

Kapitel 6: Ti_hub-modulen

Övning 1: Python och TI-Innovator™ Hub

I den här lektionen kommer vi att använda TI-Innovator™- Hub i ett Python-program.

Mål:

- Använda ti_hub modulen
- Några grundläggande användningar av hubben

I ti_hub-modulen hittar du kommandona för att kommunicera med TI-Innovator-hubben från Python.

Vi ska skapa ett program som tändar den färgade lysdioden i hubben och använder den inbyggda högtalaren.

Anslut hubben till handenheten eller till datorn via ingången märkt DATA

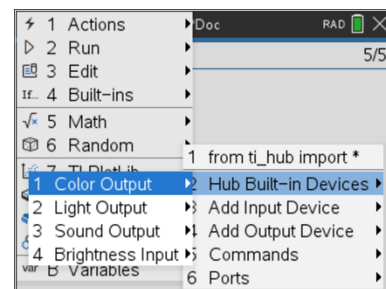


Öppna ett nytt Python-program där du inte väljer någon typ. För att använda Innovator-hubben behöver du ti_hub-modulen. Välj från menyn TI Hub och sedan alternativ 1: från ti_hub import *. Du har då tillgång till alla hub-funktioner.



Hubben har en LED som kan lysa i alla färger (r,g,b). Du kan sätta färgen med kommandot **color.rgb(r,g,b)**. Du hittar det under meny>TI Hub>Hubb inbyggda enheter>Färgutgång>1: rgb(red, green, blue).

Kommandot blink() i samma meny har två parametrar, blinkfrekvens och hur lång tid det ska blinka.





I exemplet till höger har röd färg valts och att lampan ska blinka i 3 sekunder med en frekvens av två gånger per sekund.

```
1.2 1.3 1.4 *Dok RAD 4/5
Hubb.py
from ti_hub import *

color.rgb(255,0,0)
color.blink(2,3)
```

Vi kan också lägga till ljud.

I ti_hub-menyn kan du välja meny>TI Hub>Hubb inbyggda enheter>Ljudutgång. där kan du välja tone(frekvens, tid)Spela nu en 800 Hz-ton i 2 sekunder.

```
1.2 1.3 1.4 *Dok RAD 6/7
Hubb.py
from ti_hub import *

color.rgb(255,0,0)
color.blink(2,3)

sound.tone(800,2)
```

Vi kan variara variara tonhöjden.

Skapa en slinga som kör frekvensen från 400 till 2000 Hz i steg om 100 Hz där varje ton varar i 0,2 sekunder.

```
1.2 1.3 1.4 *Dok RAD 8/9
*Hubb.py
from ti_hub import *

color.rgb(255,0,0)
color.blink(2,3)

for f in range(400,2000,100):
    sound.tone(f,0.2)
```

När du kör detta program märker du att tonerna inte varar 0,2 sekunder.

Det är viktigt att tiden mellan två på varandra följande toner också är minst 0,2 sekunders. Du kan använda kommandot **sleep()** för detta.

```
1.3 1.4 1.5 *Dok RAD 3/9
Hubb.py
from ti_hub import *

color.rgb(255,0,0)
color.blink(2,3)

for f in range(400,2000,100):
    sound.tone(f,0.2)
    sleep(0.2)
```