TD Raspberry Pi : Systèmes embarqués
Montage LED, Buzzer et écran LCD

Exercice 1 : Faire clignoter une Led [Correction]

```
from gpiozero import LED
from time import sleep

led = LED(21)

While (True):
led.on()
sleep(1)
led.off()
sleep(1)
```

Exercice 2 : Transposer une phrase en morse [Correction]

```
from gpiozero import LED, Buzzer
from time import sleep

led = LED(4)
buzzer = Buzzer(17)

dicoMorse = {"A":(1,2), "B":(2,1,1,1), "C":(2,1,2,1), "D":(2,1,1), "E":(1,0), "F":(1,1,2,1),
"G":(2,2,1), "H":(1,1,1,1), "I":(1,1), "J":(1,2,2,2), "K":(2,1,2), "L":(1,2,1,1), "M":(2,2),
"N":(2,1), "O":(2,2,2), "P":(1,2,2,1), "Q":(2,2,1,2), "R":(1,2,1), "S":(1,1,1), "T":(2,0),
"U":(1,1,2), "V":(1,1,1,2), "W":(1,2,2), "X":(2,1,1,2), "Y":(2,1,2,2), "Z":(2,2,1,1)}

def emit_signal(letter, current_word):
    for item in dicoMorse[letter]:
        led.on()
        buzzer.on()
        sleep(0.1) if item == 1 else sleep(0.3)
        led.off()
        buzzer.off()
    sleep(0.3)

def main():
    current_word = ""
    text = raw_input("Text : ").upper()

    for letter in text:
        if letter == " ":
            sleep(0.7)
            current_word += " "
            continue
        current_word += letter
        emit_signal(letter, current_word)

if __name__ == "__main__":
    main()
```

Exercice 3 : Afficher une lettre à l'écran [Correction]

```
from gpiozero import LED, Buzzer
from time import sleep
from RPLCD.gpio import CharLCD
from RPi import GPIO

led = LED(4)
buzzer = Buzzer(17)
lcd = CharLCD(pin_rs=15, pin_rw=None,  pin_e=16, pins_data=[21,22,23,24],
numbering_mode=GPIO.BCM, cols=16, rows=2, dotsize=8,charmap='A02', auto_linebreaks=True)

dicoMorse = {"A":(1,2), "B":(2,1,1,1), "C":(2,1,2,1), "D":(2,1,1), "E":(1,0), "F":(1,1,2,1),
"G":(2,2,1), "H":(1,1,1,1), "I":(1,1), "J":(1,2,2,2), "K":(2,1,2), "L":(1,2,1,1), "M":(2,2),
"N":(2,1), "O":(2,2,2), "P":(1,2,2,1), "Q":(2,2,1,2), "R":(1,2,1), "S":(1,1,1), "T":(2,0),
"U":(1,1,2), "V":(1,1,1,2), "W":(1,2,2), "X":(2,1,1,2), "Y":(2,1,2,2), "Z":(2,2,1,1)}

def emit_signal(letter, current_word):
    for item in dicoMorse[letter]:
        lcd.write_string(current_word)
        led.on()
        buzzer.on()
        sleep(0.1) if item == 1 else sleep(0.3)
        led.off()
        buzzer.off()
        lcd.clear()
    sleep(0.3)

def main():
    current_word = ""
    lcd.clear()
    text = raw_input("Text : ").upper()

    for letter in text:
        if letter == " ":
            sleep(0.7)
            current_word += " "
            continue
        current_word += letter
        emit_signal(letter, current_word)

if __name__ == "__main__":
    main()
```