

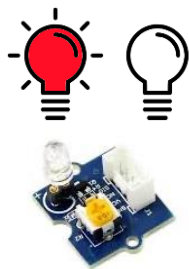
Ontwikkel en codeer een hitte-alarm dat afhankelijk van de temperatuur en al dan niet een baby in de auto (gesimuleerd door een magneet) een alarm laat afgaan en een boodschap weergeeft. Open het document HitteAlarm.tns in de folder Unfold.

Flikkerend LED

Sluit de LED-module aan op poort OUT 1.
Run de code flash.py op pagina 1.1!

```
from ti_hub import *
flash = led("OUT1")

for i in range(5):
    ♦♦ flash.on()
    ♦♦ sleep(1)
    ♦♦ flash.off()
    ♦♦ sleep(1)
```



Alarm

Combineer beide codes voor een alarm.
Run de code alarm.py op pagina 3.1!

```
from ti_hub import *
flash = led("OUT1")
sound = speaker("BB1")

for i in range(5):
    ♦♦ flash.on()
    ♦♦ sound.tone(500,1)
    ♦♦ sleep(1)
    ♦♦ flash.off()
    ♦♦ sound.tone(375,1)
    ♦♦ sleep(1)
```

Pas de code aan zodat:

- het alarm 10x afgaat,
- ieder signaal 0,5 seconden duurt.

Sirenetoon

Sluit de luidspreker aan op de poort BB1.
Run de code sirene.py op pagina 2.1!

```
from ti_hub import *
sound = speaker("BB1")

for i in range(5):
    ♦♦ sound.tone(500,1)
    ♦♦ sleep(1)
    ♦♦ sound.tone(375,1)
    ♦♦ sleep(1)
```



Temperatuur meten

Sluit de luidspreker aan op de poort BB1.
Run de code temperatuur.py op pagina 4.1!

```
from ti_hub import *
temp = temperature("IN1")

while get_key() != "esc":
    ♦♦ t = temp.measurement()
    ♦♦ print("Temperature =",t)
    ♦♦ sleep(0.5)
```



Magneetdetectie

Voeg een if..else.. Statement toe.
Run de code detectie.py op pagina 6.1!

```
from ti_hub import *
mag = magnetic("IN3")

while get_key() != "esc":
    ♦♦ m = mag.measurement()
    ♦♦ print("Spanning =",int(m))
    ♦♦ if m < 100:
        ♦♦♦♦ print("Magneet aanwezig")
    ♦♦ else:
        ♦♦♦♦ print("Magneet niet aanwezig")
    ♦♦ sleep(0.5)
```

Magneet

Sluit de Hall-sensor* aan op de poort BB1.
Run de code magneet.py op pagina 5.1!

```
from ti_hub import *
mag = magnetic("IN3")

while get_key() != "esc":
    # int(m) = gehele deel m
    ♦♦ m = mag.measurement()
    ♦♦ print("Spanning =",int(m))
    ♦♦ sleep(0.5)
```





Hitte-Alarm

Combineer voorgaande codes voor de simulatie van een hitte-alarm.

Run de code hitte_alarm.py op pagina 7.1!



```

from ti_hub import *
from ti_draw import *
geluid = speaker("BB1")
flash = led("OUT1")
temp = temperature("IN1")
mag = magnetic("IN 3")

def alarm_groen(x):
    ♦♦ draw_text(80,70,"HITTE-ALARM")
    ♦♦ draw_text(80,90,"Temperatuur in auto = {}".format(x))
    ♦♦ set_color(14,107,14)
    ♦♦ draw_text(80,110,"Baby niet in auto")
    ♦♦ draw_text(80,130,"Geen gevaar :-)")

def alarm_oranje(x):
    ♦♦ draw_text(80,70,"HITTE-ALARM")
    ♦♦ draw_text(80,90,"Temperatuur in auto = {}".format(x))
    ♦♦ set_color(255,131,0)
    ♦♦ draw_text(80,110,"Baby in auto")
    ♦♦ draw_text(80,130,"Baby wil mee! :-|")

def alarm_rood(x):
    ♦♦ draw_text(80,70,"HITTE-ALARM")
    ♦♦ draw_text(80,90,"Temperatuur in auto = {}".format(x))
    ♦♦ set_color(255,0,0)
    ♦♦ draw_text(80,110,"Baby in auto")
    ♦♦ draw_text(80,130,"Baby wil mee! :-(!")

use_buffer()

while get_key() != "esc":
    ♦♦ clear();set_color(0,0,0)
    ♦♦ t = temp.measurement()
    ♦♦ m = mag.measurement()
    ♦♦ if m > 100:
    ♦♦♦♦ alarm_groen(t)
    ♦♦ else:
    ♦♦♦♦ if t <= 25:
    ♦♦♦♦♦♦ alarm_oranje(t)
    ♦♦♦♦ else:
    ♦♦♦♦♦♦ alarm_rood(t)
    ♦♦♦♦♦♦ for i in range(1)
    ♦♦♦♦♦♦♦♦ flash.on();sound.tone(500,1)
    ♦♦♦♦♦♦♦♦ sleep(0.5)
    ♦♦♦♦♦♦♦♦ flash.off();sound.tone(375,1)
    ♦♦♦♦♦♦♦♦ sleep(0.5)
    ♦♦♦♦ paint_buffer()

```

