

Kapitel 6: Ti_hub-modulen

Övning 2: TI-Innovator Hub, Python och sensorer

I den här lektionen ska vi titta på hur man kan läsa in data från externa sensorer.

Mål:

- Ansluta externa sensorer
- Använda avståndssensorn.

Förutom de inbyggda enheterna kan du också ansluta externa in- och utdataenheter till hubben.

Vi ska göra ett program som kommer att använda en avståndssensor (Ranger) för att mäta avståndet till ett objekt.

Programmet ska också producera ett ljud vars tonhöjd beror på det uppmätta avståndet.

Anslut avståndssensorn till port IN1 – som finns på ena sidan av hubben och anslut hubben till en handenhet eller dator.



Öppna ett nytt Python-program och importera ti_hub-modulen.

För att kunna använda avståndsmätaren så måste vi ange vilken port den är ansluten till och vi måste ge den ett annat namn.

Från menyn TI Hub väljer du Lägg till inmatningsenhet och sedan väljer du Ranger.

Var = ranger ("port") visas på skärmen.

Markören är nu där du kan ange namnet.

Skriv in ett namn (till exempel rg) och ange vilken port Ranger är ansluten till (IN1 i detta fall).

```
1.3 1.4 1.5 *Dok RAD 4/4
from ti_hub import *
rg=ranger("IN 1")
```

Att läsa data från avståndsmätaren görs med kommandot **rg.measurment()**.

Du kan enkelt ange hitta detta genom att skriva variabelnamnet, rg, omedelbart följt av en punkt.

Om du vill använda det uppmätta värdet kan du lagra det i en variabel, till exempel a.

```
1.3 1.4 1.5 *Dok RAD 4/4
from ti_hub import *
rg=ranger("IN 1")
a=rg.measurement()
```

Om du kör programmet till höger kommer det uppmätta avståndet visas.

Det uppmätta avståndet är i meter.



```
1.3 1.4 1.5 *Dok RAD 6/6
*Range.py
from ti_hub import *

rg=ranger("IN 1")
a=rg.measurement()
print(a)
```

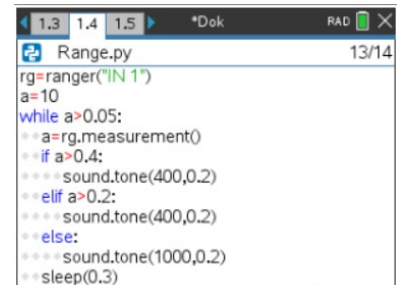
Ändra programmet så att det kontinuerligt mäter avståndet och avger en ton beroende på resultatet (som en parkeringssensor fungerar). Använd en while-loop för detta.

Om avståndet är större än 0,4 m ska frekvensen vara 1000Hz.
Mellan 0,2 m och 0,4 m. ska frekvensen vara 800Hz.
Mellan 0,05 m och 0,2 m ska frekvensen vara 400 Hz.
Om avståndet är mindre än 0,05 m (5 cm) ska programmet stoppas.

Välj 0,2 sek för tonens längd och vänta 0,3 sek för nästa mätning.

Ett möjligt program visas till höger.

Du kan också experimentera med avstånd, frekvenser och tonlängder.



```
1.3 1.4 1.5 *Dok RAD 13/14
Range.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(400,0.2)
  else:
    sound.tone(1000,0.2)
  sleep(0.3)
```