

1. Dashboard

Bij het lezen/meten van sensoren kan met de statements `text_at()` en `cls()` van de TI Hub-module een elementair dashboard gecodeerd worden in het grafisch (handheld) venster van de Python shell:

- `text_at(row,"text","align")` row = 1..13 en align: left – center – right
- `cls()` wissen van het output-venster

Voor het runnen van deze commando's moet het document in Handheld Page Size-mode staan.

Voor het meten van de ingebouwde lichthelderheid sensor kan dit als volgt:

```
from ti_hub import *
while get_key() != "esc":
    ♦♦ helderheid=brightness.measurement()
    ♦♦ cls()
    ♦♦ text_at(5,"De helderheid is","center")
    ♦♦ text_at(6,"{0:5.3f}".format(helderheid),"center")
    ♦♦ sleep(0.5)
    ♦♦ get_key()
```



De boodschap die in het dashboard wordt weergegeven moet als een string ingegeven worden. Met de stringmethode `format()` kunnen meetwaarden toegevoegd worden.

Voor de bovenstaande code flinkt het dashboard bij een refresh van het venster. Gebruik om dit te voorkomen de functies `use_buffer()` en `paint_buffer()` van de TI Draw-module (waarover meer in BootCamp 4).

Het statement `use_buffer()` zorgt dat alle output voor het grafische venster van de Python shell uitgevoerd wordt in de achtergrond (geheugen) en dit tot `paint_buffer()` wordt uitgevoerd, dat alles van de buffer weergeeft in het venster.

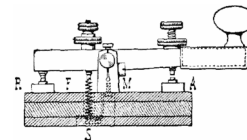
```
from ti_hub import *
use_buffer()
while get_key() != "esc":
    ♦♦ helderheid=brightness.measurement()
    ♦♦ cls()
    ♦♦ text_at(5,"De helderheid is","center")
    ♦♦ text_at(6,"{0:5.3f}".format(helderheid),"center")
    ♦♦ paint_buffer()
    ♦♦ sleep(0.5)
    ♦♦ get_key()
```



2. Morse-code

Morse-code werd in 1835 uitgevonden door Samuel Morse voor het versturen van berichten. Morse bestaat uit met tussenpauzes uitgezonden signalen. Voor de telegraaf had je de volgende keuzes uit twee toestanden:

- sleutel naar beneden (stroom) of naar boven (geen stroom)
- tijdsduur kort of lang



De hedendaagse Internationale Morse-code kent twee symbolen: punten en streepjes, ofwel *dits* en *dahs*. De lengte van de 'dit' bepaalt de snelheid waarmee de boodschap wordt verzonden en wordt als 'eenheid' gebruikt.

We coderen de Morse Code met als output een combinatie van de ingebouwde luidspreker en RGB-led en definiëren/coderen de Morse-code als een dictionary in een apart python programma alfabet.py.

alfabet.py

```
alfabet={
♦♦ "A": ".-.",
♦♦ "B": "-...",
♦♦ "C": "-.-.",
♦♦ "D": "-..",
♦♦ "E": ".",
♦♦ "F": ".-.-",
♦♦ "G": "--.",
♦♦ "H": "....",
♦♦ "I": "..",
♦♦ "J": ".-.-.-",
♦♦ "K": "-.-.",
♦♦ "L": ".-..",
♦♦ "M": "--",
♦♦ "N": "-.",
♦♦ "O": "---",
♦♦ "P": "-.-.",
♦♦ "Q": "--.-",
♦♦ "R": ".-.-",
♦♦ "S": "...",
♦♦ "T": "-.",
♦♦ "U": ".-.",
♦♦ "V": "-.-.",
♦♦ "W": "-.-.",
♦♦ "X": "-.-.",
♦♦ "Y": "-.-.",
♦♦ "Z": "--.."
}
```

International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.


A ● ———
B ——— ● ● ●
C ——— ● ——— ●
D ——— ● ●
E ●
F ● ● ——— ●
G ——— ——— ●
H ● ● ● ●
I ● ●
J ● ——— ——— ———
K ——— ● ———
L ● ——— ● ●
M ——— ———
N ——— ●
O ——— ——— ———
P ● ——— ——— ●
Q ——— ——— ● ———
R ● ——— ●
S ● ● ●
T ———

U ● ● ———
V ● ● ● ———
W ● ——— ———
X ——— ● ● ———
Y ——— ● ——— ———
Z ——— ——— ● ●

1 ● ——— ——— ———
2 ● ● ——— ———
3 ● ● ● ———
4 ● ● ● ● ———
5 ● ● ● ● ●
6 ——— ● ● ● ●
7 ——— ——— ● ● ●
8 ——— ——— ——— ● ●
9 ——— ——— ——— ——— ●
0 ——— ——— ——— ——— ———



Hieronder Python-code voor de TI-Innovator hub die Morse-code simuleert.

 morse.py

```
from ti_hub import *
from alfabet import *

# Lengte kort signaal 0.11 s
t=0.11

# Input boodschap en omzetten naar hoofdletters
while True:
tekst=input("Tekst (enter=stoppen): ")
♦♦ if tekst==" " or tekst[:3]=="esc":
♦♦♦♦ break
♦♦ tekst=tekst.upper()

# Output in 650 Hz en witte flits rgb(255,255,255)
# . = t en _ = 3*t
# tijd tussen twee letters 3*t
# tijd tussen twee worden 7*t
♦♦ for k in tekst:
♦♦♦♦ if k==" ":
♦♦♦♦ sleep(7*t)
♦♦♦♦ else:
♦♦♦♦♦♦ if k in alfabet:
♦♦♦♦♦♦♦♦ for d in alfabet[k]:
♦♦♦♦♦♦♦♦♦♦ if d==" .":
♦♦♦♦♦♦♦♦♦♦♦♦ x=t
♦♦♦♦♦♦♦♦♦♦♦♦ else:
♦♦♦♦♦♦♦♦♦♦♦♦ x=3*t
♦♦♦♦♦♦♦♦♦♦♦♦♦♦ sound.tone(650,x)
♦♦♦♦♦♦♦♦♦♦♦♦♦♦ color.rgb(255,255,255)
♦♦♦♦♦♦♦♦♦♦♦♦♦♦ sleep(x+t)
♦♦♦♦♦♦♦♦♦♦♦♦♦♦ color.off()
♦♦♦♦♦♦♦♦♦♦♦♦♦♦ sleep(3*t)
```