

## Table des matières

1. [Mise en œuvre d'un capteur de température LM35](#)
  - [introduction](#)
  - [calculs préliminaires](#)
  - [programmes](#)
    - [Algorithme d'affichage de la température sur un écran LCD](#)
    - [Montage avec alarme](#)
  - [conclusion](#)
2. [capteur DHT11](#)
  - [introduction](#)
  - [claculs préliminaires](#)
3. [capteur de luminosité](#)
  - [introduction](#)
  - [étude du capteur](#)
  - [Montage capteur luminosité / Arduino / Led](#)
4. [conclusion générale](#)

## Compte Rendu de Travaux Pratiques

### Mise en œuvre d’un capteur de température LM35

#### Introduction

Le travail pratique consiste à mettre en œuvre un capteur de température LM35 avec une carte Arduino. Le LM35 est un capteur analogique, et nous devons effectuer des calculs préliminaires pour déterminer la précision de la mesure en utilisant le convertisseur analogique-numérique (CAN) de l'Arduino.

#### Calculs préliminaires

No.	Paramètre	Valeur
1.	Type du capteur	Analogique
2.	Tension à 25 °C	250 mV
3.	Précision à 25 °C	+ -0.2°C
4.	Quantum de l'Arduino	0.0048828125 V
5.	Résolution en °C	0.4°C
6.	Précision du CAN	0.6°C
7.	Précision finale	+ -0.8°C
8.	Quantum avec référence interne	0.00107421875 V
9.	Précision avec nouvelle référence	0.1°C
10.	Précision finale avec nouvelle ref.	0.35°C

#### Réalisation du programme

#### Algorithme d'affichage de la température sur un écran LCD

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3f,16,2);
const int pinTemp = A0;

void setup() {
    lcd.init();
    lcd.backlight();
    Serial.begin(115200);
    analogReference(INTERNAL);
}

void loop() {
    float sensorValue = analogRead(pinTemp);
    float temp = sensorValue/1024*100;
    lcd.setCursor(0,0);
```

```
lcd.print("Sensor = ");
lcd.print(temp);
Serial.print("Sensor = ");
Serial.println(sensorValue);
delay(1000);
}
```

## Montage avec alarme

Le même programme, avec l'ajout d'un buzzer sur le pin D13 pour une alarme lorsque la température dépasse 30°C.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3f,16,2);

const int pinTemp = A0;
const int pinBuzz = 13;

void setup() {
  lcd.init();
  lcd.backlight();
  Serial.begin(115200);
  analogReference(INTERNAL);
  pinMode(pinBuzz, OUTPUT);
}

void loop() {
  float sensorValue = analogRead(pinTemp);
  float temp = sensorValue/1024*100;
  lcd.setCursor(0,0);
  lcd.print("Sensor = ");
  lcd.print(temp);
  Serial.print("Sensor = ");
  Serial.println(sensorValue);
  if (temp > 30.0) {
    digitalWrite(pinBuzz, HIGH);
  } else {
    digitalWrite(pinBuzz, LOW);
  }
  delay(1000);
}
```

## Conclusion

En conclusion, la précision de la mesure de température dépend de plusieurs paramètres, notamment la résolution du CAN de l'Arduino et la référence utilisée. En utilisant la référence interne, la précision peut être améliorée. Le programme Arduino permet de lire la valeur du capteur LM35 et de l'afficher sur un écran LCD, avec la possibilité d'ajouter une alarme.

## Autre Capteur : DHT11

### Introduction

Cette partie du TP concerne la mise en œuvre du capteur DHT11, qui mesure la température et l'humidité ambiantes.

### Calculs préliminaires

No.	Paramètre	Valeur
1.	Type du capteur	Numérique
2.	Caractéristiques techniques	Voir datasheet
3.	Principe de mesure température	Thermistance
4.	Principe de mesure humidité	Condensateur variable
5.	Schéma de définition de la trame datas	À vérifier
6.	Précision	±1°C pour la température, ±4% pour l'humidité relative

## Autre Capteur : Luminosité

### Introduction

Cette partie concerne la mise en œuvre d'un capteur de luminosité pour mesurer le niveau de luminosité ambiante.

### Étude du capteur

No.	Caractéristique	Valeur
1.	Type du capteur	Analogique
2.	Tension de fonctionnement	3~5V
3.	Courant de fonctionnement	0.5~3 mA
4.	Temps de réponse	20-30 ms
5.	Longueur d'onde de crête	540 nm
6.	Poids	4 g

## Montage capteur luminosité / Arduino / Led

### Matériel :

- Capteur de luminosité
- Arduino UNO
- Résistance de 220 Ohm
- LED

1. **Code Arduino** : À recopier à partir de la vidéo.
2. **Fonctionnement du programme** : Mesure la valeur du capteur, allume ou éteint une LED en fonction de la luminosité, envoie la valeur sur le port série.
3. **Seuil de luminosité** : Inférieur à 200.
4. **Conclusion sur le fonctionnement du montage** : Utilisation d'une photoresistance, conversion analogique-numérique, et allumage de la LED en fonction de la luminosité.

## conclusion générale

En conclusion, ce travail pratique a permis de mettre en œuvre différents capteurs et de comprendre leur fonctionnement, ainsi que l'importance des calculs préliminaires pour garantir la précision des mesures. L'utilisation d'Arduino facilite la mise en place de systèmes de mesure et de contrôle.