

## 6 Analooq input/ ADC

---

### 6.1 Einddoel

Opdracht 1: Met 3 potmeters wordt een schakeling en code gemaakt om de kleur van een RBG-LED in te stellen.

Opdracht 2: Twee joystick controllers ga je uitlezen wat de stand is van de controllers. Als je op de knop van de controller drukt wordt de waarde van dat moment op het scherm getoond.

Als je klaar bent met de opdracht lever je in It's learning in:

- De definitieve python code waar de verschillende opdrachten in verschillende functies staan.
- Een foto van de opstelling waarop duidelijk te zien is hoe alles in aangesloten.
- Een mp4-filmpje met een demo van je programma.

Als dit gedaan is laat je de opdracht zien aan de docent zodat deze de opdracht kan goedkeuren.

### 6.2 Kennis

Voor deze opdracht heb je kennis nodig van microPython.

### 6.3 Benodigdheden

Voor deze opdracht heb je de onderstaande materialen nodig. Controleer aan het begin of al deze spullen aanwezig zijn. Bij het opruimen dien je weer te controleren of alles aanwezig is. Indien er iets defect is geraakt moet je de docent op de hoogte brengen.

- Raspberry Pi Pico – H
- UBS-usb mini kabel
-

6.4 Opdracht

6.4.1 Wat is analoog

We werken veel met digitale poorten. Digitaal betekend dat de waarde van de poorten 0 of 1 zijn, True of False, 0V of 3.3V. Dit is heel handig voor een computer want die kan keuzes maken op deze boolean. In de wereld buiten de computer zijn er alleen nog veel meer waarden. Tussen 0 en 3.3V zitten een oneindig aantal waarden zoals 0.1V, 2.000066686V, 2.999999999V etc. etc. etc.

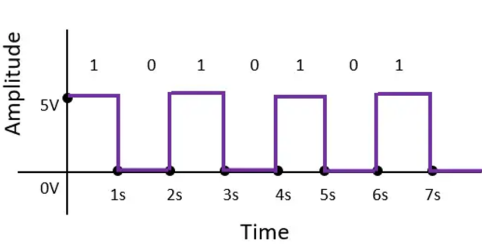


Fig. a digital signal

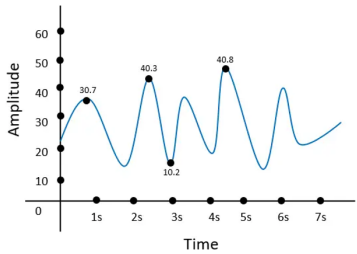
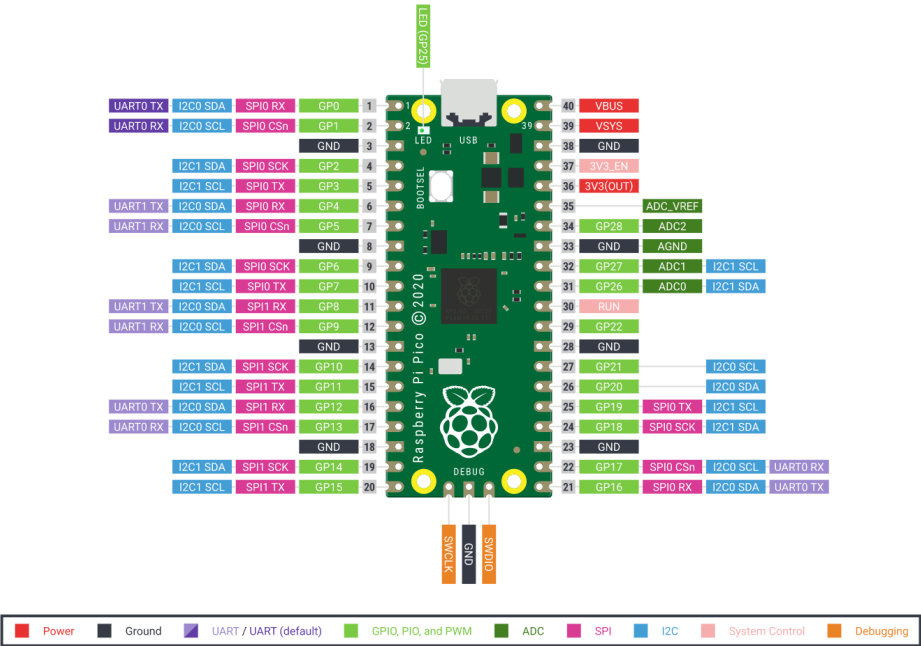
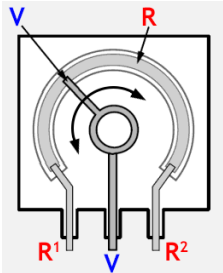


Fig. an analog signal

Deze waardes kan je met de Raspberry pi Pico gelijk uitlezen met de donkergroene ADC pinnen. Op pin 31 zit ADC0, pin32 zit ADC1 en pin34 zit ADC2. Heb je meer analoge ingangen nodig dan zal je van een externe ADC-converter gebruik moeten maken. Deze externe ADC-converter heb je ook nodig als je met een Raspberry pi gaat werken, de pi heeft namelijk geen ingebouwde converters.

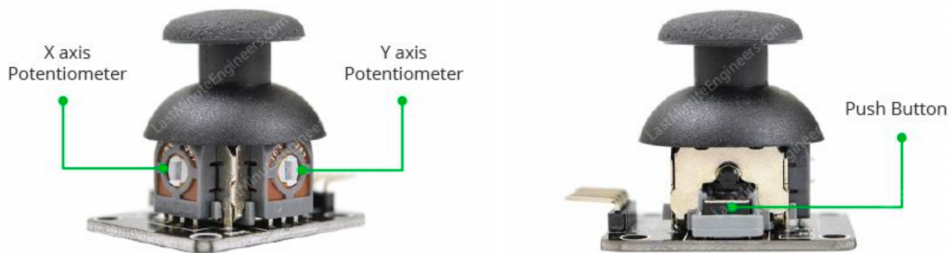


## 6.4.2 Een potmeter uitlezen



Een potmeter oftewel een potentiometer is een regelbare weerstand waar 3 pootjes aan zitten. Op de linker poot sluit je de GND aan, en op de rechter de +3.3V. Tussen de twee buitenste pootjes zit een materiaal dat een vaste weerstand heeft. Laten we zeggen dat de weerstand tussen R1 en R2 1000 Ohm is. Als we pootje V helemaal naar R1 draaien zien we dat de weerstand dan 0 Ohm is tussen R1 en V, en daardoor op V de spanning van 0V staat. Draaien we V naar R2 dan staat op V een spanning van 3.3V. Alle posities ertussen geeft een spanning tussen deze twee waarden naar ratio van de weerstand.

In de Thumbstick van een controller voor een spelcomputer zitten twee van deze potmeters en een drukknop.

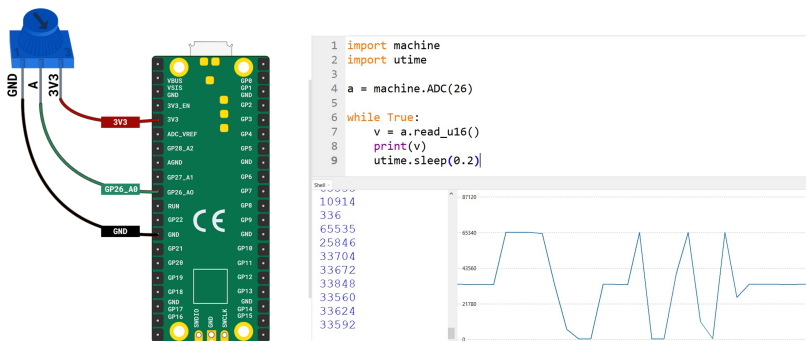


Een uitgebreide uitleg kan je bekijken op: <https://www.youtube.com/watch?v=WHdNMwWzUk8>

Als de joystick in rust staat, is hetzelfde als de potmeters halverwege staan. Op het pootje V zal dan de helft van de spanning staan, dus  $3.3V / 2 = 1.65 V$

**Als je een drift op je controller hebt, hoe zal dat dan komen? Vertel dit aan het einde aan de docent.**

We sluiten een potmeter aan op de onderstaande manier, en kijken welke waarden er op het scherm getoond worden als de de Thumbstick bewegen.



Als we dit programma runnen dan zien we als we niets doen een waarde tussen de 33000 en 34000, als we de Thumbstick bewegen dan is de waarde minimaal rond de 300, en maximaal 65535.

We zien helemaal geen waarden tussen de 0 en 3.3V wat we verwachten. Dit komt doordat de analoge waarde wordt omgezet in een digitale waarde, vandaar ook de naam **Analoog Digitaal Converter**.

Met het menu item: view-> Plotter zie je de waarden die gemeten zijn.

#### 6.4.3 Nauwkeurigheid

We zien een waarde tussen de 0 en 65536. Deze waarde wordt met een 12 bits ADC omgezet. 12 bits betekend dat er  $2^{12} = 4096$  verschillende waardes gemeten kunnen worden. Als we 3.3V delen door 4096 dan zien we dat het kleinste verschil wat we kunnen meten  $\frac{3.3}{4096} = 0,00081 \text{ volt}$  is. Als we een externe ADC aansluiten op de pico met 16 bits dan kunnen we nog nauwkeuriger meten. Er kunnen dan 65536 verschillende stappen gemaakt worden, waardoor de nauwkeurigheid  $\frac{3.3}{65536} = 0.000050 \text{ volt}$  is.

#### 6.4.4 Opdracht 1: potmeters

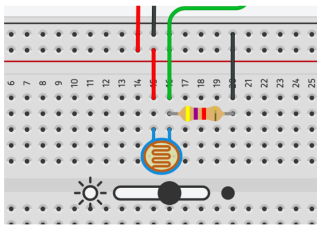
Sluit de tweede potmeter van de Thumbstick aan op poort 27 en de drukknop op poort 28. Als je de Thumbstick naar boven en naar beneden beweegt dan gaan de Leds 0 t/m 7 aan, aan de hand van de stand van de Thumbstick. Als je de andere beweegt dan gaan 8 t/m 15 aan.

Druk je op de knop dan kan je de Thumbstick even loslaten want dan blijft de huidige stand 5 seconden staan.

#### 6.4.5 Opdracht 2: Weerstanden die in waarde veranderen.

Het principe van de spanningsdeler wordt niet alleen bij een potentiometer gebruikt, maar zien we terug in heel veel elektronische opstellingen. Als we een vaste weerstand hebben en een andere weerstand die van waarde verandert door de invloed van licht (LDR) of temperatuur (TMP36, PTC, NTC) of druk (LM35) hebben we weer een spanningsdeler die op de analoge poort uitgelezen kan worden.

Hieronder zien we de opstelling van een **Light Depending Resistor**. Als er meer licht op de LDR schijnt dan zal de weerstand dalen. De LDR staat in serie met een gewone weerstand van 4K7  $\Omega$ . Dit lijkt een beetje op het draaien van de potmeter.



Maak de bovenstaande opstelling en kijk hoe de waarde wijzigt. Aan de hand van de waarden die je hebt bepaald laat je de leds 0 t/m 15 aan en uitgaan. Als er veel licht is gaan alle leds aan, als er geen licht is gaan ze allemaal uit en alle tussenliggende waarden.

Op <https://www.youtube.com/watch?v=WHdNMwWzUk8> wordt dit systeem uitgelegd voor een Arduino Uno. Let op dat dit een andere programmeertaal is, maar het principe is hetzelfde.