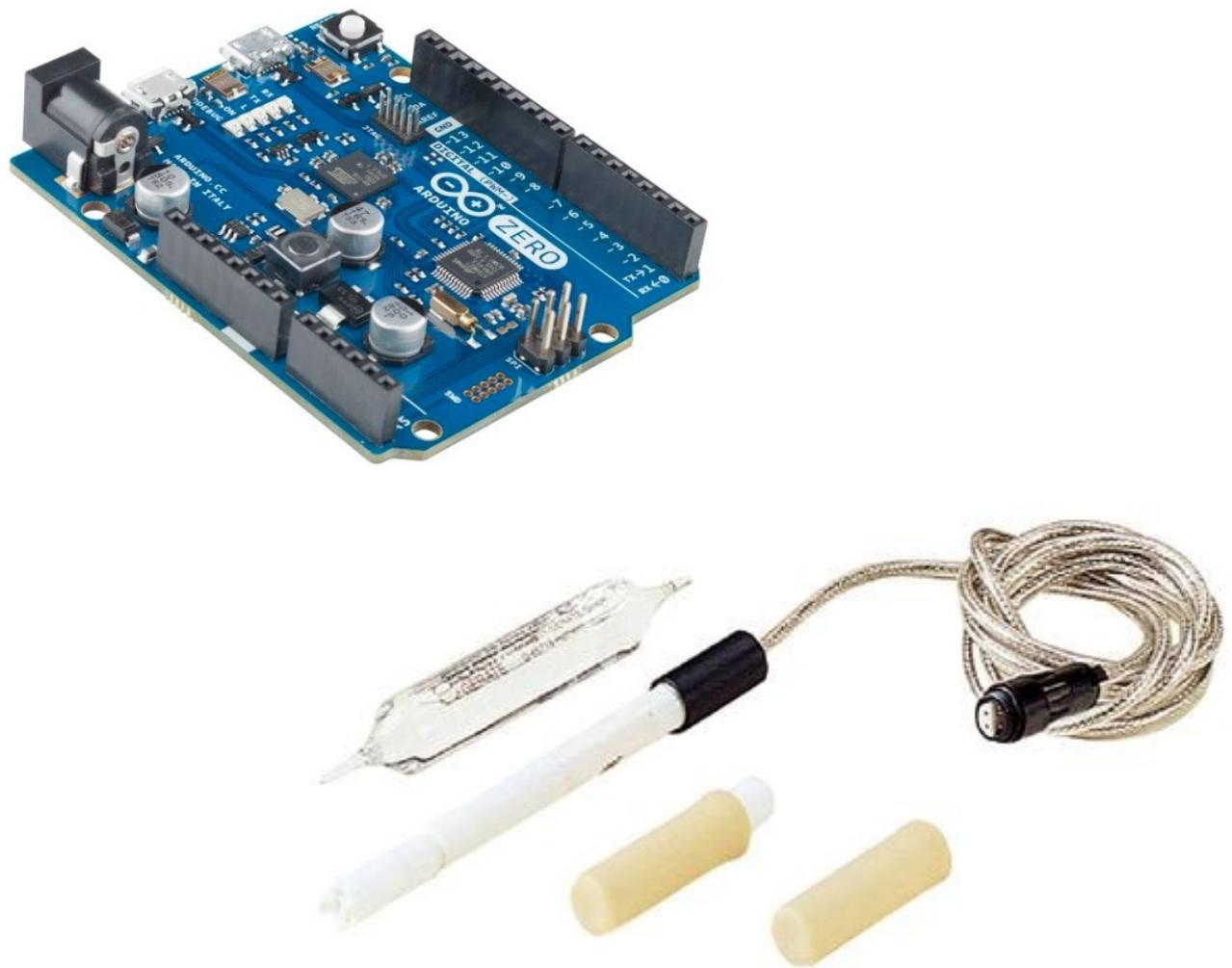


Mesure d'une concentration en dioxygène à l'aide d'une carte Arduino reliée à une sonde de Clark



DOMENJOD Quentin
GAMIN Cécile
TS1
Projet ISN 2014/2015
Lycée Antoine Roussin

SOMMAIRE

Présentation.....	p3
Analyse du besoin-Recherches d'idées.....	p4
Répartition des tâches-Démarche collaborative	p5
Réalisation	p6
1)Démarche	
2)Difficultés	
Intégration-Validation	p8
Bilan et Perspectives	p9
1)Bilan	
2)Perspectives	
Bibliographie	p10
Annexes	p11

Présentation

Ce projet a pour but premier de mesurer à partir d'une carte Arduino qui sera reliée à une sonde de Clark, une concentration en dioxygène. Avant il va nous falloir mesurer deux valeurs de potentiel qui correspondent à des concentrations connues, on les numérise ensuite.

À partir de ces valeurs on devra construire une fonction linéaire qui va permettre de convertir une tension mesurée en une concentration. Ces valeurs pourront être réutilisées. Ces valeurs correspondent à une tension minimale correspondant à un pourcentage de 0 % de dioxygène et une tension maximale correspondant à 21 % de dioxygène. Pour permettre cela, l'utilisation d'un tableau de mesures est envisageable. Il faut de plus savoir faire une correction de température avec les couples d'étalonnage. À partir de tous ces différentes étapes, il faut programmer un enregistrement pour permettre de relever des mesures à intervalles de temps réguliers et lorsque la variation de concentration dépasse un seuil fixé.

Ce projet permet d'apprendre à étalonner un capteur, à faire des programmes avec le logiciel Arduino et permet d'apprendre les langages. Ce projet mêle informatique et physique.

Analyse du besoin

Recherches d'idées

Cahiers de charges

Le but visé par notre projet est :

1-de mesurer une concentration en dioxygène à partir d'une sonde de Clark reliée à un carte Arduino. Pour cela on enregistre de valeurs de potentiels correspondant à des concentrations connues dans notre cas 0 % et 21 % de dioxygène,

2-de programmer en fonction des enregistrements des différentes mesures.

Les différents objectifs à réaliser sont de faire des mesures avec la sonde, connaître comment faire une fonction linéaire et maîtriser le logiciel Arduino de même que son langage pour faire des programmes.

Pour mettre au point ces différents objectifs, on à notre disposition une carte Arduino, et la sonde de Clark.

Pour réaliser ce projet, on doit s'appuyer sur les exemples de programmes déjà présents sur le logiciel Arduino, sur des fichiers qui nous expliquent comment utiliser la carte Arduino.

Liste du matériel

- la carte Arduino Uno
- le logiciel Arduino
- des fils électriques
- une plaque labdec ou breadboard
- une résistance de 10 **K**
- un potentiomètre
- la sonde de Clark

Répartition des tâches

Démarche collaborative

Pour ce projet, on a dû partager les tâches et bien s'organiser. Chacun d'entre nous eu des tâches bien précises. Tout d'abord pour pouvoir comprendre le langage Arduino, on a dû lire une multitude des fichiers PDF ainsi qu'un blog très détaillé qui expliquait comment réaliser les programmes demandés, quels étaient leur but et comment les corriger. D'autres sites internet nous ont permis de comprendre la signification de chaque constante et variable du code.

Nous nous sommes beaucoup entraïdés afin de régler les difficultés que nous rencontrions.

Pour ma part j'avais le rôle du bêta-testeur, je me suis donc occupé à tester les programmes et de les corriger si il y avait un problème. Malgré tout j'ai réalisé aussi les programmes que ma partenaire n'arrivait pas faire,nous avons partagés les différents programmes à faire et j'ai aussi travaillé avec mon professeur sur le montage de la carte arduino et le potentiomètre et de la sonde de clark. Le travail de ma partenaire consistait à réaliser les programmes.Afin de mieux cibler notre travaux nous avons fait un tableau récapitulatif de nos taches:

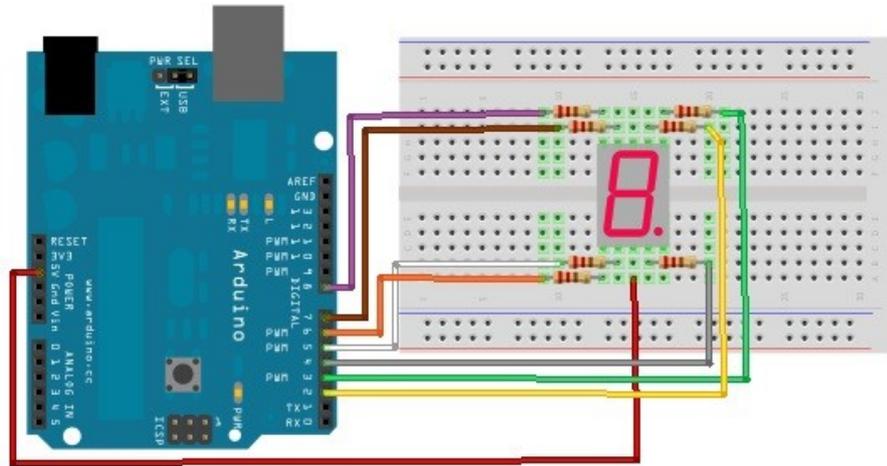
Tâches	Noms
Programme calibration,mesures	Cécile
Programme calcule des valeurs et de leur correspondance en %	Quentin
Mettre les commentaires des programmes	Cécile
Tester les programmes et les corrigés	Quentin
Mise en page du dossier	Cécile,Quentin

Réalisation

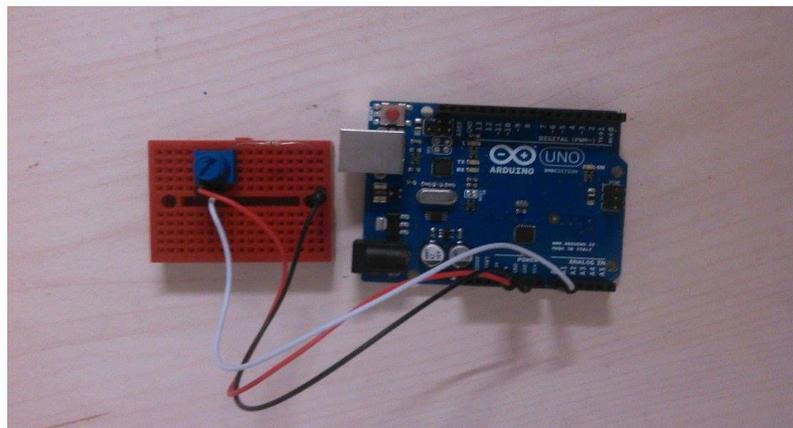
Démarche



Nous avons eu une première approche de la carte Arduino avec le côté électronique et le côté programmation. Nous avons tout d'abord eu recours à l'afficheur 7 segments.



La seconde étape a été d'utiliser un potentiomètre avec la carte Arduino. Le potentiomètre peut ainsi prendre une valeur minimale et une valeur maximale donc ça m'a aidé à comprendre comment commencer et comprendre le sujet avant d'utiliser la sonde. Pour le programme des mesures j'ai dû apprendre les différentes variables qui composent les programmes d'Arduino afin de pouvoir comprendre ce que voulait dire le programme et ainsi mieux cibler les bugs. Avec le potentiomètre, il a fallu savoir étalonner et calibrer le capteur.



Le montage avec le potentiomètre ne change rien à l'algorithme avec la sonde réelle car il permet de mesurer une tension minimale qui correspond à un pourcentage de 0 % de dioxygène et une tension maximale qui correspond elle à 21 % de dioxygène. Pour le code on s'est appuyé sur les codes déjà présents sur le logiciel. Pour le programme calibration, on a eu seulement à ajouter des fonctions, des commandes pour l'ajuster à notre projet. On a aussi travaillé sur la petite partie mathématique du projet puisqu'il fallait trouver une droite d'équation permettant de trouver les valeurs en tension qui correspondait au pourcentage de dioxygène. Pour la droite d'équation $y=ax+b$, on s'est aidé d'un algorithme et d'un programme qui permettait de calculer une équation de droite en utilisant les fonctions avec Python vu en début d'année.

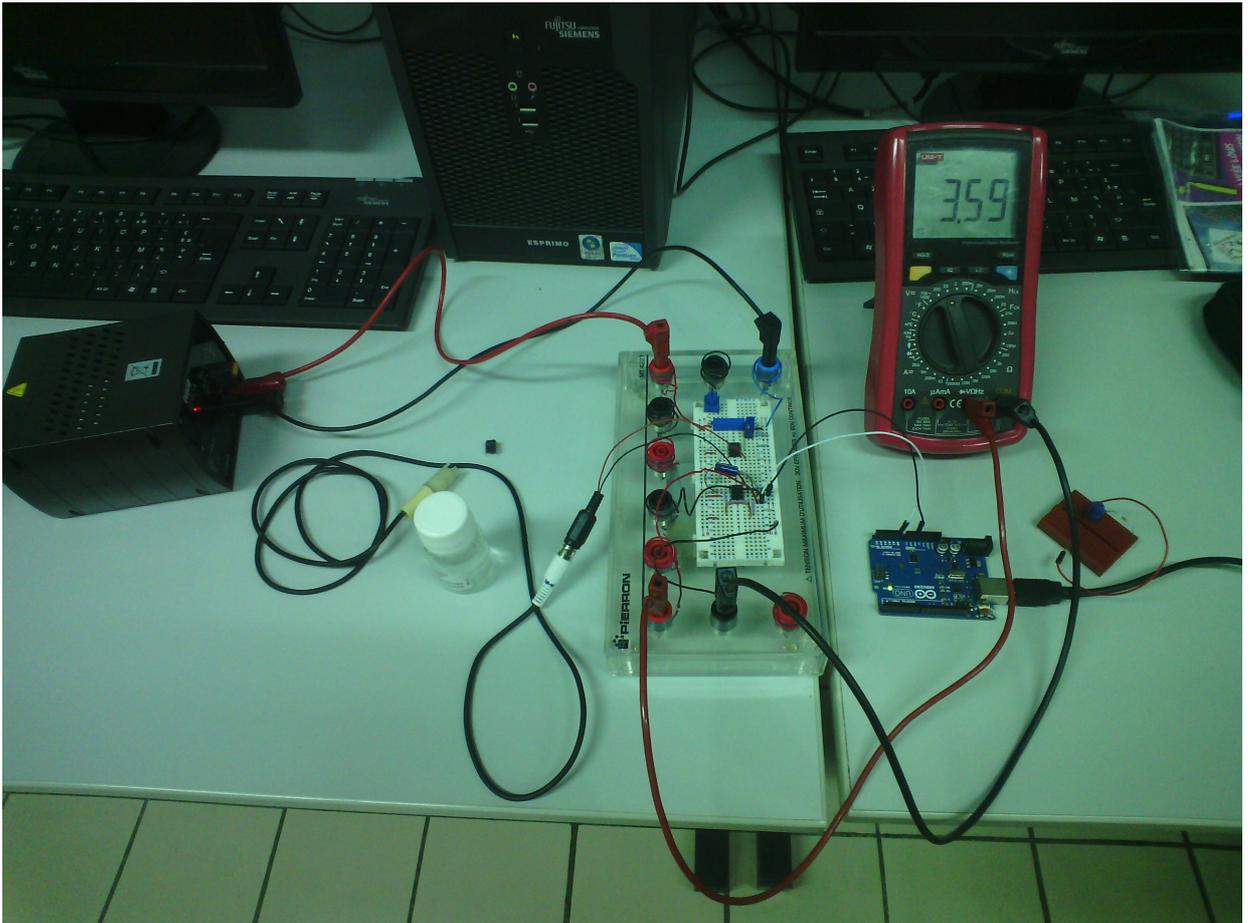
Difficultés

Pendant la durée du projet, on a été confronté à des difficultés, j'ai eu beaucoup de mal à comprendre comment faire le montage d'Arduino avec le potentiomètre. On a eu assez de problème à comprendre la syntaxe du langage Arduino, il y avait beaucoup de variables et il fallait comprendre comment les utiliser et savoir les placer dans les programmes. Pour transformer les valeurs en tension ça été aussi dur à faire. De plus le montage avec la sonde de Clark a posé beaucoup de problèmes. Je trouve que ce projet comportait énormément de physique et qu'il n'y avait pas assez d'exemples sur lesquels s'appuyer pour s'aider, la partie informatique est restreinte au profit de la physique. En outre pour les codes quand il y avaient la moindre erreur il fallait reprendre tout depuis le début pour discerner où se situait la faute. De part ces maladroites on s'est rendu compte qu'il fallait être minutieux dans les codes.

Il a été difficile aussi de modifier le code pour passer de la phase calibration à la phase de mesures.

Intégration

Validation



La réalisation du projet avec le potentiomètre a favorisé la réalisation finale avec la sonde réelle. Cela n'a eu aucun changement sur nos programmes et sur notre sujet de départ.

Grâce aux codes réalisés on a pu faire les programmes demandés c'est à dire enregistrer des mesures, un programme pour la phase de la calibration.

Bilan et Perspectives

Bilan

Ce projet a été un bon moyen de mêler physique, informatique et un peu de mathématique. Ce projet peut être amélioré et même devrait moins comporter des notions de physique. La réalisation du projet fut long et difficile mais cela nous a permis de renforcer nos bases en informatique ainsi qu'en physique. De plus être en groupe de deux plutôt qu'à trois a permis de réellement répartir les tâches à faire et ainsi être plus performant lors des travaux réalisés en classe.

Perspectives

Ce projet a permis de travailler notre esprit de travail en groupe. Il nous a permis aussi d'avoir plus de connaissances en informatique, surtout sur les programmes et le logiciel Arduino qu'on ne connaissait pas. Ce projet nous a aidé aussi à travailler sur notre patience car ça été très dur de faire et comprendre les programmes. Ça nous a motivé de rendre ce projet dans les temps avec on a eu beaucoup de difficulté avec faire fonctionner la sonde de Clark avec la carte Arduino. D'ailleurs ça nous a donné l'occasion de voir que pour un code, il ne faut pas faire la moindre erreur sinon il ne marche pas. Ici on pourrait dire que le code cherche la perfection.

Bibliographie

<http://eskimon.fr/ebook-tutoriel-arduino>

<http://eskimon.fr/1583-arduino-502-differents-types-de-mesures#777662>

<http://www.arduino.cc/en/Tutorial/HomePage>

Arduino pour bien commencer en électronique et en programmation

Le logiciel Fritzing pour le montage de la carte Arduino avec l'afficheur 7 segments

<http://arduino-from-scratch.over-blog.org/article-la-led-clignotante-88541471.html>

http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.Float

Annexes

```
/*
  Calibration
  Le programme permet donc de calibrer le potentiomètre. Avant de commencer, il faut déclarer les constantes et les variables. Grâce à la fonction setup() on initialise
  les variables. Dans cette partie, on calibre pendant les cinq premières secondes.
  Après viens la fonction loop() qui permet l'exécution et la réponse dans un boucle, du programme.
*/

// Ces constantes ne changeront pas:
// On affecte là la variable "sensorPin" une valeur numérique proportionnelle à la tension mesurée sur l'entrée analogique A0.
const int sensorPin = A0; // La broche du capteur est branché sur A0
const int ledPin = 9; // La broche de la led est fixé à 9
float tension = 0.0;
int valeur = 0;

// Variables:
int sensorValue = 0; // La valeur du capteur
int Min = 0; // Valeur minimum du capteur
int Max = 1023; // Valeur maximale du capteur

// La fonction setup() ets la partie d'initialisation et de configuration des entrées et sorties
void setup() {
  /*On allume la LED pour signaler le début de la période d'étalonnage:
  On distingue cette phase à la phase de mesures.
  pinMode(), c'est la configuration des broches numériques en entrée ou en sortie.
  */
  pinMode(13, OUTPUT); // OUTPUT signifie que la connexion est utilisée en sortie
  digitalWrite(13, HIGH);
  /* digitalWrite (), pour la broche numérique configurée en sortie, cette instruction permet de mettre son niveau logique à HAUT ou BAS
  HIGH : niveau haut 5V ce qui permet de faire circuler un courant dans la LED dont l'autre borne est reliée à la masse 0V
  */
  Serial.begin(9600);
  // Fixe le débit de communication en nombre de caractères par seconde (l'unité est le baud) pour la communication en série

  /* On calibre au cours des cinq premières secondes:
  analogRead() : lecture de la valeur de la tension présente sur une entrée analogique (A à 5V sur la Uno). La fonction retourne la valeur issue de la Conversion
  Analogique Numérique, soit une valeur comprise entre 0 et 1023 (convertisseur 10 bits)
  */
  while (millis() < 5000) {
    sensorValue = analogRead(sensorPin);

    //On enregistre la valeur maximale du capteur:
    if (sensorValue > Max) {
      Max = sensorValue;
    }

    //On enregistre la valeur minimale du capteur:
    if (sensorValue < Min) {
      Min = sensorValue;
    }
  }

  //On signale la fin de la période d'étalonnage:
  digitalWrite(13, LOW);
}

// La fonction loop() est la partie principale qui s'exécute en boucle
void loop() {

  //On lit la valeur du capteur:
  sensorValue = analogRead(sensorPin);

  //On applique la calibration à la lecture du capteur:
  // map() : ré-étalonne un nombre d'une fourchette de valeur vers une autre fourchette
  sensorValue = map(sensorValue, Min, Max, 0, 255);
}
```

```

//Dans le cas où la valeur du capteur est hors de la plage vu lors de l'étalonnage:
// constrain() : contraint un nombre à rester dans une fourchette précise
sensorValue = constrain(sensorValue, 0, 255);

// Estomper la LED en utilisant la valeur calibrée:
analogWrite(ledPin, sensorValue);
Serial.println(sensorValue);
// Serial.println(b) affiche b en tant que nombre décimal sous la forme d'une chaîne de caractères ASCII suivi d'un retour de chariot et d'un saut à la ligne
delay(1000);
// delay() : réalise une pause dans l'exécution du programme pour la durée (en millisecondes) indiquée en paramètre
}

// Permet d'afficher la valeur lue par la carte en une tension
void afficheValeurTension() {

    sensorValue = analogRead(sensorPin);
    tension = valeur * 5.0 / 1023; // Multiplie la valeur par la tension , ici 5V et la divise par 1023 la valeur CAN

    // On affiche la valeur lue sur la liaison de sortie
    Serial.print("valeur = ");
    Serial.println(valeur);

    // On affiche la tension calculée
    Serial.print("Tension = ");
    Serial.print(tension,2);
    Serial.println("V");

    Serial.println(); // On saute une ligne entre les deux affichages
    delay(500); // On attends une demi-seconde pour que l'affichage ne soit pas trop rapide
}

void mesure() {

    sensorValue = analogRead(sensorPin);
    if (sensorValue>Max){
        Max=sensorValue;
    }
    delay(5000);

    if (sensorValue<Min){
        Min=sensorValue;
    }
    delay(5000);
    Serial.print("sensor = ");
    Serial.print(sensorValue);

    /* Permet de calculer une équation d'une droite du type y=ax+b connaissant les valeurs x et y des deux points
    int : un int (pour integer) est un entier relatif codé sur 2 octets, ce qui signifie qu'il possède 2^16 valeurs différentes. Il peut prendre des valeurs comprises dans
    l'intervalle [-32,768;32,767]
    float : déclare des variables de type "virgule-flottante", c'est à dire des nombres à virgules. Les nombres à virgules sont souvent utilisés pour l'expression des
    valeurs analogiques et continues, parce qu'ils ont une meilleure résolution que les nombres entiers. Il est codé sur 4 octets. Sa précision n'excède pas 6 ou 7 décimales.
    */
    int Min = 0;
    int Miny = 0;
    int Max = 1023;
    int Maxy = 21;
    float a;
    float b;

    a = (Max - Min) / (Maxy - Miny);
    b = Max - ((Max - Min) / (Maxy - Miny)) * Min;
}

```

