


Voorkennis	Aansturen van een Pin op een ESP32. Opvragen van tijd op een ESP32.	Niveau	
Benodigheden	ESP32 en dobbelsteen	Leerdoelen	

Een dobbelsteen

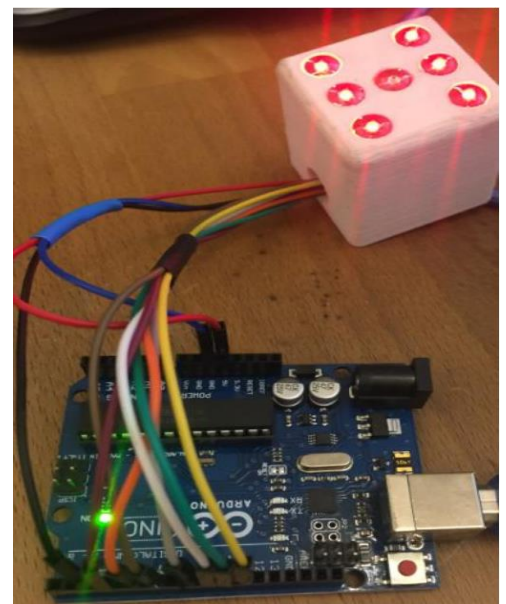
De elektronische dobbelsteen heeft 7 LEDs voor het weergeven van de 'gegooide' ogen. In de elektronische dobbelsteen is een kwikschakelaar die wel of geen contact maakt als je de elektronische dobbelsteen schudt.

De elektronische dobbelsteen heeft twee groepen aansluitdraden. Een groep heeft een **RODE draad** (3.3), een **BLAUWE draad** (GND) en een **ZWARTE draad** (schakeldraad is 3.3V of 0V afhankelijk van de kwikschakelaar). De andere groep draden (allerlei keuren) is om de 7 LEDs aan te sturen. Alle benodigde weerstanden voor de LEDs en schakelaar zitten in de elektronische dobbelsteen, dus daar hoeft jij je geen zorgen over te maken.

Vraag de mentor even wat meer uitleg te geven over de gebruikte hardware en een demo van wat de bedoeling is.

We gaan gebruik maken van enkele modules.

```
import machine # Om de pinnen te kunnen aan sturen
import time   # Om de tijd te kunnen opvragen
import random # Voor willekeurig nummers
```



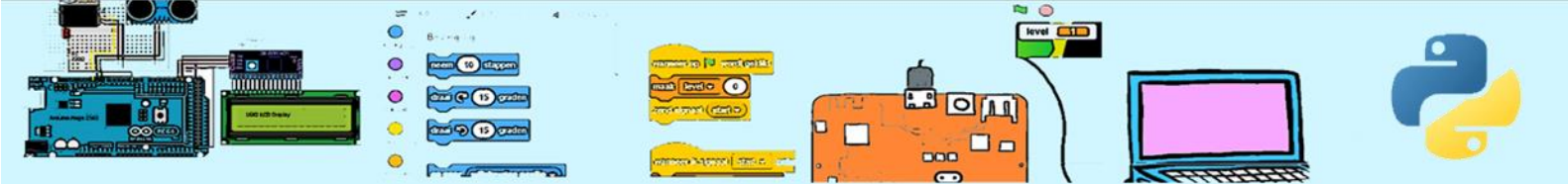
1. Controle van de pinnummers

De dobbelsteen is al aangesloten op de ESP32 (computer). Probeer te bepalen welke pin welke led bestuurt. Neem onderstaande over en vervang de ?? met het juiste pinnummer. Deze benaming gaan we later in ons script gebruiken.

```
ledLBpin = machine.Pin(??, machine.Pin.OUT) # led Linkerkant Boven
ledRBpin = machine.Pin(??, machine.Pin.OUT) # led Rechterkant Boven
ledLMpin = machine.Pin(??, machine.Pin.OUT) # led Linkerkant Midden
ledMMpin = machine.Pin(26, machine.Pin.OUT) # led Midden Midden
ledRmpin = machine.Pin(??, machine.Pin.OUT) # led Rechterkant Midden
ledLOpin = machine.Pin(??, machine.Pin.OUT) # led Linkerkant Onder
ledROpin = machine.Pin(??, machine.Pin.OUT) # led Rechterkant Onder
```

Doe hetzelfde voor de pin die verbonden is met de kwikschakelaar.

```
schakelPin = machine.Pin(??, machine.Pin.IN, machine.Pin.PULL_DOWN)
```



2. Eenvoudige functies

We gaan eerst eenvoudige functie maken die details van het aansturen van de led's verbergen. Na het maken van een functie moet je ze testen.

Maak een functie die alleUitzet().

Maak een functie die alleAanzet().

Maak een functie met de naam een() die zorgt dat alleen de ledMM brandt. Tip: zet eerste alle leds uit.

Maak een functie met de naam twee() die zorgt dat alleen de ledLB en ledRO branden.

Maak een functie met de naam drie() die zorgt dat LedLB, LedRO en ledMM branden. Tip: 2 van deze leds komen overeen met de leds in twee()).

Maak een functie met de naam vier() die zorgt dat ledLB, ledRB, ledLO en led RO branden.

Maak een functie met de naam vijf() die naast de leds in vier() ook ledMM laat branden.

Maak een functie met de naam zes() die alle Leds laat branden behalve ledMM.

3. Een willekeurigeUitkomst()

Maak een functie die een willekeurige uitkomst laat zien. Tip: de functie `random.randint()` genereert een willekeurig getal. Tip2: gebruik de eerder gemaakte functies `een()`, `twee()`, ... , `zes()` om het resultaat te laten zien. Test de functie.

4. Een worp()

Maak een functie `worp()`. De worp moet een gooi van de dobbelsteen na doen. Een echte dobbelsteen rolt en alvorens tot stilstand te komen zijn er meerdere uitkomsten zichtbaar. Maak ook de digitale dobbelsteen een beetje spannend door meerdere uitkomsten te generen alvorens tot rust te komen. Gebruikt "willekeurigeUitkomst()" en probeer ook de rollengte willekeurig te maken. Maak de start van een worp duidelijk door een vast patroon te laten oplichten.

5. Main()

Maak het hoofdprogramma (`main()`). Lees in een loop de status uit van de kwikschakelaar. Indien er contact wordt gemaakt roep je de functie `worp(??)` aan. Voorkom dat de kwikschakelaar continue wordt uitgelezen door een `time.sleep_us(1)` in de loop op te nemen. Hoeveel seconden slaapt de controller dan?