



Unit 6: Het gebruik van de ti_hub module

Oefenblad 2: Werken met een externe sensor

In deze les lezen we een externe sensor af met de TI-Innovator Hub in een Python programma.

Doelen :

- Het aansluiten van een externe sensor.
- Het gebruik van de Ranger.

Naast de ingebouwde apparaten kun je ook externe input- en output apparaten aansluiten aan de hub.

We gaan een programma maken dat m.b.v. een afstandssensor (Ranger) de afstand tot een voorwerp meet.

Ook moet het programma een geluid geven waarvan de toonhoogte afhangt van de gemeten afstand.

Sluit de Ranger aan in poort IN1. (deze poort zit aan een van de zijanten van de hub) en sluit de hub aan op de rekenmachine of de computer.



Open een nieuw Python programma en importeer de Hub-module.

Om de Ranger te kunnen gebruiken moeten we aangeven in welke poort hij zit aangesloten en we moeten hem nog een naam geven.

Kies uit het TI Hub-menu: Add Input Device, en kies dan voor Ranger.

Op het scherm verschijnt `var=ranger("port")`.

De cursor staat nu op de plek waar je de naam kan intypen.

Typ een naam in (bijvoorbeeld `rg`) en geef aan in welke poort de Ranger is aangesloten (`IN1` in dit geval).

```

1.1 1.2 *Doc RAD
*test.py 4/4
from ti_hub import *
rg=ranger("IN 1")

```

Het aflezen van de Ranger gaat met de opdracht `rg.measurement()`.

Je kunt dit makkelijk invoeren door de variabele-naam (`rg`) in te typen direct gevolgd door een punt.

Als je de gemeten waarde wilt gebruiken dan kun je die opslaan in een variabele (bijvoorbeeld `a`).

```

1.1 1.2 *Doc RAD
*test.py 5/5
from ti_hub import *
rg=ranger("IN 1")
a=rg.measurement()

```

Als je het programma van hiernaast uitvoert dan wordt de gemeten afstand afgedrukt.

De gemeten afstand is in meters.

```

1.1 1.2 *Doc RAD
*test.py 6/6
from ti_hub import *
rg=ranger("IN 1")
a=rg.measurement()
print(a)

```



Pas nu het programma aan zodat het voortdurend de afstand blijft meten en afhankelijk van de uitkomst een toon laat horen (zoals een parkeersensor werkt)

Gebruik hiervoor de while lus.

Neem de volgende voorwaarden:

Als de afstand groter is dan 0.4 m dan moet de frequentie 1000Hz zijn.

Tussen 0.2 m en 0.4 m is de frequentie 800Hz.

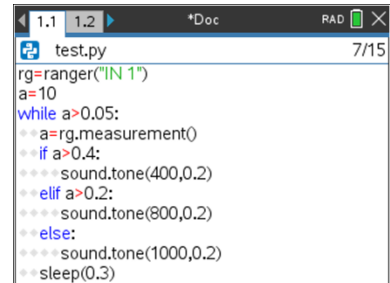
Tussen 0.05 m en 0.2 m is de frequentie 400Hz.

Als de afstand kleiner is dan 0.05 m (5 cm) dan moet het programma stoppen.

Kies voor de toonduur 0.2 sec en wacht telkens 0.3 sec tot de volgende meting.

Een mogelijk programma staat hiernaast.

Je kunt nu ook experimenteren met de afstanden en de toonduur.



```
1.1 1.2 *Doc RAD 7/15
test.py
rg=ranger("IN 1")
a=10
while a>0.05:
    a=rg.measurement()
    if a>0.4:
        sound.tone(400,0.2)
    elif a>0.2:
        sound.tone(800,0.2)
    else:
        sound.tone(1000,0.2)
    sleep(0.3)
```