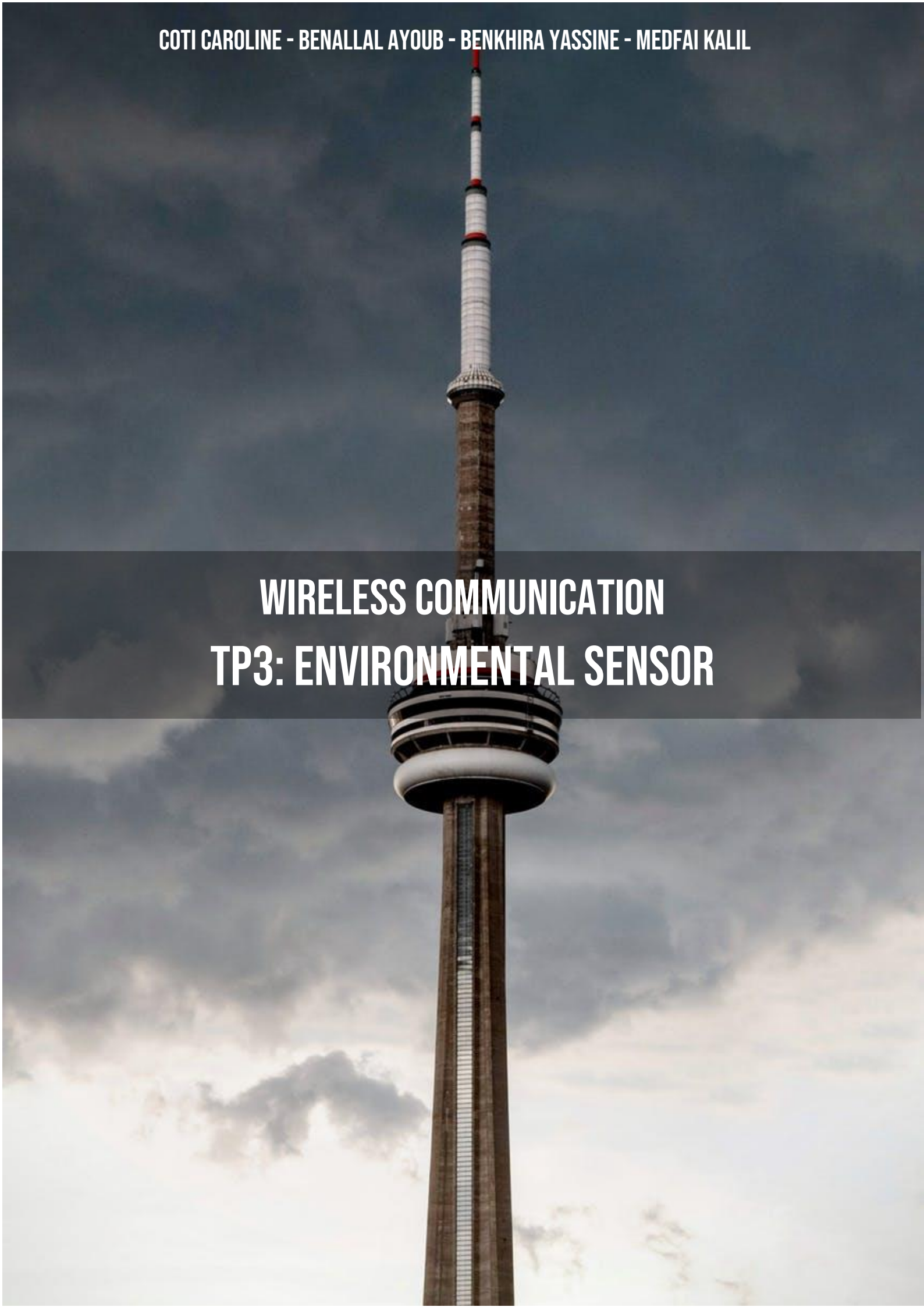
COTI CAROLINE - BENALLAL AYOUB - BENKHIRA YASSINE - MEDFAI KALIL



WIRELESS COMMUNICATION

TP3: ENVIRONMENTAL SENSOR

Table des matières

• INTRODUCTION :

Lors de ce TP nous allons aborder deux types de capteurs, le premier est le capteur « light sensor » et le deuxième est le capteur « Temperature and Humility Sensor ».
Ces deux capteurs ont la particularité d'être deux capteurs environnement.

Le but du TP est d'analyser le code Arduino des deux capteurs séparément puis un programme quand ils sont réunis.

Les professeurs nous ont fourni des capteurs afin de les souder sur la carte Arduino. Ensuite, des programmes ont été utilisé afin de comprendre l'utilisation de ces capteurs

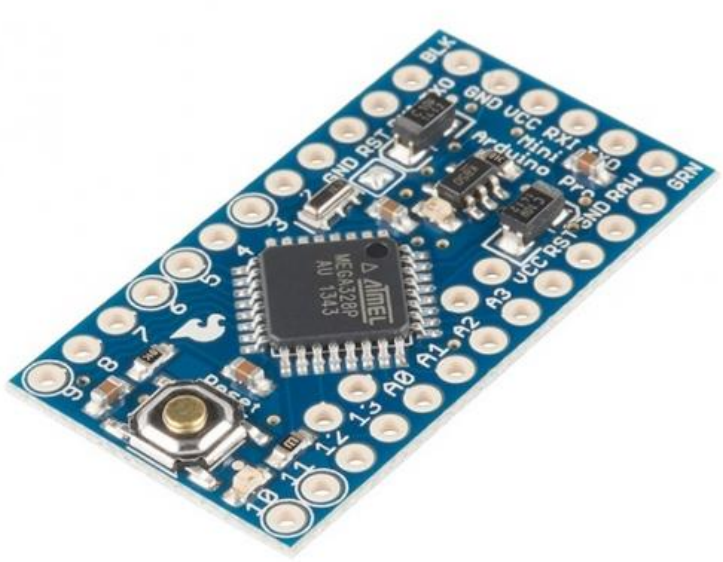


Figure 1: Carte Arduino Pro

• LIGHT SENSOR :

A.LE CAPTEUR:

Nous allons commencer par parler du capteur TEMT6000, permet de mesurer la quantité lumineuse reçue.
La sortie se raccorde sur une entrée analogique d'un microcontrôleur.

Figure 2: Capteur TEMT6000

Voici la fiche technqiue du capteur TEMT6000

Alimentation: 3 à 5 Vcc
Bande passante: 360 à 970 nm
Température de service: -40°C à +85°C
Dimensions: 10 x 10 mm
Référence fabricant: BOB-08688
Photos CC BY-NC-SA 3.0

Figure 3: fiche technique capteur TEMT6000

B.ANALYSE DU PROGRAMME;

Tout d'abort il faut installer la librairie qui est fondamental pour faire marcher le programme

readTEMT6000

```
#include <Wire.h>
```

Figure 4:librairie TEMT6000

Ensuite on définit le « serial begin » a 115200. Il est utilisé pour la transmission de données en série pour communiquer avec l'ordinateur

```
#define LAPIN A0 // PIN with Light sensor analog output
#define LPPIN 4 // PIN with Light power input


static float light;

void setup() {
    Serial.begin(115200);
    pinMode(LPPIN, OUTPUT);

}
```

Figure 5: Serial Begin

Ensuite le programme definit un « Serial.print » qui va affiche la mesure du capteur en Lux.

```
void loop() {

    // temperature is an integer in hundredths
    light = readLight();

    Serial.print("Light :");
    Serial.print(light);
    Serial.println("Lx");

    delay(500);
}
```

Figure 6:print

Au debut le pin d'entrée du capteur de lumière est connecté au pin d'entrée A0

Le capteur retourne une valeur entre 0 et 1023 qui est ensuite convertie plus bas

Pour obtenir une valeur en volts et elle est ensuite convertie en lux (multiplication par 2000)

Et ensuite l'alimentation du capteur est coupée

```
float readLight() {
  float result;
  // Light sensor Voltage
digitalWrite(LPPIN, HIGH); // Power the sensor
delay(10);
int sensorValue = analogRead(LAPIN);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 3.3V):
  float voltage = sensorValue * (3.3 / 1023.0); // Batvalue is 3.3V
result = voltage*2000; // multiply by 2000 to have Lx
digitalWrite(LPPIN, LOW); // switch off the sensor
  return result;
}
```

Figure 7: conversion en lux

• DEUXIÈME PARTIE :

A. LE CAPTEUR:

Après avoir étudié le capteurTEMT6000, nous allons analyser le capteur d'humidité et de température (SI7021 sensor).

Ce capteur permet de relever des mesures d'humidité et de température dans un environnement (température maximum de 150°C). Certains environnements sont déconseillés surtout si de l'eau peut toucher le capteur.

Il a une précision d'humidité de 2% et une précision de température de 0.3°C. C'est un capteur numérique avec une sortie de signal numérique étalonnée, et c'est également un capteur analogique a digital. Il a une interface I2°C pour le transfert de données.

Figure 8: capteur SI7021

B.ANALYSE DU PROGRAMME:

L'utilisation des librairies est essentielle pour faire marcher le capteur avec Arduino (« Wire.h » et « SI7021.h »)

Figure 9: Librairie Arduino

Comme souvent dans les programmes Arduino, un « serial begin » est utilisé pour la transmission de données en série pour communiquer avec l'ordinateur , plusieurs débits sont disponibles, sur ce programme on utilise 115200.

```
void setup() {
    Serial.begin(115200);
    sensor.begin();
}
```

Figure 10: Serial Begin

Tout d'abord le programme commence par définir une valeur pour la température, il utilise le « sensor.getCelciusHundredths » qui permet de lier la température, il le divise ensuite par 100 pour que la température soit un entier en centième.

On utilise aussi un delay qui permet de mettre le programme en pause pendant 0.5 seconde (500 ms).

```
void loop() {

    // temperature is an integer in hundredths
    int temperature = sensor.getCelsiusHundredths();
    temperature = temperature / 100;
    Serial.print("Temperature :");
    Serial.print(temperature);
    Serial.println("°Celcius");

    delay(500);
```

Figure 11: Temperature

Ensuite on a défini aussi une valeur d'humidité comme pour la température mais cette fois ci on a comme unité le pourcentage et on ne divise donc pas par 100, on utilise le « sensor.getHumidityPercent() » pour nous donner une valeur d'humidité en pourcentage, on utilise ensuite un delay pour terminer.

```
// humidity is an integer representing percent
int humidity = sensor.getHumidityPercent();
Serial.print("Humidity :");
Serial.print(humidity);
Serial.println("%");


delay(500);
```

Figure 12: Humidité

Ensuite on teste le capteur avec le « set.heater », si le test est bon on active le heater(« chauffage ») qui va générer de la chaleur si le test est bon.

```
// enable internal heater for testing
sensor.setHeater(true);
  Serial.println("Activate heater");

delay(20000);
sensor.setHeater(false);
Serial.println("Desactivate heater");
```

Figure 13: Set Heater

On utilise alors une condition qui dit que si la température change après le « heater » alors le programme recommence à définir la température et d'en relever une nouvelle et toujours avec la même unité, ensuite un delay de 20 secondes pour faire refroidir le capteur.

```
// see if heater changed temperature
temperature = sensor.getCelsiusHundredths();
temperature = temperature / 100;
Serial.print("Temperature :");
Serial.print(temperature);
Serial.println("°Celcius");


//cool down
delay(20000);
```

Figure 14: Print

Après avoir refroidi le capteur on utilise le « getHumidityAndTemperature » pour obtenir la température et l'humidité en un coup pour économiser de l'énergie, cela est possible car le capteur détecte la température quand il y a de l'humidité.
On affiche ensuite la température en Celsius que l'on divise encore par 100
Et pour terminer on utilise un delay.

```
    // get humidity and temperature in one shot, saves power because sensor takes temperature when doing humidity anyway
    si7021_env data = sensor.getHumidityAndTemperature();
    Serial.print("Temperature :");
    Serial.print(data.celsiusHundredths/100);
    Serial.println("°Celcius");

    delay(500);
}
```

Figure 15: print

Donc le programme permet de récupérer une température et un taux d'humidité grâce au capteur, il faut bien sûr pour pouvoir récupérer une mesure : vérifier, téléverser et afficher le moniteur série.

MULTI-SENSOR:

Pour terminer on a essayé de réunir les deux capteurs en un programme, cela a plus d'avantage que de les séparer, par exemple cela permet d'économiser de l'énergie. Cependant pour éviter les erreurs il est préférable de les séparer.

On réunit un capteur de lumière avec un capteur de température et d'humidité, théoriquement cela est totalement possible car ce sont deux capteurs environnementaux, ils sont compatibles et ils ont presque la même fonction.

Au niveau du programme Arduino cela est possible il faudra juste assembler les deux sans faire d' erreur en mélangeant les conditions ou variables afin d' éviter les erreurs.

Il faudra à la fin du programme faire afficher les 3 mesures différentes (« serial.print »), il faut aussi bien séparer dans l' Arduino pour faire refroidir le capteur. De plus ne pas oublier d' inclure les bibliothèques qui sont essentielles pour faire marcher parfaitement le programme avec les capteurs.

• CONCLUSION :

En conclusion, les deux capteurs séparés peuvent grâce à un programme Arduino mesurer des valeurs (température, humidité, lumière), il est quand même préférable de réunir les programmes en un seul programme cela réduit le cout d'énergie mais les deux programmes séparent sont plus compréhensibles pour un nouvel utilisateur qui découvre les capteur et l'Arduino.