

Kapitel 6: Använda TI-Innovator Hub och TI-Innovator Rover

Övning 1:

Integrerade sensorer hos hubben

I denna första övning i kapitel 6 kommer du att upptäcka hur man använder biblioteket **ti_hub** för att styra integrerade komponenter hos hubben.

I den här övningen kommer du att använda **ti_Innovator**-biblioteket för att visuellt uppmärksamma en förändring i ljusintensiteten för att senare simulera en skymningsbrytare eller spela in en serie mätningar vid soluppgång eller skymning. Du kommer också att börja fundera på hur du kombinerar det här biblioteket med de du redan känner till (**ti_plotlib** och **ti_system**) för att skapa ett mer komplett naturvetenskapligt projekt. Skriptet du ska skriva motsvarar följande enkla algoritim:

```
Mäta omgivande ljusintensitet:  
Lum0 ← mätning ± (tolerance?)  
Ändra ljusintensitet (lampa; täckning  
framför ljussensorn)  
Lum1 ← mätning  
If Lum1 > Lum 0:  
    Tänd RGB LED i rött (2s)  
Annars If Lum0 < Lum1:  
    Tänd RGB LED i grönt (2s)  
Annars: Gör ingenting
```

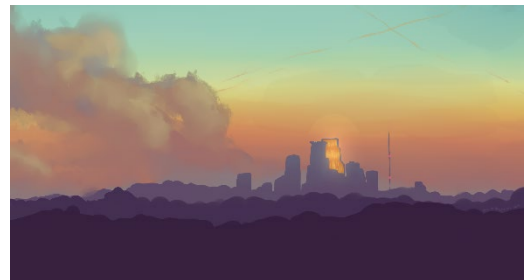
Starta ett nytt skript och döp det KAP6OVN1

Det här skriptet måste integrera **ti_hub**-biblioteket. För detta har du flera möjligheter:

- Tryck på f3 [Types] och välj alternativ 6 i menyn: 6:Hub Project. Bekräfta genom att trycka på **enter** två eller en gång och sedan OK.
- Du kan också börja från ett tomt skript och sedan manuellt införliva de bibliotek du behöver.
- Slutligen kan du också använda de bibliotek och variabler som har skrivits under TI STEM-projekten. Dessa kommer omedelbart att kopieras som "Projektobjekt". Det sista alternativet är bara intressant när du vill skapa ett skript som liknar dem som definierats under [STEM-projektet](#) (engelska) Se även den [svenska sidan för STEM-projekt](#)

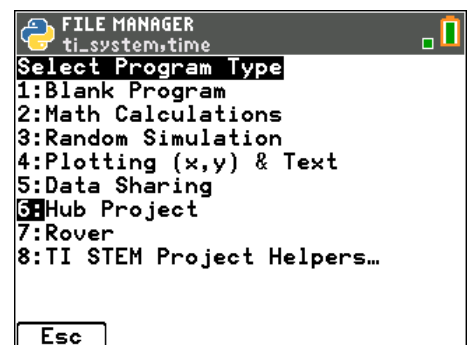
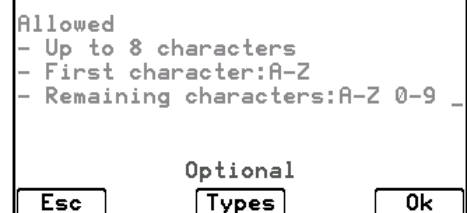
Syfte:

- Upptäcka hubb-modulen
- Skriva ett skript där man integrerar TI-Hub-biblioteket för de integrerade komponenterna



Skymning inträder när dag går mot natt då solen passerar horisonten och solskivan endast delvis är synlig.

Längden på skymningen beror på hur lång tid det tar för solen att komma 6° under horisonten. Detta i sin tur beror på hur flack solens skenbara bana är relativt horisonten.



Genom att välja typ **6: Project Hub** så får du skärmen här:

Vi kommer att använda **TI-Innovators** inbyggda ljusstyrkesensor och RGB-diod. För att skriptet ska kunna hantera dem, låt oss integrera motsvarande bibliotek. För att göra det, välj i **Modul-menyn** alternativ **6: ti_hub** och slutligen i undermenyn **1: Hub-Built-in devices**.

Kommentar: De två andra möjligheterna gäller sensorer och styrenheter som ansluts direkt till hubbens IN- och OUT-portar eller möjligtvis till kopplingsplattans portar (Breadboard, BB).

```

EDITOR: KAP60VM1
PROGRAM LINE 0004
from ti_system import *
from time import
    
```

Fns... a A # Tools Run Files

```

EDITOR: KAP60VM1
ti_hub module
Import Commands Ports Advanced
1: Hub Built-in devices...
2: Input devices...
3: Output devices...
    
```

Esc Modul

```

EDITOR: KAP60VM1
PROGRAM LINE 0006
from ti_system import *
from time import *
import color
import brightns
    
```

Fns... a A # Tools Run Files

```

EDITOR: KAP60VM1
Func Ctl Ops List Type I/O Modul
1: math...
2: random...
3: time...
4: ti_system...
5: ti_plotlib...
6: ti_hub...
7: ti_rover...
8: Color... <Hub Output>
9: Brightness... <Hub Input>
    
```

Esc Help

- För att programmet ska kunna visa information på en "ren" skärm rensar du skärmen med hjälp av instruktionen `disp_clr()` i `ti_system`-biblioteket.
- Skapa en `lum0`-variabel till vilket ett ljusstyrt mått tilldelas. Satsen `brightns.measurement()` finns i modul **9: Brightness**, som läses in så snart motsvarande modul importerats.
- Visa ett meddelande där användaren uppmanas att ändra ljusintensiteten i närheten av den inbyggda sensorn `disp_at()`.



- **Sätt programmet i vänteläge `disp_wait()`, den tid som ska ändras.**
- Skapa en `lum1`-variabel som den nya mätningen är tilldelad.
- Mätningarna jämförs sedan. Beroende på resultatet av villkorsinstruktionen tänds RGB-lampan i grönt eller rött under 2 sekunder.

```

EDITOR: KAP6OVN1
PROGRAM LINE 0001
Hub Projet
from ti_system import *
from time import *
import color
import brightns
#mät
disp_clr()
lum0=brightns.measurement()
disp_at(5,"ändra ljusstyrkan","1
eft")
disp_wait()
lum1=brightns.measurement()
#comparaison
if lum0<lum1:
*color.rgb(255,0,0)
*sleep(2)
*color.off()
elif lum0>lum1:
*color.rgb(0,255,0)
*sleep(2)
*color.off()
if lum0<lum1:
*color.rgb(255,0,0)
*sleep(2)
*color.off()
elif lum0>lum1:
*color.rgb(0,255,0)
*sleep(2)
*color.off()
else:
*color.off()_

```

Utvidgning av övningen

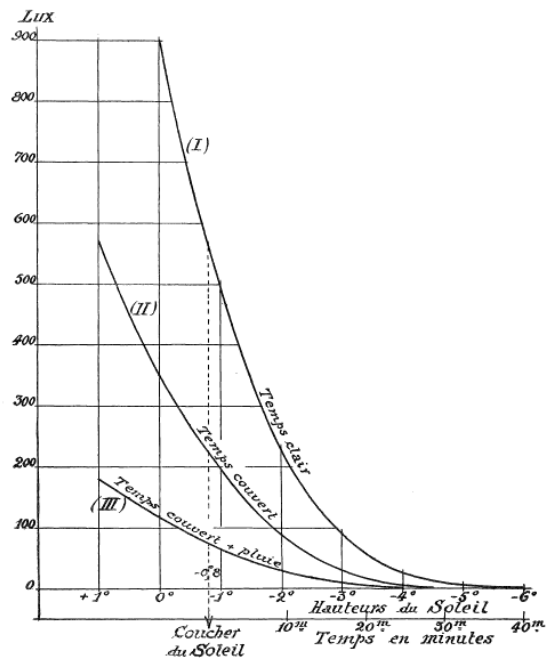
Ändra föregående skript eller skapa ett nytt med namnet KAP6OV1B. Det nya skriptet bör registrera mätningarna av ljusstyrkan i 40 minuter.

Kommentar: Observera att ljussensorn på TI-Innovator inte är kalibrerad i Lux, men detta spelar ingen roll så länge man bara är intresserad av ljusstyrkevariationerna och inte i själva mätningen i Lux.

Måtten sparas i en lista [].
 De som motsvarar tidsvärdet i en t[]-lista.

Skriptet som föreslås nedan ger också en grafisk representation av mätningarna. Du kan jämföra den med diagrammet här till höger.

Export till räknarens listeditor föreslås också.



Avtagande av naturligt ljus under den borgerliga skymningen

- Ordförklaringar: Temps clair= klart väder*
- Temps couvert=molnigt väder*
- Temps couvert + pluie= Molnigt väder + regn*
- Hauteurs du soleil = solhöjd*
- Coucher de soleil =Solnedgång*

a) Datainsamling

```

EDITOR: KAP6OV1B
PROGRAM LINE 0001
# Hub Projekt
from ti_system import *
from time import *
import ti_plotlib as plt
import brightns
disp_clr()
def bri(n):
    r=[]
    t=[]
    for i in range(n):
        r.append(brightns.measuremen
            t())
        t.append(i)
        sleep(60)
    return t,r
#grafisk representation
    
```

b) Grafisk representation

```

EDITOR: KAP6OV1B
PROGRAM LINE 0025
#grafisk representation
def graf(t,r):
    plt.cls()
    plt.auto_window(t,r)
    plt.labels("t(min)", "r")
    plt.title("skymning")
    plt.color(255,0,255)
    plt.scatter(t,r,"+")
    store_list("1",t)
    store_list("2",r)
    plt.show_plot()
    
```

Bri(n)-funktionen utför datainsamling varje minut i n minuter och returnerar t- och r-listorna.

Funktionen **graf(t,r)** ritar data t[] och r[] och exporterar dem sedan till räknarens listor L₁ och L₂.