omdat ik de sensoren voed met 12 V, heb ik relatief zware MOSFET's nodig (zoals de IRFZ-24, een van mijn favorieten), waarmee ik ook het board kan hergebruiken indien ik nog een ontwerp zou maken voor andere situaties.

De juiste sensor kiezen

Voor mij is het duidelijk dat ik niet alleen de bodemvochtigheid, maar ook temperatuurgegevens moet vastleggen, dus de TEROS 10 is geen optie. Na enkele eerste tests in het lab gaf de TEROS 11 me een betere indruk qua mechanische robuustheid en waterdichtheid. De sensoren worden begraven in gravel, en ik verwacht dat ze in direct contact staan met water, dus de TEROS 11 is in dat opzicht ook beter. Ik kan echter geen beslissing nemen zonder enige vorm van veldonderzoek te doen. Ik ben van plan om een paar stuks TEROS 11 en TEROS 21 een paar maanden in de renbaan te begraven om enig inzicht te krijgen in hun degelijkheid. Als de prijs een beperkende factor wordt (dat zou later in het proces kunnen zijn), moet je weten dat de TEROS 21 meer dan € 350 per stuk kost, meer dan genoeg reden om hem niet te kiezen. Je kunt de TEROS 11 al voor iets meer dan de helft van die prijs kopen. Indien ik elke 10 meter een groep van drie sensoren in de baan zou begraven, dan heb ik 300 sensoren nodig. Je mag zelf uitrekenen hoeveel alleen al de sensoren dan gaan kosten.

Volgende stappen

Dit project is nog lang niet voorbij. Ik werk momenteel aan een constructie voor het Portenta breakout-board dat de voedingsaansluitingen voor de sensoren, de power-cycling MOSFET's en de drivers voor de eendraads seriële communicatie zal herbergen. Zodra ik dat klaar heb, voeg ik een robuuste 12V-voeding en een standaard ABS-behuizing toe, installeer alles ter plaatse, en sla ik de gegevens op een SD-kaart op en stuur ik deze via WiFi naar de Arduino Cloud (er is een WiFi-verbinding op de baan). Over een paar maanden weet ik dan welke sensoren het beste zijn. Als je dat ook wilt weten, volg dan de voortgang op mijn Arduino-blog [7], waar ik de verschillende stappen zal beschrijven, inclusief wat ik zoal nog zal tegenkomen. ◀

220582-03 (vertaling: Marc Gauw)

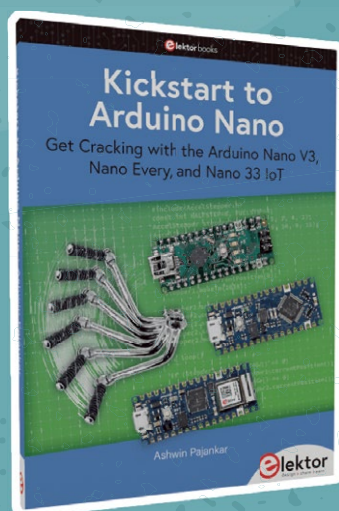


Gerelateerde producten

- > **Arduino Portenta H7 Development Board (SKU 19351)**
www.elektor.nl/19351

WEBLINKS

- [1] Jägersro Hästcenter: <https://jagersrohaestcenter.se/>
- [2] METER TEROS 10 handleiding: http://publications.metergroup.com/Manuals/20788_TEROS10_Manual_Web.pdf
- [3] METER TEROS 11/12 handleiding: https://publications.metergroup.com/Manuals/20587_TEROS11-12_Manual_Web.pdf
- [4] METER TEROS 21 GEN 2 handleiding: https://publications.metergroup.com/Manuals/20854_TEROS21_Gen2_Manual_Web.pdf
- [5] SDI-12 Support Group: <https://sdi-12.org>
- [6] Arduino-SDI-12 library op GitHub: <https://github.com/EnviroDIY/Arduino-SDI-12>
- [7] Arduino Blog: <https://blog.arduino.cc/>



Kickstart to Arduino Nano

Dit boek dient als eerste stap voor beginners en liefhebbers van microcontrollers die een begin willen maken met het programmeren met Arduino. Het boek volgt een stap-voor-stap aanpak om concepten en de werking van dingen uit te leggen. Elk concept wordt stevast gevolgd door een to-the-point schema en codevoorbeelden. Daarna volgt gedetailleerde uitleg over de syntaxis en de gebruikte logica. Door de concepten nauwgezet te volgen, raakt u vertrouwd met het bouwen van schakelingen, het programmeren met Arduino, de werking van de codevoorbeelden en de gepresenteerde schakelschema's.

www.elektor.nl/20241

