

Analoge signalen en microcontrollers

ADC's, DAC's, stroommeting en meer

Mathias Claussen (Elektor)

Terwijl een MCU intern met enen en nullen werkt, is de wereld eromheen analoog. Daarom is het soms nodig om analoge waarden in te lezen of uit te voeren. Met slechts een paar componenten kan een MCU worden aangesloten op de analoge wereld.

Of je microcontroller (MCU) nu een Arduino UNO, STM32 of wat dan ook is – vaak is het nodig om meer dan alleen digitale signalen te verwerken. De wereld om ons heen is analoog: niet alleen licht of donker maar ook elke tint daartussen. Daarom wil dit artikel enig inzicht geven in hoe analoge waarden kunnen worden ingelezen door een microcontroller en hoe ze ook kunnen worden uitgevoerd. Voor ervaren ontwikkelaars valt deze tekst zeker in de categorie 'lichte lectuur', maar we moesten allemaal ooit ergens beginnen. En hoewel het instellen en lezen van digitale pinnen heel eenvoudig is, wordt het een beetje lastiger als het gaat om het inlezen en uitvoeren van analoge waarden.

De pricipiële vragen zijn: hoe krijgen we analoge waarden in een digitaal werkende MCU, en hoe moeten ze daartoe worden voorbereid? En hoe kan een MCU analoge waarden uitvoeren? Om deze vragen te beantwoorden kijken we eerst naar twee belangrijke basiselementen, de analoog/digitaal-converter (ADC) en de digitaal/analoog-converter (DAC).

Analoog/digitaal-converter (ADC)

Een analoog/digitaal-converter (ADC) kan een analoog signaal omzetten in een digitale representatie. Deze kan dan later door de microcontroller worden gebruikt om de gegevens te verwerken. Twee veel voorkomende parameters van een ADC zijn de resolutie in bits en de bemonsteringssnelheid in monsters (samples) per seconde. Bij microcontrollers ligt de resolutie meestal tussen 10 en 12 bit. Dit betekent dat de ADC zijn analogeingangsspannings-

bereik onderverdeelt in 1024 waarden (10 bit) of 4096 waarden (12 bit) en vervolgens de aangelegde spanning toewijst aan een van deze waarden. Het waardenbereik is nu eindig en het analoge signaal moet worden toegewezen aan de best passende digitale waarde, ook al komt die niet precies overeen met het rekenkundig bepaalde niveau. Deze afwijking wordt kwantisatiefout genoemd. De tweede grootte, de bemonsteringsfrequentie in samples per seconde, geeft aan hoeveel keer per seconde de ADC een analoge waarde kan omzetten in een digitale waarde. Tenzij het signaal een gelijkspanning is, bepaalt de bemonsteringsfrequentie ook de maximale bandbreedte van een signaal dat kan worden verwerkt. Het omgekeerde van de bemonsteringsfrequentie geeft aan hoe lang het duurt voordat de ADC een analoog/digitaal-conversie heeft voltooid – met andere woorden, hoe lang de software moet wachten nadat een conversie is gestart voordat de resulterende waarde kan worden verwerkt.

Spanningsbereik, resolutie en referentiespanning

Als we uitgaan van een ingangsspanningsbereik van 0 V tot 5 V met een resolutie van 10 bit, krijgen we:

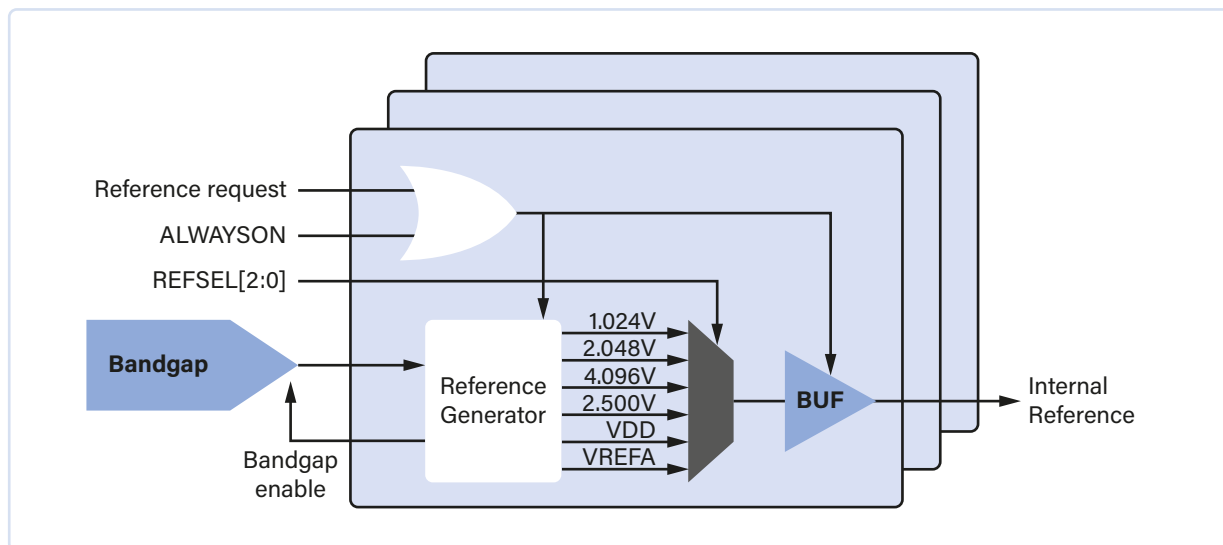
$$5 \text{ V} / (2^{10} \text{ bit} - 1) = 4,89 \text{ mV/bit}$$

Een spanning van bijvoorbeeld 452 mV krijgt zodoende intern de waarde 92:

$$452 \text{ mV} / 4,89 \text{ mV/bit} = 92$$

en zou dus verder verwerkt worden als 449,88 mV. De kwantisatiefout bedraagt dus 2,12 mV. De afwijking kan niet groter zijn dan 0,5 LSB (minst significante bit), wat in dit voorbeeld net geen 2,5 mV is.

Als het te meten spanningsbereik aanzienlijk kleiner is dan het toegestane ingangsspanningsbereik van de ADC, kan dit bereik bij de meeste ADC's worden aangepast. De ADC vergelijkt de analoge waarden altijd met een referentiespanning. Veel MCU's bieden tegenwoordig een interne referentiespanning, of zelfs meerdere. De AVR128DB van Microchip dient als voorbeeld. **Figuur 1** toont het blokschema van de spanningsreferentie van deze controller. Zes referentiespanningen van 1,024 V, 2,048 V, 4,096 V, 2,5 V, VDD en



Figuur 1. Spanningsreferentie AVR128DB (bron: Microchip, AVR DB Family datasheet [PDF] – <https://elektor.link/AVR128DBDatasheet>).

V_{REFA} zijn beschikbaar. De eerste drie spanningen kunnen worden voorgesteld als 2^N en maken het mogelijk de digitale waarden van de ADC later heel gemakkelijk terug te rekenen naar volt. Iets dat op bijna alle MCU's te vinden is, is GND (massa) versus AGND (massa voor het analoge deel van de MCU). Omdat er in het digitale domein hoogfrequente ruis kan optreden die ook door het massapad loopt, heeft alles wat een ADC of DAC is zijn eigen massa. In het ontwerp kunnen GND en AGND dus van elkaar worden ontkoppeld om storingen te voorkomen. Indien verwacht wordt dat de gemeten grootheid niet groter zal zijn dan 1 V, kan 1,024 V als referentie worden gekozen. Dit zou dan resulteren in een resolutie van 1,00098 mV/bit. 452 mV zou dan overeenkomen met de digitale waarde 452; bij de verdere verwerking zou de kwantisatiefout dus aanzienlijk kleiner zijn dan in het eerste voorbeeld. De juiste keuze van de referentiespanning is echter niet het enige dat ervoor kan zorgen dat de resultaten van de conversies dicht bij de fysieke (analoge) waarde liggen. Ook de kwaliteit van de referentiespanning is van invloed.

In de AVR128DB kunnen VDD en V_{REFA} ook als referentiespanningen worden beschouwd. VDD is de voedingsspanning van de chip, en V_{REFA} is een externe pin van de AVR128DB waarop je je eigen externe referentiespanningsbron kunt aansluiten. Het gebruik van VDD als referentie kan een ongelukkige keuze zijn, omdat deze spanning belast is met ruis van het digitale deel van de MCU, en het is moeilijk deze ruis 'weg te filteren'. De situatie is anders met V_{REFA} : hier kan een externe hoge-precisie spanningsbron worden gebruikt als referentie. Natuurlijk is deze nog steeds verbonden met de 'vervulde' voedingsspanning, maar als externe component kan deze veel beter worden gefilterd. Een stabiele en storingsvrije referentie is de basis van een goede analoog/digitaal-conversie. Er is nog een andere manier om de resolutie van een ADC te verhogen zonder de referentiespanning te veranderen; dit wordt oversampling genoemd. Oversampling gebruikt het minst significante bit, dat heen en weer geschakeld wordt tussen nul en één als de signaalwaarde niet precies overeenkomt met de beschikbare resolutie

van de ADC. Door het gemiddelde van dit minst significante bit te nemen wordt een fractie van een één verkregen, wat een hogere resolutie betekent. Deze methode verlaagt echter de effectieve bemonsteringssnelheid. Voor elk bit dat door oversampling wordt toegevoegd, wordt de bemonsteringssnelheid met een factor 4 gereduceerd. Als bijvoorbeeld een 12bit-ADC maximaal 130 ksp/s (kiloamples per seconde) aankan, zouden vier oversampling-bits, dus 16 bit als totale resolutie, resulteren in slechts 507 samples per seconde:

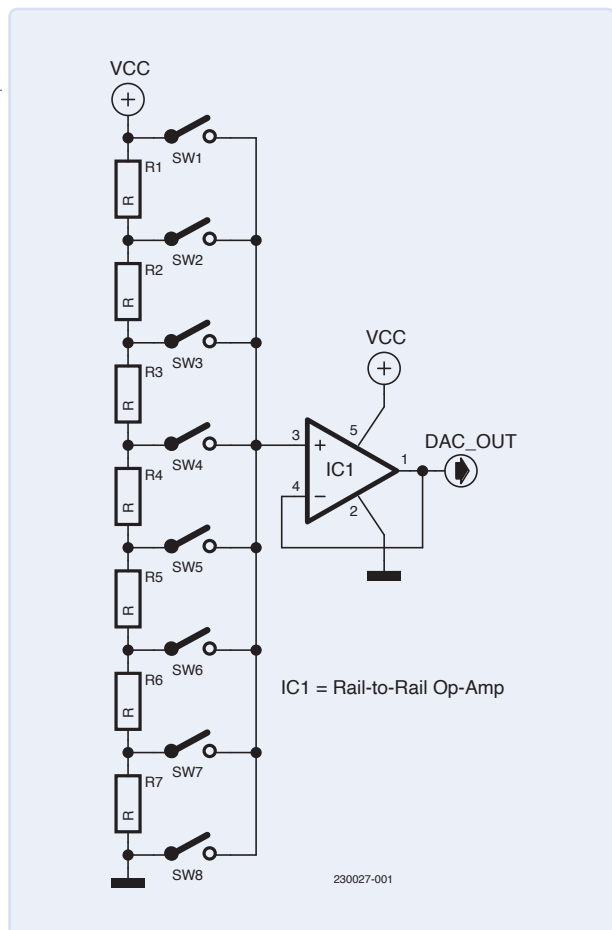
$$130 \text{ ksp/s} / 4^4$$

Bovendien moet de processor, afhankelijk van de ADC, het gemiddelde berekenen (nodig voor oversampling) wat extra rekenwerk oplevert. Sommige nieuwere ADC's, zoals die van sommige STM32-controllerfamilies, voeren deze taak volledig autonoom uit zonder de CPU extra te belasten.

ADC's zijn er in verschillende uitvoeringen. De meest voorkomende ADC's in controllers zijn van het SAR-type (successive approximation). Voor een hoge resolutie (24 bit en meer) bij relatief lage samplesnelheden (van ksp/s tot enkele samples per seconde) worden meestal delta/sigma-ADC's gebruikt. De details van de delta/sigma-conversie en de implementatie ervan in een CPLD/FPGA (Complex Programmable Logic Device/Field-Programmable Gate Array) zijn te vinden in een artikel van Elektor [1]. Er zijn andere ADC-types, waarvan het Flash ADC-type de snelste conversietijd belooft, en daarom vaak gebruikt wordt in modern flash-geheugen.

Digitaal/analoog-converter (DAC)

Met een DAC genereer je een analoge waarde uit een digitale waarde. In sommige MCU's is een DAC ingebouwd, hoewel niet zo vaak als een ADC. Bij de AVR128DB is dat een 10bit-DAC, die dus 1024 verschillende spanningen kan produceren. Als er geen DAC in de MCU zit, kan een externe DAC-chip zoals de Microchip MCP4922 worden gebruikt. In beide gevallen betreft het een DAC met een



Figuur 2. Structuur van een DAC met weerstandsketen.

weerstandsketen, waarbij de MCP4922 is een tweekanaals 12bit-DAC is. Terwijl een weerstandsketen-DAC goed kan worden geïntegreerd in een chip, is zoiets nauwelijks haalbaar met discrete weerstanden. Een blik op de 3-bit weerstandsketen-DAC van **figuur 2** maakt aannemelijk waarom dat zo is: voor de DAC zijn

$$2^N - 1 \quad (N = \text{aantal bits})$$

weerstanden en N schakelaars nodig. Voor een 16bit-DAC betekent dat dus 65.535 weerstanden en 65.536 schakelaars!

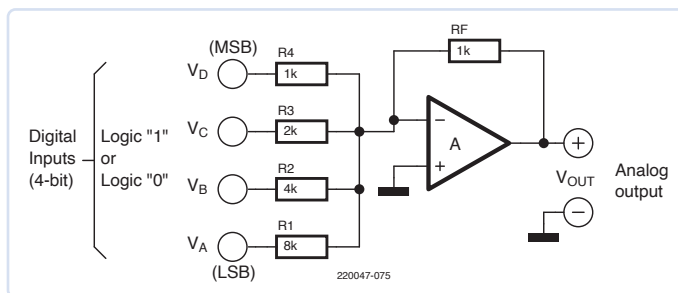
Net als een ADC heeft ook een DAC een referentiespanning nodig die het bereik aangeeft waarbinnen de uitgangsspanning kan liggen. De DAC zelf kan echter slechts een paar mA aansturen, zelfs als de uitgang wordt gebufferd door een opamp.

Als er echter geen interne DAC is en er nog voldoende pinnen van de MCU beschikbaar zijn, bestaan er andere manieren om een externe DAC te bouwen. Eén optie is een binair gewogen DAC, zoals in **figuur 3**. Hier zijn slechts N+1 weerstanden en een opamp nodig voor N bits. De waarden van de weerstanden worden gegeven door

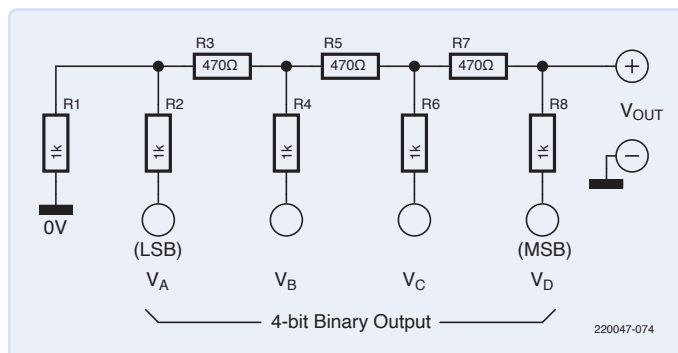
$$R_N = R_{(N-1)} \cdot 2$$

en worden dus telkens verdubbeld. Een dergelijke DAC kan worden gebruikt om bijvoorbeeld analoge VGA-signalen te genereren, zoals in de datasheet bij [2].

Een andere veelgebruikte variant is een R-2R DAC. In tegenstelling tot een binair gewogen DAC zijn hier 2N weerstanden en een opamp nodig voor N bits, maar er zijn slechts twee weerstandswaarden, terwijl we in een binair gewogen DAC met N meestal lastig



Figuur 3. Schakeling van een binair gewogen DAC.



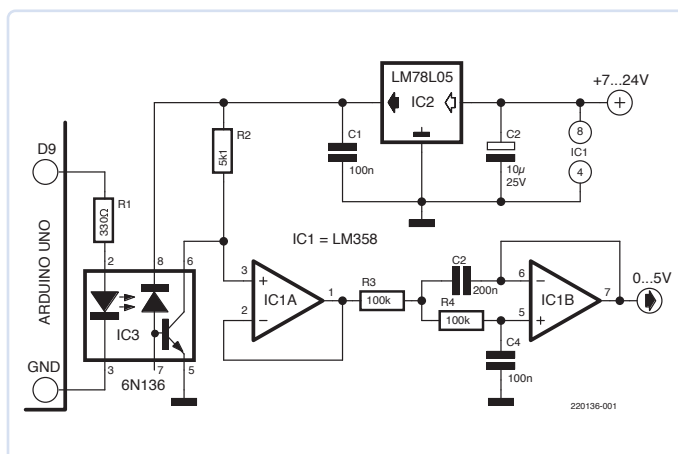
Figuur 4. Principe van een R-2R DAC.

te vinden fractionele waarden te maken hebben. Het schema van zo'n DAC is geschetst in **figuur 4**.

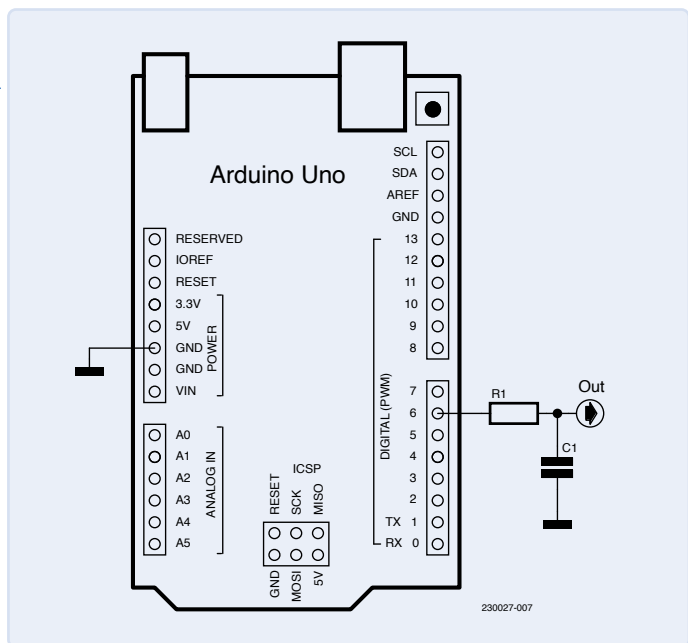
Als I/O-pinnen echter schaars zijn op de gebruikte MCU, kan de PWM-eenheid met wat externe onderdelen ook als DAC dienen. De `analogWrite()` functie voor de Arduino UNO produceert op deze manier analoge waarden met behulp van de digitale pinnen.

Figuur 5 toont een voorbeeld uit het Elektor-artikel in [3] voor een Arduino UNO.

Een eenvoudiger opzet is de schakeling van **figuur 6**, die alleen bestaat uit een RC-laagdoorlaatfilter om het signaal af te vlakken. Een dergelijke DAC heeft echter nadelen. Aangezien het PWM-sig-naal moet worden afgevlakt, moeten de waarden voor R en C



Figuur 5. Geïsoleerde analoge uitgang voor de Arduino UNO.



Figuur 6. Arduino UNO met RC-laagdoorlaatfilter.

worden afgestemd op de betreffende toepassing. Ook zal er altijd een klein PWM-residu in het analoge signaal achterblijven, dat als ongewenste ruis merkbaar wordt.

Signaalvoorbereiding

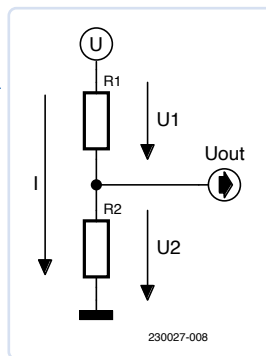
De te meten of uit te voeren signalen zijn niet altijd direct compatibel met de gekozen MCU. In de digitale wereld zijn veel ontwikkelaars bekend met het probleem van het aansluiten van 5V-componenten op 3,3V-systemen. Soortgelijke problemen doen zich ook voor in de analoge wereld en diens interfaces. Audio, video en 4-20 mA current loops zijn slechts enkele voorbeelden die signaalvoorbereiding (signaalconditionering) vereisen. De eenvoudigste vorm van signaalconditionering is waarschijnlijk een spanningsdeler, die niet alleen kan worden gebruikt voor analoge waarden, maar ook voor het aanpassen van digitale signalen.

Spanningsdelers en ADC's

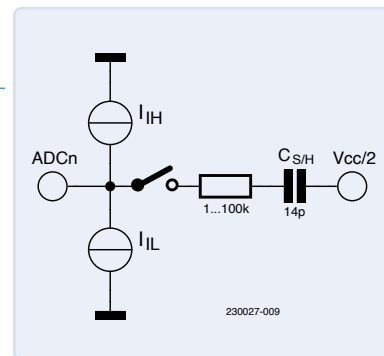
Figuur 7 toont een onbelaste spanningsdeler – dat is de geïdealiseerde versie. Wanneer een spanning (U) wordt aangelegd, gaat er een stroom (I) door $R_1 + R_2$ lopen. Over weerstand R_1 valt nu een spanning van $R_1 \times I = U_1$ en over R_2 een spanning $R_2 \times I = U_2$, waarbij $U = U_1 + U_2$. Als je nu bijvoorbeeld een bereik van 0 tot 10 V wilt meten met een ADC die alleen 0 tot 5 V kan verwerken, moet de spanning gehalveerd worden. Het is duidelijk dat dan $R_1 = R_2$ moet zijn.

De stroom (I) moet echter ook in aanmerking worden genomen bij de keuze van de weerstandswaarden. Hoewel $10\ \Omega$ voor R_1 en R_2 rekenkundig gezien zou resulteren in de gewenste spanningen voor U_1 en U_2 , zou er een stroom van 500 mA door de weerstanden lopen. Dit zou niet alleen de spanningsbron kunnen overbelasten, maar ook de weerstanden zelf. Een vermogensdissipatie van $P = R \times I^2 = 2,5\ \text{W}$, die elke weerstand in de vorm van warmte moet dissiperen, is namelijk veel te hoog voor de meeste kleinere weerstanden.

Een andere optie om de vermogensdissipatie te beperken, zijn weerstanden van $1\ \text{M}\Omega$. Dan zou er een stroom van slechts $5\ \mu\text{A}$ lopen, en de vermogensdissipatie zou $25\ \mu\text{W}$ bedragen, maar dit andere uiterste veroorzaakt een nieuw probleem. Als in de ADC



Figuur 7. Onbelaste spanningsdeler.

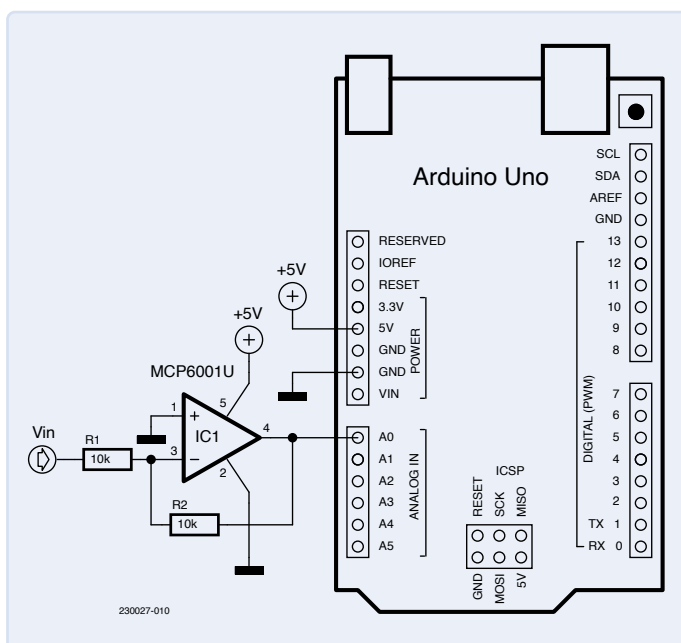


Figuur 8. ATmega328P ADC interne ingangsstructuur (bron: Microchip, Atmel Atmega328P datasheet [PDF] – <https://elektor.link/ATmega328PDatasheet>).

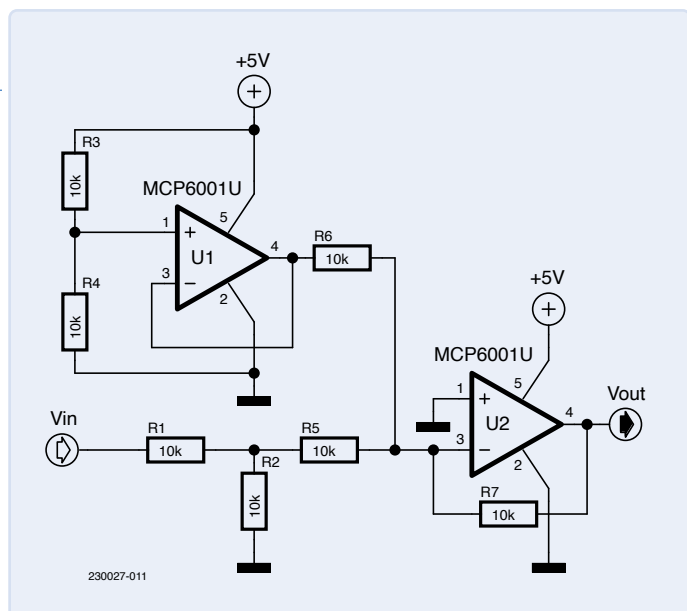
een sample&hold-trap is ingebouwd, zoals in de ATmega328 in **figuur 8**, kan dit leiden tot foutieve meetwaarden. De ADC heeft een kleine condensator aan de ingang die opgeladen wordt. Tijdens een conversie wordt de condensator voor een korte tijd (voor de duur van de ingestelde bemonsteringstijd) verbonden met de analoge ingang en opgeladen. Als de stroom te gering is, bestaat de mogelijkheid dat de condensator tijdens de bemonsteringstijd niet volledig wordt geladen. Als de ADC later de waarde van de onvolledig geladen condensator omzet, zal het resultaat onjuist zijn. Een blik op de datasheet van de MCU van je keuze helpt om de juiste waarden te bepalen. Voor de AVR in dit geval zou een waarde van $4,7\ \text{k}\Omega$ voor R_1 en voor R_2 en dus een stroom van $1,06\ \text{mA}$ de juiste keuze zijn.

Negatieve spanningen

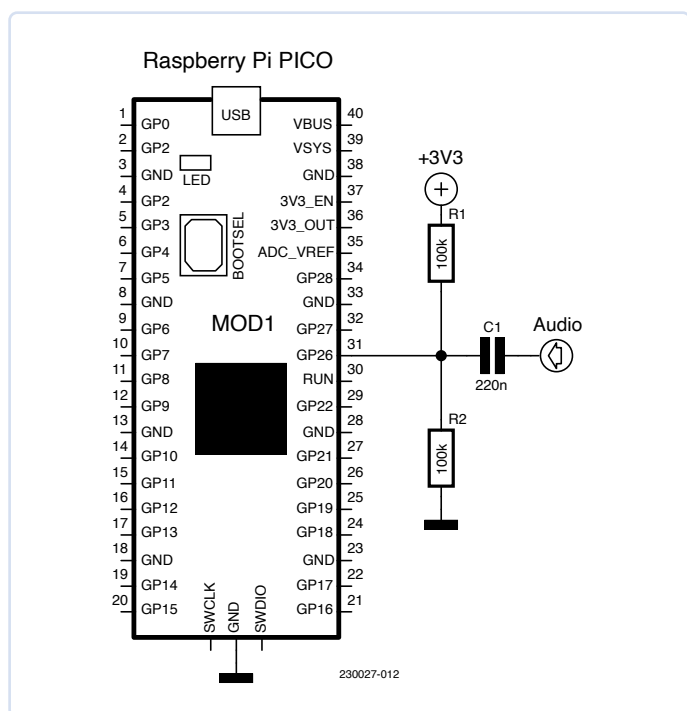
Positieve spanningen zijn vrij gemakkelijk te meten, bijvoorbeeld met een spanningsdeler. Maar wat gebeurt er als de te meten spanning negatief is? Door een opamp met een paar weerstanden te gebruiken als inverterende versterker, zoals in **figuur 9**, kan de spanning worden geïnverteerd.



Figuur 9. Inverterende versterker.



Figuur 10. DC-offset met operationele versterkers.



Figuur 11. Audio-ingang voor de ADC van een Microchip AVR.

Zolang de spanning aan de ingang van deze schakeling onder 0 V blijft, zal de opamp correct werken. Als echter een spanningsbereik van -5 V tot $+5\text{ V}$ moet worden bestreken, is iets meer moeite nodig. Ervan uitgaande dat het toegestane ingangsbereik van de controller 0 tot 5 V bedraagt, moet het signaal met een zwaai van 10 V eerst worden gehalveerd door een spanningsdeler $R1/R2$. We zouden dus te maken krijgen met waarden van $-2,5\text{ V}$ tot $+2,5\text{ V}$. Dit bereik wordt nu met $2,5\text{ V}$ verhoogd door de spanningsdeler $R3/R4$. De tweede, niet-inverterende, opamp $U1$, buffert deze spanning. Het volledige schema is gegeven in **figuur 10**.

Bij de latere evaluatie van de gemeten waarden, moet je bedenken dat het nulpunt van de ingangsspanning nu met de helft van de voedingsspanning is verhoogd, in dit voorbeeld dus met $2,5\text{ V}$.

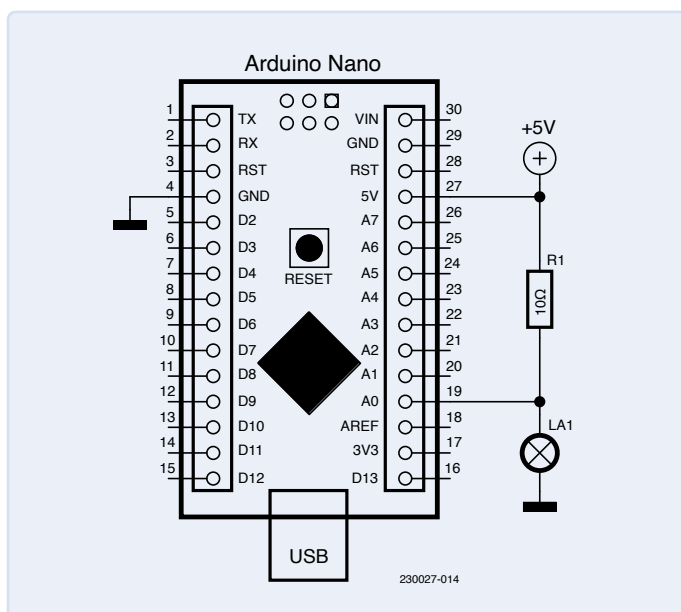
Audiosignalen en ADC's

Een interessante toepassing voor een MCU is de verwerking van audiosignalen. Gangbare audiosignalen hebben een top/top-spanning van $0,894\text{ V}_{PP}$ (-10 dBV) tot $3,472\text{ V}_{PP}$ ($+4\text{ dBV}$), afhankelijk van de signaalbron. De meeste thuisapparaten werken op -10 dBV , dat is $0,894\text{ V}_{PP}$. Het signaal heeft dus een minimumspanning van $-0,477\text{ V}$ en een maximum van $+0,477\text{ V}$. De ADC van een MCU zoals de Raspberry Pi RP2040 kan dit niet aan. Maar waarom zou je überhaupt audio willen verwerken met een 10bit-ADC? De AVR en snellere MCU's zijn bijvoorbeeld heel goed in staat om het frequentiespectrum van een audiosignaal te berekenen en weer te geven. Voor deze toepassing kunnen we profiteren van het feit dat het DC-gedeelte niet interessant is; alleen het AC-gedeelte van het audiosignaal is nodig ($C1$) en voorzien van een DC-offset ter waarde van de helft van de voedingsspanning voor de ADC ($R1/R2$) om het audiosignaal naar het positieve bereik te tillen. Een schema hiervoor staat in **figuur 11**.

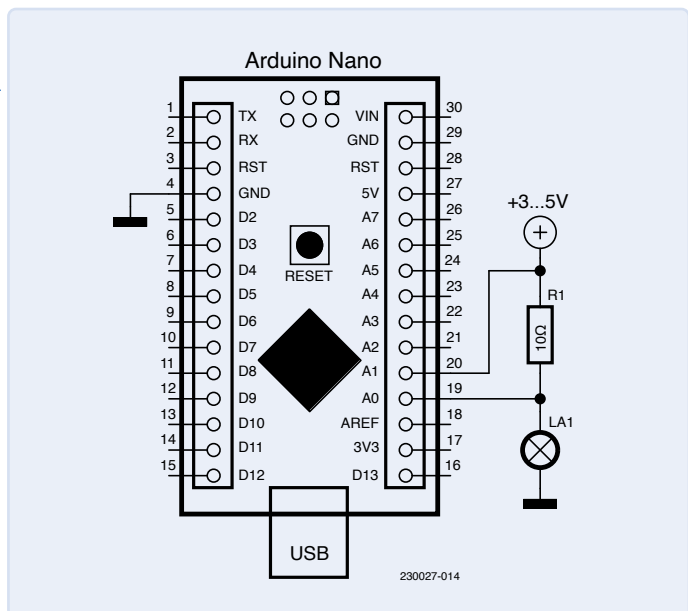
Bij een dergelijk signaal moet rekening worden gehouden met het (tamelijk smalle) spanningsbereik. Bij een maximale spanning van minder dan 1 V moet het signaal worden versterkt of moet de interne referentiespanning van de ADC correct worden gekozen om onnodig grote kwantisatiefouten te voorkomen.

Stroommeting

Een ADC kan gewoonlijk alleen spanningen meten. Maar in veel gevallen moeten ook stromen worden gemeten. De eenvoudigste methode hiervoor is het gebruik van een shuntweerstand. Als het stroomverbruik van een schakeling moet worden gemeten, zijn er twee mogelijkheden: low side- en high-side meten. **Figuur 12** toont een uiterst eenvoudige schakeling voor een high-side meting met een Arduino Nano.



Figuur 12. High-side meting.



Figuur 13. High-side meting voor en na de shuntweerstand.

De shuntweerstand wordt tussen de positieve voedingsspanning en de belasting opgenomen. Om de stroom door de weerstand en dus door de belasting te bepalen, moet de spanningsval over de weerstand worden gemeten. Indien de (constante en bekende) voedingsspanning van de belasting kleiner is dan of gelijk aan de referentiespanning van de ADC, kan de spanningsval over de shunt relatief eenvoudig worden bepaald. Als de voedingsspanning van de belasting echter varieert is (maar nog steeds lager dan de referentiespanning van de ADC), moeten de spanningen aan beide zijden van de shuntweerstand worden gemeten, zoals geschetst in **figuur 13**.

Toegegeven, dit voorbeeld is erg geïdealiseerd omdat de voedingsspanning van de belasting kleiner is dan of gelijk is aan de referentiespanning van de ADC. Als de voedingsspanning groter is dan de referentiespanning van de ADC of zelfs van de MCU, wordt het moeilijker.

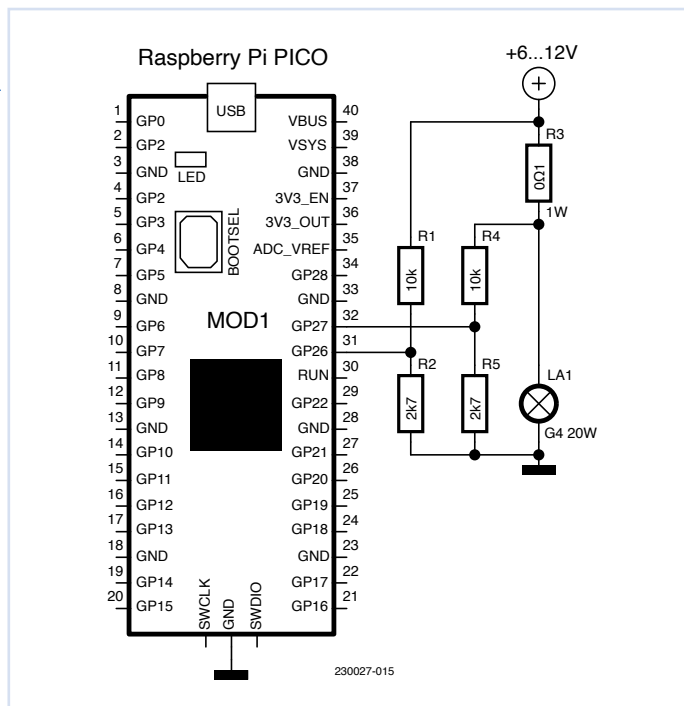
De stroom door een halogeenlamp meten

In het volgende voorbeeld willen we het stroomverbruik van een dimbare G4-halogeenlamp meten en monitoren, die wordt gevoed met een maximale gelijkspanning van 12 V. De lamp mag maximaal 20 W 'verstoken' (1,67 A bij 12 V). Een stroom van meer dan 2 A of minder dan 25 mA moeten door de MCU worden geïnterpreteerd als 'lamp defect'.

We zouden nu spanningsdelers kunnen gebruiken om de spanning voor en na de shuntweerstand in een bereik te brengen dat door de ADC kan worden verwerkt. **Figuur 14** toont een schakeling om dit voor elkaar te krijgen.

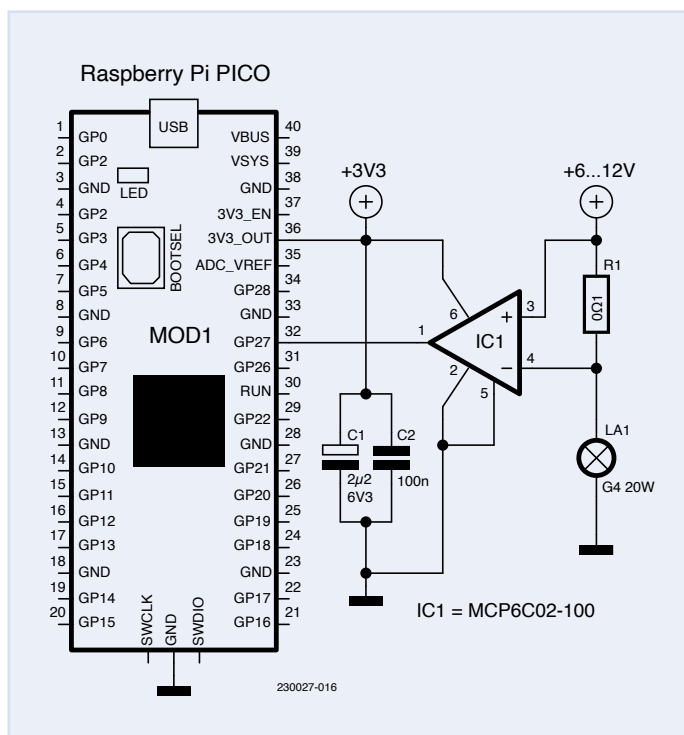
De MCU wordt gevoed met 3,3 V en de weerstanden zijn zo gekozen dat er bij 12 V maximaal 2,6 V op de pennen van de ADC komt te staan. De waarde van de shuntweerstand is 0,1 Ω. Bij 2 A veroorzaakt deze een spanningsval van 0,2 V, maar dissipeert nog steeds 0,4 W als warmte. De methode werkt in principe, maar is niet erg slim, zoals we straks zullen zien.

De spanningsdeler reduceert het meetbereik nog verder, zodat de ADC van de MCU een maximale spanningsval van minder dan 0,05 V bij 2 A moet kunnen verwerken. Ook het feit dat de spanningsval wordt gemeten via twee ADC-kanalen maakt de verdere verwerking niet eenvoudiger.

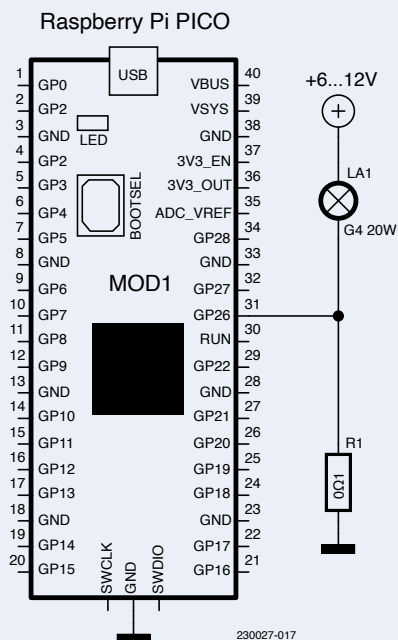


Figuur 14. Aangepaste schakeling met spanningsdelers.

In dit geval is het gebruik van een high-side stroommeetversterker zoals de INA138 van Texas Instruments of de MCP6Co2 van Microchip een veel betere oplossing. Bovendien kan de waarde van de shuntweerstand dan aanzienlijk kleiner zijn. Een shunt van 10 mΩ dissipeert veel minder dan zijn tien keer zo grote tegenhanger. **Figuur 15** toont de schakeling met stroommeetversterker. De uitgangsspanning is evenredig met de spanningsval over de weerstand en kan dus gemakkelijk worden verwerkt met een ADC-kanaal.



Figuur 15. Schakeling met stroommeetversterker.



Figuur 16. Low-side meting.

Low-side meting

In plaats van een shuntweerstand direct voor de belasting te plaatsen, kan deze ook worden aangesloten tussen de belasting en massa. Laten we bij het voorbeeld van de halogeenlamp blijven. **Figuur 16** toont zo'n low-side meting.

Hier wordt een weerstand van $0,1\ \Omega$ gebruikt, die bij 2 A een spanningsval veroorzaakt van 0,2 V. Het voordeel is nu dat de potentiaal van de te meten spanningen onder die van de referentiespanning van de ADC en de voedingsspanning van de MCU ligt. Bij 25 mA bedraagt de spanningsval echter slechts 2,5 mV (0,1 mV per mA). Onder ideale omstandigheden zou de 12bit-ADC met een referentiespanning van 3,3 V nog steeds een resolutie van 0,8 mV per bit hebben. Er moet echter worden gekeken naar het effectieve aantal bits (Effective Number Of Bits, ENOB) van de ADC, rekening houdend met de meetfouten die in de praktijk optreden als gevolg van kwantisatiefouten, vervorming, ruis en dergelijke. Voor de RP2040 is een ENOB van ongeveer 9 gespecificeerd, wat neerkomt op ongeveer 6,44 mA per bit. Voor deze toepassing zou de resolutie van de stroommeting net voldoende zijn. Als echter een hogere resolutie nodig is, kan een opamp worden ingezet, waardoor dan ook een kleinere weerstandswaarde mogelijk is.

Opamps en temperatuursensoren

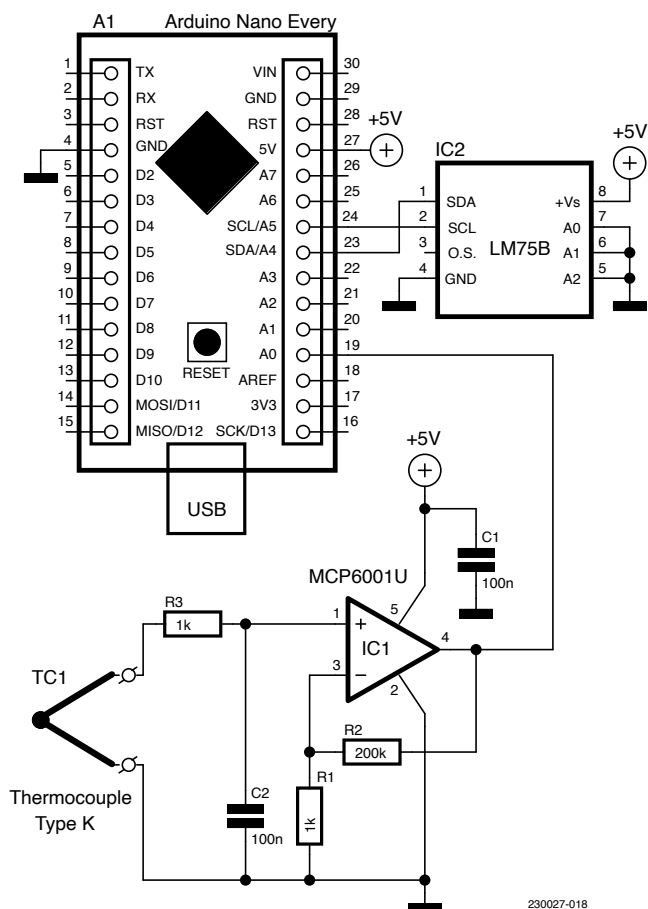
Hele bibliotheken zijn volgeschreven over operationele versterkers. Bij de juiste bedrading hebben deze fundamentele en veelzijdige componenten een talloze toepassingen, waarvan er in dit artikel al enkele zijn genoemd. Bij het meten van spanningen zijn het niet alleen shuntweerstandsen waarvan de spanningen zich in een minder dan ideaal bereik bevinden voor de ADC van een MCU. Temperatuursensoren zijn ook kandidaten voor deze behandeling, tenminste als het zuiver analoge, resistieve types zijn zoals een PT100 of thermokoppels van twee verschillende metalen (in tegenstelling tot exemplaren met een digitale interface zoals de DS18B20 of de DHT11).

Thermokoppels

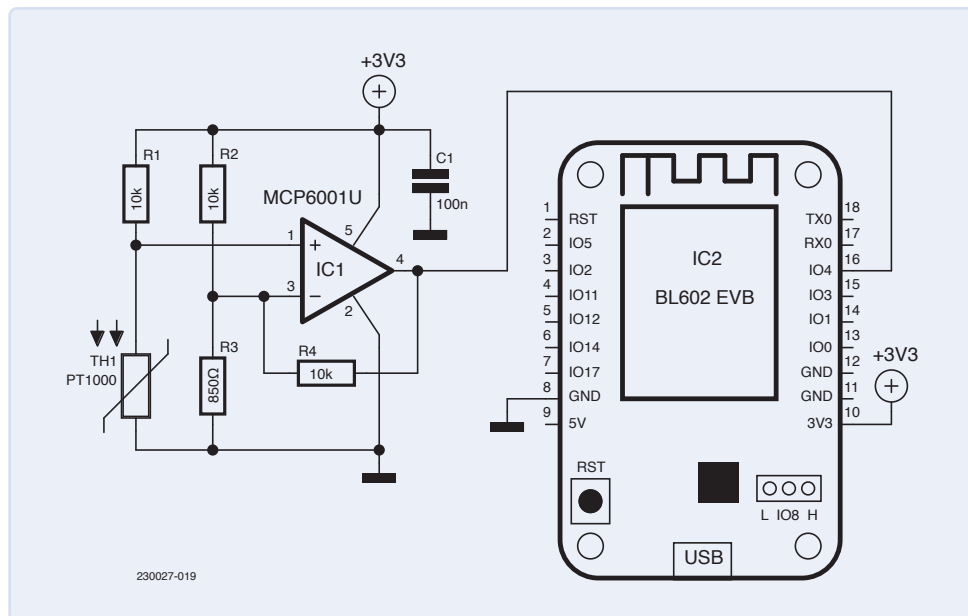
Deze uit twee verschillende metalen bestaande elementen fungeren als temperatuurafhankelijke spanningsbronnen. Het veelgebruikte K-type thermokoppel dient als voorbeeld. Bij $21\ ^\circ\text{C}$ of $69,8\ ^\circ\text{F}$ levert zo'n component een spanning van 0,838 mV als gevolg van het Seebeck-effect; bij $22\ ^\circ\text{C}$ of $71,6\ ^\circ\text{F}$ is dat 0,879 mV. Het verschil van slechts ongeveer $40\ \mu\text{V}$ per graad celsius stelt de ADC van een MCU zwaar op de proef. Bovendien is bij gebruik van thermokoppels ook koude-las-compensatie nodig [4].

Een opamp kan worden gebruikt om de spanning van een thermokoppel te versterken. **Figuur 17** toont een voorbeeld van een dergelijke schakeling, opgezet voor een Arduino Nano Every, waarbij een temperatuurbereik van $0\ ^\circ\text{C}$ tot $400\ ^\circ\text{C}$ wordt verwacht (0...16,396 mV thermokoppel-spanning).

Hier wordt een niet-inverterende opamp gebruikt met een versterking $A = 200$. Om storingen aan de ingang van de operationele versterker te minimaliseren is een RC-filter toegevoegd; een eenvoudige digitale temperatuursensor van het type LM75 zorgt voor de temperatuurdetectie ten behoeve van de koude-las-compensatie.



Figuur 17. Uitlezen van een thermokoppel met de Arduino Nano Every.



Figuur 18. Brug van Wheatstone voor een PT1000.

PT100 en PT1000

PT100 en PT1000 zijn temperatuurafhankelijke weerstanden met een waarde van 100 Ω (PT100) of 1000 Ω (PT1000) bij 0 $^{\circ}\text{C}$. Om de temperatuur te berekenen moet eerst de weerstandswaarde van de sensor worden gemeten. Een goede mogelijkheid om die weerstandswaarde nauwkeurig te bepalen is de brug van Wheatstone. Net als bij thermokoppels zijn de spanningen bij PT100 en PT1000 vrij gering, en moet het kleine verschil tussen twee meetpunten worden bepaald. Ook hier biedt een opamp (aangesloten als niet-inverterende verschilversterker) de nodige assistentie. De schakeling voor een board met een voedingsspanning van 3,3 V, zoals het Pinecone BL602-evaluatieboard, is getekend in **figuur 18**.

Het spanningsverschil tussen de positieve en negatieve ingang van de opamp wordt versterkt. Maar wat als de temperatuur lager wordt dan 0 $^{\circ}\text{C}$? In dat geval zou de opamp een negatieve spanning moeten uitvoeren, wat niet mogelijk is. Een optie is hier om de weerstand R3 van 10 k Ω te vervangen door een weerstand van 850 Ω . De spanning tussen R1 en R2 zou dan 0,3917 V zijn, wat dan ook geldt voor het punt tussen R3 en de PT1000 bij een temperatuur van -38 $^{\circ}\text{C}$. Het meetvenster zou dus beginnen bij -38 $^{\circ}\text{C}$ met een spanningsverschil van 0 V. Bij 20 $^{\circ}\text{C}$ zou het spanningsverschil in de schakeling 0,09444 V bedragen.

Samenvattend

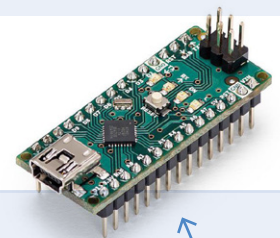
De verwerking en uitvoer van analoge waarden kan met acceptabele inspanningen plaatsvinden met een microcontroller. Ook al lijkt het proces niet zo eenvoudig als het aansturen van een digitale

I/O-pin, de omgang met de ADC is iets wat iedereen zou moeten proberen. De hier getoonde voorbeelden zijn bedoeld als suggesties voor eigen pogingen om de microcontroller te laten interageren met de analoge wereld buiten zijn behuizing. 

230027-03

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de redactie van Elektor via redactie@elektor.com.



Gerelateerde producten

- > **Arduino Nano (SKU 17002)**
www.elektor.nl/17002
- > **Raspberry Pi Pico RP2040 (SKU 19562)**
www.elektor.nl/19562
- > **Pinecone BL602 Evaluation Board (SKU 19914)**
www.elektor.nl/19914

WEBLINKS

- [1] Guido Nopper, "Eenvoudige A/D-converter met PLD", Elektor september/oktober 2019:
<http://www.elektormagazine.nl/magazine/elektor-112/51093>
- [2] Hardware design with RP2040: <https://datasheets.raspberrypi.com/rp2040/hardware-design-with-rp2040.pdf>
- [3] Giovanni Carrera, "Geïsoleerde analoge uitgang voor Arduino Uno", Elektor november/december 2022:
<http://www.elektormagazine.nl/magazine/elektor-282/61132>
- [4] Thermokoppel (Wikipedia): <https://nl.wikipedia.org/wiki/Thermokoppel>