



CONSEIL ONTARIEN  
POUR L'ÉDUCATION  
TECHNOLOGIQUE

## ÉCHANGEUR D'AIR AUTOMATISÉ

---

Technologie des systèmes informatiques

TEJ3/4M

Juin 2020

A stylized grey hand is shown pointing towards the right, with several concentric grey circles around the index finger, suggesting a digital interaction or a cursor.

**RESSOURCE  
EN LIGNE**



# Table des matières

---

Introduction .....	3
Aperçu du projet.....	3
Connaissances préalables .....	3
Activités d'apprentissage.....	3
Activité 1 – Signaux numériques et analogiques en Arduino (entrées et sorties) .....	3
Activité 2 – DEL RGB (Sortie analogue) .....	5
Activité 3 – Capteur de température (Entrée analogue).....	6
Activité 4 – Micro servomoteur (avec bibliothèques d'Arduino) .....	7
Activité 5 – Contrôle d'un moteur DC.....	7
Activité 6 – Projet d'échangeur d'air automatisé .....	8
Stratégies pédagogiques et de motivation .....	9
Attentes et contenus d'apprentissage .....	9
Différenciation pédagogique.....	10
Évaluation .....	10
Réflexion .....	10
Annexe A – Instructions pour utiliser des classes sur Tinkercad.....	11
Activité 1 – Signaux numériques et analogues en Arduino (entrées et sorties).....	14
Activité 1a – Activités de programmation pour l'Arduino .....	16
Activité 1b – Tinkercad, brochez et programmez un bouton .....	18
Activité 1c – Potentiomètre pour abaisser un DEL - Entrée et sortie analogue .....	20
Activité 2 – DEL RGB (Sortie analogue).....	22
Activité 2 – Vidéo RGB DEL avec Tinkercad.....	23
Activité 3 – Capteur de température (Entrée analogue) .....	24
Activité 3 – DEL RGB à température contrôlée .....	26
Activité 4 – Micro servomoteur (avec bibliothèques d'Arduino).....	29
Activité 5 – Contrôle d'un moteur CC(DC) Arduino .....	31
Activité 5 – Contrôle d'un moteur DC - Arduino L293D Motor Driver .....	35
Activité 6 – Projet d'échangeur d'air automatisé.....	39
Grille d'évaluation pour circuits:.....	40
Journal de solution des circuits :.....	40
Réflexion du projet:.....	41
Références.....	43

# Introduction

**Code de cours:** TEJ3/4M

**Technologie à portée générale:** Technologie des systèmes informatiques

**Destination:** préuniversitaire / précollégial

**Niveau:** 11 et 12

**Prérequis:** Aucun

**Nom du projet:** Échangeur d'air automatisé

## Aperçu du projet

Le but de ce projet est de créer un système automatisé d'un échangeur d'air qui déplace l'air entre l'intérieur et l'extérieur, dans deux pièces d'une maison ou dans deux serres pour aider à régler la température. Ce projet peut être réalisé en classe (liste des pièces à fournir) ou en ligne avec Tinkercad (le simulateur Web d'Autodesk). Les élèves utiliseront un microcontrôleur Arduino, un moteur à courant continu (ventilateur), une puce de commande de moteur L293D, des DEL RGB, un moteur servo et deux capteurs de température analogiques.

## Connaissances préalables

Ce projet ne suppose aucune connaissance préalable, mais il y aura quelques concepts de base en électronique et en électricité qu'ils auront peut-être appris en sciences en 9e année.

## Activités d'apprentissage

### Activité 1 – Signaux numériques et analogiques en Arduino (entrées et sorties)

L'objectif de cette activité est de présenter aux élèves le microcontrôleur Arduino et la façon dont il va produire et recevoir des signaux par l'intermédiaire des broches d'entrée et de sortie. L'enseignant doit présenter l'interface Arduino et faire l'exemple de la "DEL clignotante" avec les élèves sur Tinkercad.com. Je suggère de faire tous ces exemples sur Tinkercad d'abord, ensuite passer à la plaquette de prototypage réel et aux kits Arduino. Cela permet aux élèves d'essayer ces circuits avec un retour d'information dans le simulateur en ligne. Également fourni d'autres ressources pour aider les enseignants et les élèves à programmer de façon élémentaire et à démarrer avec Arduino sur Tinkercad.

Temps requis: 90 minutes

Matériel/équipement requis si vous êtes en classe : Kit Arduino avec plaque de prototypage, résistances (220 Ohm et 10K Ohm), DELs, boutons et potentiomètres.

Ressources: [Les documents de cette activité.](#)

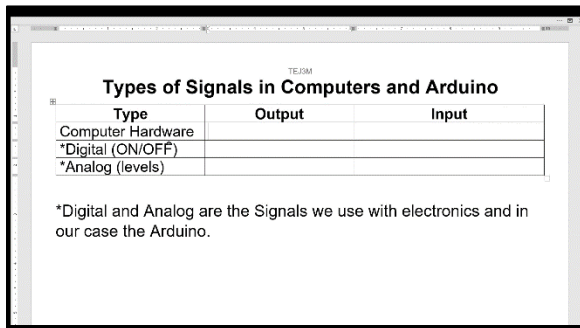
- Introduction à l'interfaçage avec un Arduino (Présentation pour enseignant)
- Arduino « Cheat Sheet » (Feuille)
- Types d'entrées et de sorties (Note pour enseignant)
- Activité de programmation pour Arduino (Fiche de travail)
- Configuration de Tinkercad pour le codage (Schéma pratique)
- Activité bouton avec Arduino – Schéma de branchage et codage (Activité)
- Activité potentiomètre avec Arduino – Schéma de branchage et codage (Activité)
- Analogique vs numérique / entrée vs sortie (Activité vidéo)
- Introduction à Arduino avec Tinkercad (Activité vidéo)
- Potentiomètre à DIM DEL – Entrée et sortie analogique (Activité vidéo)
- [Lien au site web d'OCTE](#)

Lien Web: Présentation PowerPoint – [Introduction à l'interfaçage avec un Arduino](#)



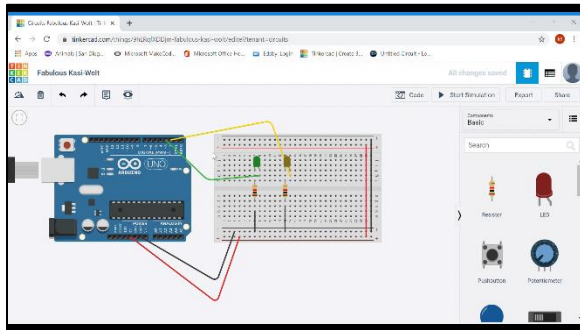
[https://www.octe.ca/download\\_file/5757/1685](https://www.octe.ca/download_file/5757/1685)

Lien Web : Vidéo – [Analogique vs numérique / entrée vs sortie](#)



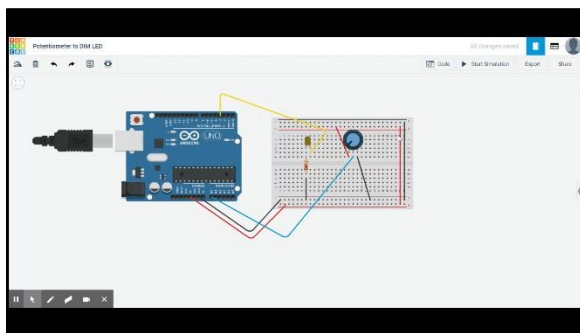
[https://www.octe.ca/application/files/2415/9693/0525/Video\\_1a.\\_Analog\\_vs\\_Digital\\_and\\_Input\\_vs\\_Output.mp4](https://www.octe.ca/application/files/2415/9693/0525/Video_1a._Analog_vs_Digital_and_Input_vs_Output.mp4)

Lien Web : Vidéo – [Introduction à Arduino avec Tinkercad](#)



[https://www.octe.ca/application/files/9015/9708/8567/Video\\_1b\\_Basic\\_Intro\\_to\\_Arduino\\_in\\_Tinkercad.mp4](https://www.octe.ca/application/files/9015/9708/8567/Video_1b_Basic_Intro_to_Arduino_in_Tinkercad.mp4)

Lien Web : Vidéo – [Potentiomètre pour abaisser une DEL – Entrée et sortie analogue](#)



[https://www.octe.ca/application/files/3815/9693/1715/Video\\_1c\\_Potentiometer\\_to\\_DIM\\_LED\\_Analog\\_Input\\_and\\_Output.mp4](https://www.octe.ca/application/files/3815/9693/1715/Video_1c_Potentiometer_to_DIM_LED_Analog_Input_and_Output.mp4)

## Activité 2 – DEL RGB (Sortie analogue)

L'objectif de cette activité est d'explorer la DEL RGB à cathode commune et de mieux comprendre comment l'Arduino peut produire et contrôler des signaux numériques et analogiques. Les élèves pourront produire des couleurs personnalisées avec leur DEL RGB en utilisant des signaux analogiques.

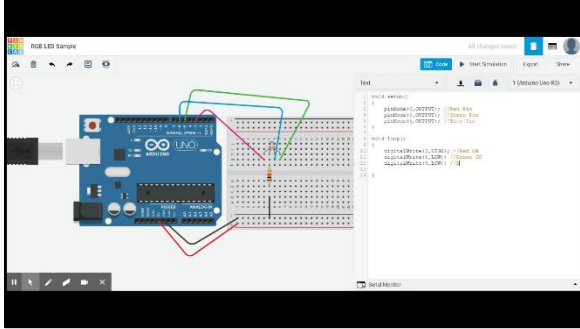
Temps requis: 30 minutes

Matériel/équipement requis si vous êtes en classe : Kit Arduino avec plaque de prototypage, résistances (220 Ohm et 10K Ohm), et une DEL RGB à cathode commune

Ressources : [Les documents de cette activité.](#)

- Activité DEL RGB avec Arduino – Schéma de câblage et code
- Activité DEL RGB avec Tinkercad (Video)

Lien Web: [RGB LED Video in Tinkercad](#)



[https://www.octe.ca/application/files/3015/9693/3502/Video\\_2.\\_RGB\\_LED\\_Video.mp4](https://www.octe.ca/application/files/3015/9693/3502/Video_2._RGB_LED_Video.mp4)

Évaluation : Les élèves doivent soumettre leurs circuits à l'enseignant pour obtenir un retour d'information.

### Activité 3 – Capteur de température (Entrée analogue)

L'objectif de cette activité est d'explorer le capteur de température TMP36 et de mieux comprendre comment l'Arduino reçoit ou lit les entrées analogiques. Les élèves réaliseront un travail sur un circuit qui utilise une DEL RGB pour représenter la température dans une pièce.

Temps requis: 45 minutes

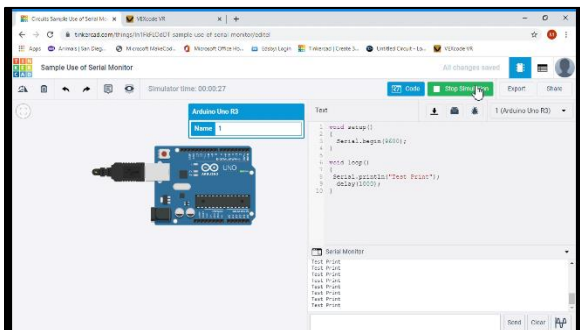
Matériel/équipement requis si vous êtes en classe : Kit Arduino avec plaque de prototypage, résistances (220 Ohm), un capteur de température TMP36 analogue et une DEL RGB à cathode commune

Ressources : [Les documents de cette activité.](#)

Activité Arduino d'un capteur de température – Câblage et codage

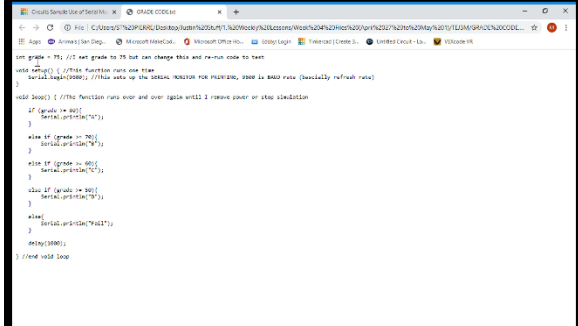
- Activité de câblage et de codage
- Contrôler une DEL RGB avec la température

Lien Web : Vidéo – [L'utilisation du « Serial Monitor » Tinkercad](#)



[https://www.octe.ca/application/files/7415/9708/8223/Video\\_3a\\_Use\\_of\\_Serial\\_Monitor\\_in\\_Tinkercad.mp4](https://www.octe.ca/application/files/7415/9708/8223/Video_3a_Use_of_Serial_Monitor_in_Tinkercad.mp4)

## Lien Web : Vidéo – [L'utilisation de « If Statements » dans Tinkercad](https://www.octe.ca/application/files/8215/9708/8065/Video_3b_Using_If_Statements_in_Tinkercad.mp4)



```
void setup() { //The function runs one time
  Serial.begin(9600); //This sets up the serial monitor for printing, use a 9600 rate (basically refresh rate)
}

void loop() { //The function runs over and over again until I receive power or stop simulation
  if (pinA == 0) {
    Serial.println("0");
  }
  else if (pinA == 70) {
    Serial.println("70");
  }
  else if (pinA == 60) {
    Serial.println("60");
  }
  else if (pinA == 80) {
    Serial.println("80");
  }
  else {
    Serial.println("null");
  }
  delay(500);
} //end void loop
```

[https://www.octe.ca/application/files/8215/9708/8065/Video\\_3b\\_Using\\_If\\_Statements\\_in\\_Tinkercad.mp4](https://www.octe.ca/application/files/8215/9708/8065/Video_3b_Using_If_Statements_in_Tinkercad.mp4)

Évaluation: Les élèves soumettront le laboratoire DEL RGB à température contrôlée pour évaluation.

### Activité 4 – Micro servomoteur (avec bibliothèques d'Arduino)

L'objectif de cette activité est de présenter les micro-servomoteur et la manière d'utiliser les bibliothèques d'Arduino. Les élèves construiront le circuit de "balayage" des servos puis ils utiliseront l'exemple du potentiomètre de l'activité n°1 pour contrôler un servo. La note clé pour eux est la fonction MAP qui, dans ce cas, devra faire correspondre la valeur du potentiomètre (0 à 1023) à la plage du servo (0° à 180°).

Temps requis: 45 minutes

Matériel/équipement requis si vous êtes en classe : Kit Arduino avec plaque de prototypage, micro servo moteur (MG90), et un potentiomètre.

Ressources : [Les documents de cette activité.](#)

- Arduino activité pour servo - Schéma de câblage et code

Évaluation: Les étudiants doivent soumettre leurs circuits aux enseignants pour obtenir un retour d'information.

### Activité 5 – Contrôle d'un moteur DC

L'objectif de cette activité est d'explorer les moteurs à courant continu et leur contrôle. Nous explorerons d'abord une commande directionnelle avec une diode, puis nous présenterons la puce L293D pour le bidirectionnel. Nous contrôlerons ces moteurs à l'aide de signaux numériques et analogiques.

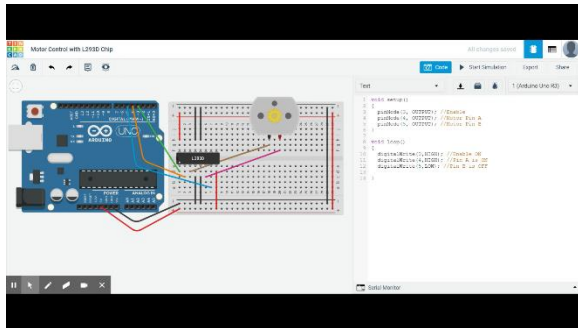
Temps requis: 60 minutes

Matériel/équipement requis si vous êtes en classe : Kit Arduino avec plaque de prototypage, une résistance ( $\approx 2k$  Ohm), moteurs DC avec fils, Transistor PNP (N2222A), et une puce L293D pour contrôler un moteur.

Ressources : [Les documents de cette activité.](#)

- Activité d'un moteur DC avec Arduino – Schéma de câblage et code
- Activité de contrôler un moteur avec la puce L293D– Schéma de câblage et code
- Questionnaire sur Arduino

Lien Web : Vidéo – [Bi-directional Motor Control with L293D](#)



[https://www.octe.ca/application/files/7015/9692/6223/Video\\_5.\\_DC\\_Motor\\_Control\\_with\\_L293D.mp4](https://www.octe.ca/application/files/7015/9692/6223/Video_5._DC_Motor_Control_with_L293D.mp4)

Évaluation : Les élèves doivent soumettre leurs circuits à l'enseignant pour obtenir un retour d'information. Les élèves devront répondre à un questionnaire sur Arduino.

## Activité 6 – Projet d'échangeur d'air automatisé

Le but de cette activité est de combiner l'apprentissage des activités précédentes pour réaliser un système d'échangeur d'air automatisé en utilisant deux capteurs de température, des DELs RGB, un servo et un moteur DC.

Temps requis:60 minutes

Matériel/équipement requis si vous êtes en classe : Kit Arduino avec plaque de prototypage, résistances (220 Ohm), un TMP36, un moteur DC avec fils, (ou un vieux ventilateur 5v de PC), un micro servo, une DEL RGB et un contrôleur à moteur L293D.

Ressources : [Les documents de cette activité.](#)

- Projet d'échangeur d'air automatisé

Évaluation : Les élèves termineront le projet et soumettront leur circuit complété pour évaluation. L'enseignant utilisera une grille d'évaluation et une réflexion pour l'évaluation.

## Stratégies pédagogiques

Les élèves peuvent réaliser toutes les activités de leur domicile avec l'aide de leur enseignant. Il existe une variété de leçons, d'activités exploratoires et de tâches qui peuvent être réalisées du domicile à l'aide d'un simulateur en ligne. Toutes les activités (leçons, devoirs, questionnaires, etc.) peuvent être facilement réalisées et partagées grâce à un système d'apprentissage en ligne. Les étudiants peuvent également soumettre en ligne tous les travaux effectués sur le simulateur en "partageant" une image de leur circuit. Des instructions pour les enseignants ont été fournies pour la mise en place d'une classe sur le site web Tinkercad. Les enseignants peuvent voir en direct les circuits des élèves et peuvent effectuer des évaluations en observant leur travail sur le site web.

## Stratégies de motivation

Les étudiants auront beaucoup de plaisir à construire ces circuits en ligne, car ils pourront construire, tester et dépanner leurs circuits en toute sécurité avec un retour d'information immédiat dans le simulateur.

## Attentes et contenus d'apprentissage à l'appui des programmes d'études de la 11e à la 12e année en Ontario

### Attentes

A2. expliquer le rôle des principaux composants électroniques et dispositifs électriques utilisés dans les interfaces matérielles et les systèmes de commande.

B1. concevoir des circuits électroniques destinés à des robots ou des interfaces matérielles en appliquant sa connaissance processus de design.

### Contenus d'apprentissage

A2.1 expliquer le rôle et les avantages de divers types d'interfaces conçues pour des ordinateurs ou des systèmes de commande (p. ex., ordinateur personnel, microcontrôleur, automate programmable industriel).

A2.3 expliquer le rôle des composants électroniques (p. ex., résistance, condensateur, transistor, diode électroluminescente [DEL]) couramment insérés dans les circuits imprimés.

A2.5 décrire l'évolution technologique des composants électroniques (p. ex., transistor, circuit intégré) et les tendances actuelles dans ce secteur (p. ex., miniaturisation, réduction des coûts).

B1.6 interpréter divers diagrammes de circuits électroniques comportant des portes logiques (p. ex., ET, NON, OU, OU-EXCLUSIF, NON-ET) et des composants (p. ex., décodeur, multiplexeur).

B1.7 concevoir un circuit (p. ex., logique, électronique, imprimé) en utilisant diverses fonctions d'un logiciel de conception assistée par ordinateur [CAO] (p. ex., représentation schématique d'un circuit, synthèse logique, simulation).

B1.8 concevoir des circuits électroniques pouvant faire partie d'un robot ou d'une interface matérielle (p. ex., pilote de moteur pas-à-pas, feu de circulation, système d'alarme) en utilisant le processus de design (p. ex., élaboration d'un cadre, évaluation d'une solution, planification du projet).

## Préoccupations et attentes liées à la santé et la sécurité

Tous les travaux pratiques peuvent être effectués à l'aide d'un simulateur en ligne, de sorte que l'on se soucie peu de la sécurité. Si les activités sont effectuées en classe, les préoccupations peuvent porter sur la surchauffe des composants qui ne sont pas correctement branchés. Les enseignants doivent superviser activement et encourager les élèves à utiliser des lunettes de sécurité.

## Ressources OCTE SÉCURIdoc et outilSÉCUR

Veuillez consulter le [document SÉCURIdoc](#) situé sur le site Web d'OCTE.

## Différenciation pédagogique

Many supporting documents for students and teachers have been provided to ensure all types of learners are given the supports they need. All circuit activities are provided with graphical guides to follow and major concepts have corresponding video lessons with screen capture instructions.

## Évaluation

### Évaluation en tant qu'apprentissage

Les élèves recevront un retour d'information descriptif sur les diverses activités du circuit qu'ils auront réalisées

## Réflexion

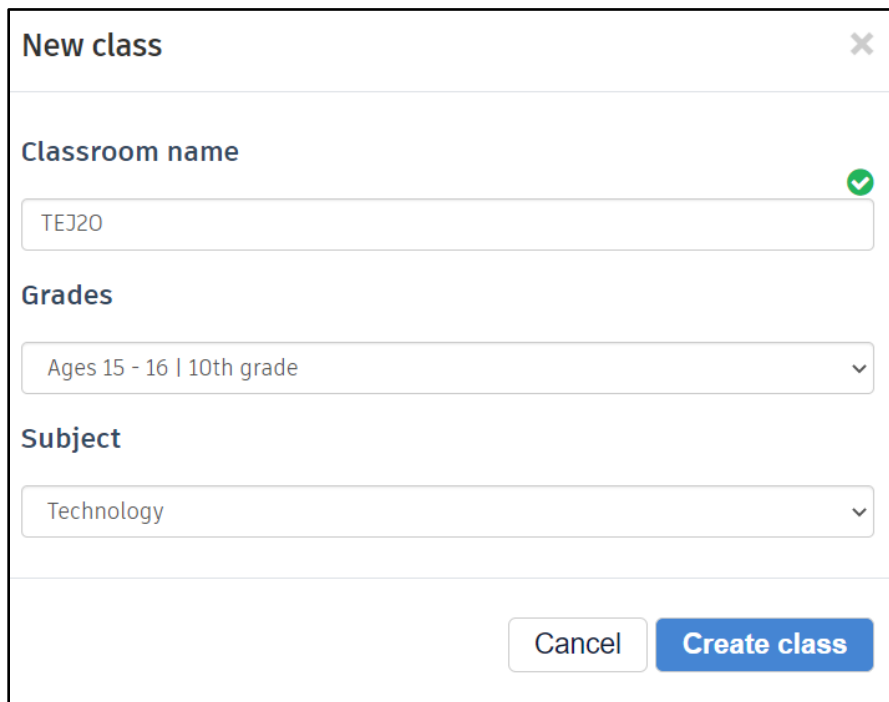
Les enseignants peuvent demander aux élèves de remplir un rapport de conception, de réfléchir ou de créer un document pour consolider leur apprentissage. Ce serait un bon moyen de saisir la compréhension de l'élève dans un format sommatif et de l'utiliser pour préparer son examen, son entrée dans l'enseignement supérieur ou sur le marché du travail.

# Annexe A – Instructions pour utiliser des classes sur Tinkercad

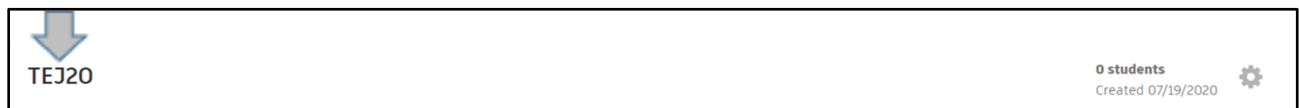
**Étape 1:** Cliquez sur “Classes”



**Étape 2:** Créer une « New class » et choisir un nom, niveau et sujet.

A screenshot of the 'New class' dialog box. It has a title bar 'New class' with a close button. Below the title bar are three input fields: 'Classroom name' with the value 'TEJ20' and a green checkmark; 'Grades' with the value 'Ages 15 - 16 | 10th grade'; and 'Subject' with the value 'Technology'. At the bottom are two buttons: 'Cancel' and 'Create class'.

**Étape 3:** Cliquez sur votre nouvelle classe pour ajouter des élèves




**Étape 4:** Ajout d'élèves, vous pouvez ajouter un élève à la fois OU cliquer sur coller une liste d'élèves.

### Add students

Class: TEJ20

Name	Nickname	
<input type="text" value="For example: Amy Zeebo"/>	<input type="text" value="AmyZ56"/>	<input type="button" value="Save Changes"/>

Nickname must be 3 or more characters, numbers, or letters.




**Étape 5:** Tapez ou collez les noms des élèves puis cliquez sur "Add # students".

### Add students





Class: TEJ20

Student names

Student One  
Student Two  
Student Three  
Student Four



**Étape 6:** Donnez à chaque élève un surnom unique. Utilisez le nom de famille et la première initiale. Exemple : Bill Gates serait "gatesb" pour faciliter l'information de tous les élèves.





<input type="checkbox"/> Student name	Nickname
<input type="checkbox"/>  Student One	<input type="text" value="ones"/> 
<input type="checkbox"/>  Student Two	<input type="text" value="twos"/> 

**Étape 7:** Vous devez maintenant partager le "Class Code" avec vos élèves.

< TEJ20 ×

Students Designs

Select action ▾ Add students **Class Code**

<input type="checkbox"/> Student name	Nickname
<input type="checkbox"/>  Student One	ones 
<input type="checkbox"/>  Student Two	twos 

**Étape 8:** Utilisez le code de classe donné et partagez le lien avec vos élèves.

Log in to TEJ20 with:

# TPI6 29LS DHMR

Student instructions

Have a class link?

1. Go to your class at <https://www.tinkercad.com/joinclass/TPI629LSDHMR>.
2. Enter the **Nickname** your teacher assigned you.

Have a class code?

1. Go to <https://www.tinkercad.com/joinclass>
2. Enter your class code: **TPI629LSDHMR**
3. Enter the **Nickname** your teacher assigned you.



## Activité 1a - Numérique vs Analogue avec Arduino – Fiche vide

### Types de signaux dans les ordinateurs et l'Arduino

Types	Sorties	Entrées
Composante d'ordinateur		
*Numérique (1 ou 0)		
*Analogue (niveaux)		

\* Les signaux numériques et analogues sont utilisés avec l'électronique et dans notre cas l'Arduino.

## Activité 1a – Numérique vs Analogue avec Arduino – Enseignant

### Types de signaux dans les ordinateurs et l'Arduino

Types	Sorties	Entrées
Composante d'ordinateur	Écran, haut-parleur, imprimante, etc.	Souris, clavier, microphone, etc.
*Numérique (1 ou 0)	DEL, moteur, sonnette, etc.	Interrupteur, bouton, etc.
*Analogue (niveaux)	DEL(ajustable), moteur (vitesse), etc.	Capteur à température, capteur à distance IR, etc.

\* Les signaux numériques et analogues sont utilisés avec l'électronique et dans notre cas l'Arduino.

# Activité 1a – Activités de programmation pour l'Arduino

## TEJ3M – Introduction aux activités de programmation

Vous devez rédiger (à l'aide d'un bloc-notes ou sur papier) un ensemble d'instructions de programmation pour accomplir les tâches suivantes.

\*À NOTER: Si vous utilisez le simulateur TINKERCAD, vous pourrez tester votre code. De toute façon, nous utiliserons le site web TINKERCAD pour "brancher" quelques exemples d'Arduino. Vous aurez besoin d'un compte.

D'autres détails:

1. Il n'y a pas d'écran standard "entrée/sortie" car la plupart des entrées et sorties proviennent de pièce électronique (DEL, moteurs, servomoteurs, boutons, capteurs de température, etc.)
2. Nous pouvons utiliser un moniteur série (Serial Monitor) pour sortir et voir les données, mais il n'est pas très bien organisé.
3. Il n'y a pas d'interfaces graphiques.

Pour écrire sur le moniteur en série, le code serait le suivant :

Dans la section void setup() écrire:

```
Serial.begin(9600);
```

Et puis dans la section void loop() écrire:

```
Serial.println("blah");  
//vous pouvez aussi mettre des variables dans cette écriture
```

\* N'oubliez pas que vous devez inclure les deux sections de code requises (setup, main) pour chaque problème

### Tâche #1 – Sortie de base

Faites apparaître votre nom dans le moniteur de série.

### Tâche #2 – Variables et arithmétique de base

Déclarer deux variables ("num1", "num2") qui contiennent les chiffres 16 et 64 respectivement. Votre code doit calculer et sortir la moyenne des deux valeurs sur le moniteur en série.

<https://www.arduino.cc/en/Reference/Arithmetic>

### Tâche #3 - Répétition

Vous devez avoir une variable int appelée "numTimes" avec une valeur de 15. Elle imprime ensuite la valeur de 0 à "numTimes" (moins un) dans le moniteur série.

<https://www.arduino.cc/en/reference/for>

#### Tâche #4 - Sélection

Votre programme aura une valeur int appelée "num". (Vous pouvez le changer au nombre entier que vous voulez.) Il affichera "Odd" ou "Even" sur le moniteur série en fonction de l'imparité ou de la parité du nombre.

<https://www.arduino.cc/en/reference/modulo>

#### Supplément – Arithmétique complexe et décisions

Écrivez le code pour convertir un nombre décimal (entre 0 et 15) en binaire. Le nombre décimal sera stocké dans la variable "decimal" qui est int. Pour les besoins de cette tâche, vous pouvez donner à "decimal" une valeur de 10 car la saisie au clavier est assez difficile en Arduino. La sortie du moniteur série devrait ressembler à ceci :

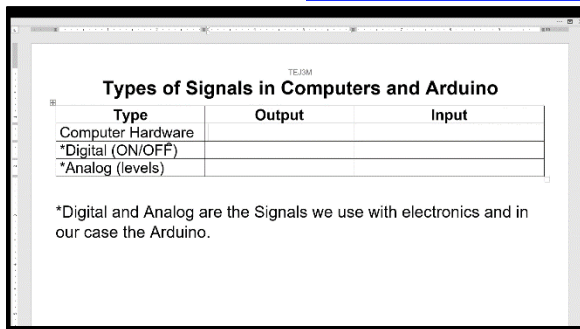
Décimal: 10

Binaire: 1010

<https://www.arduino.cc/en/Reference/While>

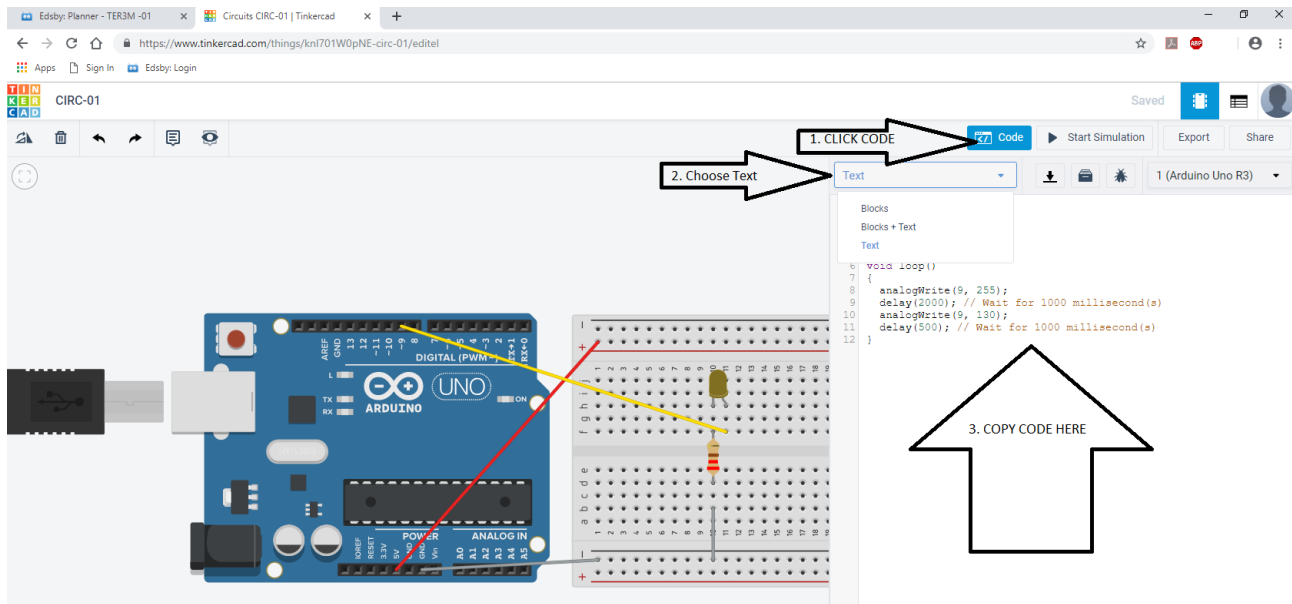
<https://www.arduino.cc/en/Reference/Arithmetic>

#### Activité 1a: Vidéo – [Analogue vs numérique / entrée vs sortie](#)

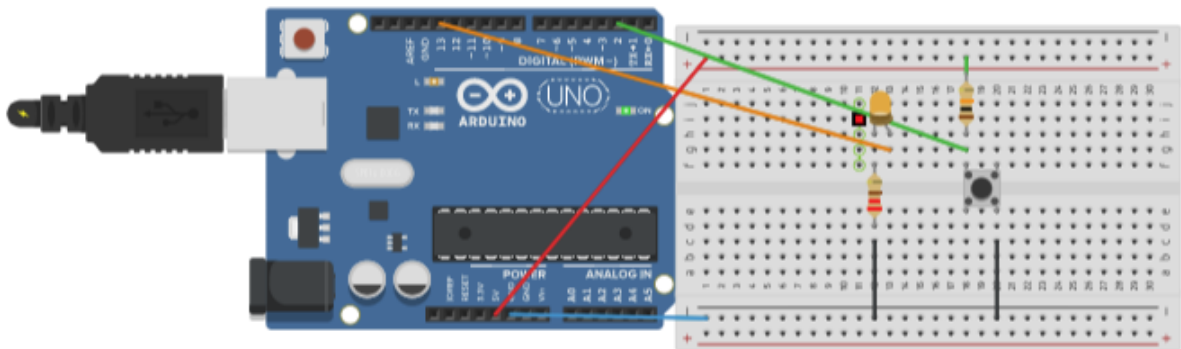


[https://www.octe.ca/application/files/2415/9693/0525/Video\\_1a.\\_Analog\\_vs\\_Digital\\_and\\_Input\\_vs\\_Output.mp4](https://www.octe.ca/application/files/2415/9693/0525/Video_1a._Analog_vs_Digital_and_Input_vs_Output.mp4)

## Activité 1b - Tinkercad, brochez et programmez un bouton

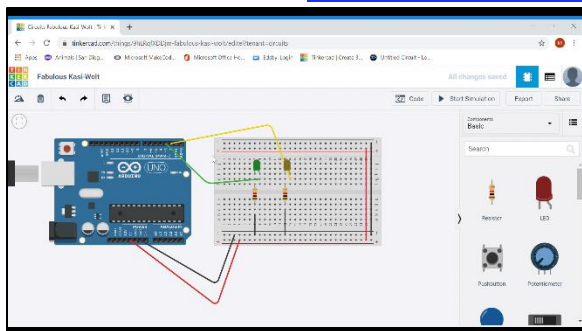


### Button Activité Wiring Diagram:



\*Note you can use 560 Ohm (Grn, Blue, Brn) instead of 220 Ohm (Red,Red, Brn)

### Activité 1b: Vidéo – Introduction à Arduino avec Tinkercad



[https://www.octe.ca/application/files/9015/9708/8567/Video\\_1b\\_Basic\\_Intro\\_to\\_Arduino\\_in\\_Tinkercad.mp4](https://www.octe.ca/application/files/9015/9708/8567/Video_1b_Basic_Intro_to_Arduino_in_Tinkercad.mp4)

## Code pour l'activité du bouton:

```
// Code pour bouton

const int buttonPin = 2;      // le numéro de la broche du bouton
const int ledPin = 13;       // le numéro de la broche de la DEL

// variables vont changer:
int buttonState = 0;         // variable pour lire l'état du
                             // bouton

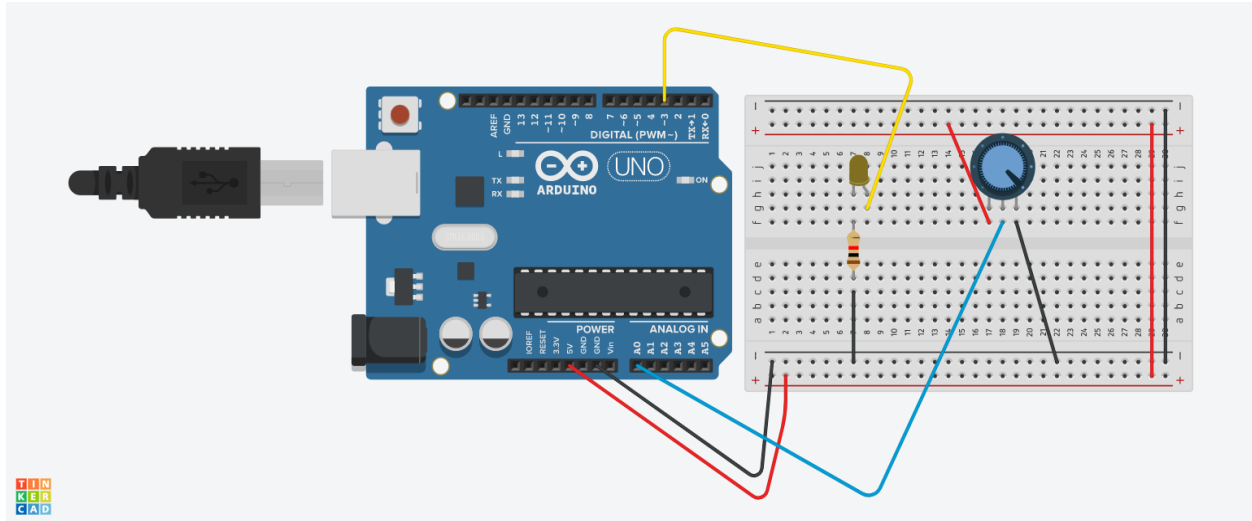
void setup() {

  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

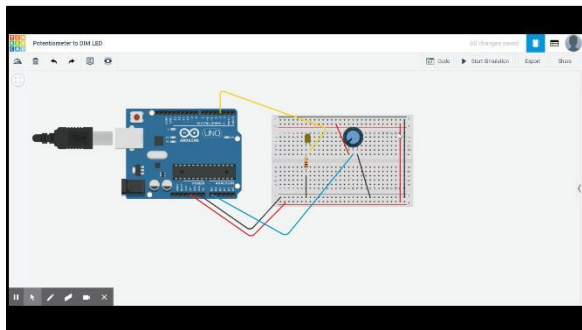
void loop(){
  // lire la valeur de l'état du bouton:
  buttonState = digitalRead(buttonPin);

  // vérifier si le bouton est enfoncé.
  // si oui, le buttonState est HIGH:
  if (buttonState == HIGH) {
    // allumé la DEL on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // sinon fermer la DEL off:
    digitalWrite(ledPin, LOW);
  }
}
```

## Activité 1c - Potentiomètre pour abaisser un DEL - Entrée et sortie analogue



## Activité 1c: Vidéo – [Potentiomètre pour abaisser un DEL - Entrée et sortie analogue](https://www.octe.ca/application/files/3815/9693/1715/Video_1c._Potentiometer_to_DIM_LED_Analog_Input_and_Output.mp4)



[https://www.octe.ca/application/files/3815/9693/1715/Video\\_1c.\\_Potentiometer\\_to\\_DIM\\_LED\\_Analog\\_Input\\_and\\_Output.mp4](https://www.octe.ca/application/files/3815/9693/1715/Video_1c._Potentiometer_to_DIM_LED_Analog_Input_and_Output.mp4)

## Code pour l'activité du potentiomètre:

```
/*
  Analog input, analog output, serial output
*/

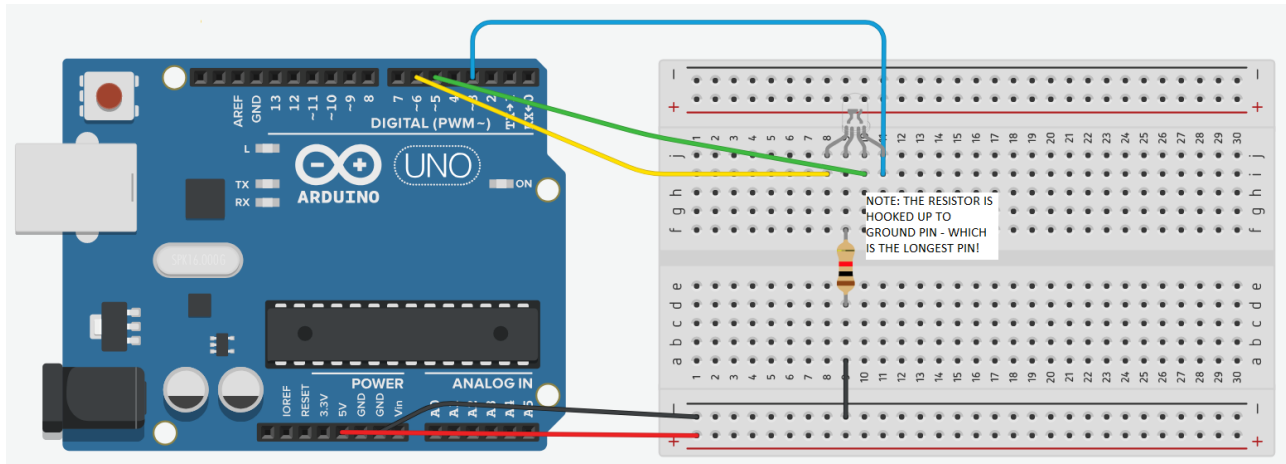
int sensorValue = 0;

int outputValue = 0;

void setup()
{
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  // lire la valeur analogue:
  sensorValue = analogRead(A0);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  // changer la valeur en sortie analogue:
  analogWrite(3, outputValue);
  // écrire les résultat dans le serial monitor:
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
  // attendre 2 millisecondes avant de faire la prochaine boucle
  //loop pour que le analog-to-digital converter se place après la
  // dernière lecture:
  delay(2); // attendre 2 millisecond(s)
}
```

## Activité 2 – DEL RGB (Sortie analogue)



### Code pour DEL RGB:

```
//RGB Analog Example

void setup () {

pinMode (3, OUTPUT ); // Blue
pinMode (5, OUTPUT ); // Green
pinMode (6, OUTPUT ); // Red

}

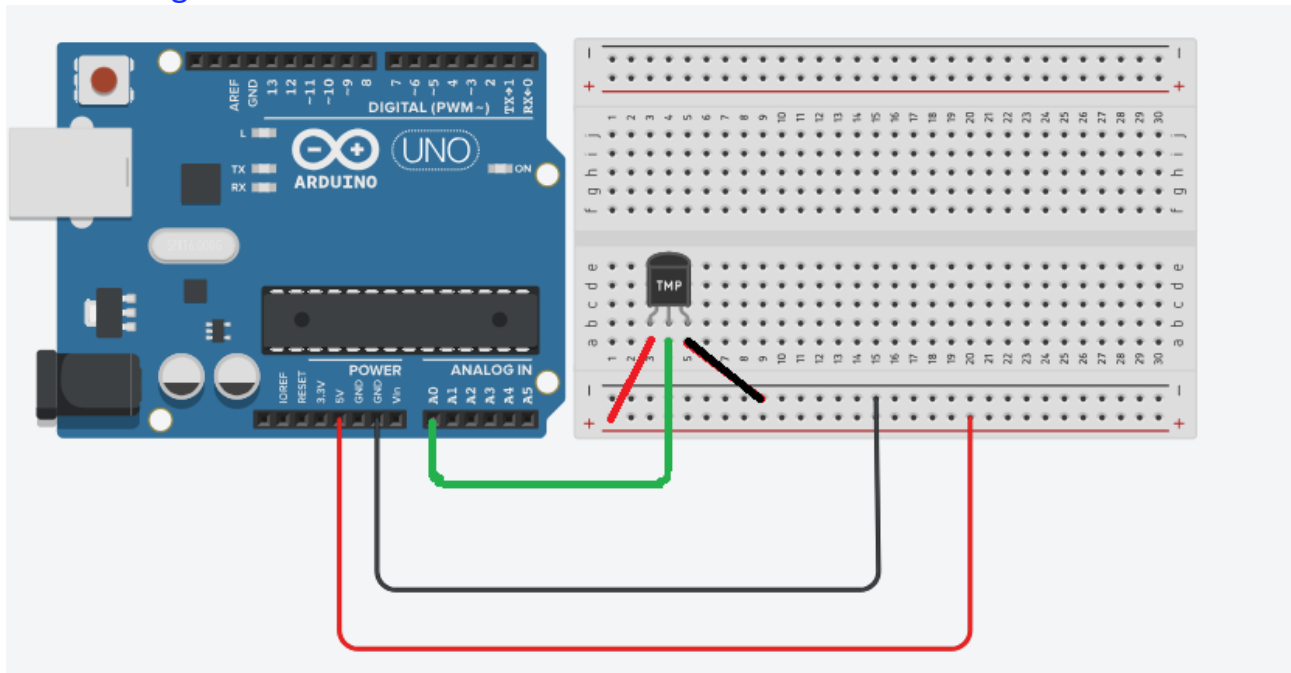
void loop () {
//Cette combinaison produit un orange que vous pouvez recréer
//avec n'importe valeur de couleur
analogWrite(3, 0);
analogWrite(5, 128);
analogWrite(6, 255);

}
```



## Activité 3 – Capteur de température (Entrée analogue)

TMP36 Diagramme:



## Code du TMP 36:

```
//TMP36 Pin Variables
int temperaturePin = A0; //brancher TMP36's Vout à A0

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  float temperature = getVoltage(temperaturePin);
  //aller chercher la valeur du voltage
  //lire du capteur de température
  temperature = (temperature - .5) * 100;
  //convertir 10 mv par degré avec 500mV offset
  //par degrés ((voltage 500mV) fois 100)
  Serial.println(temperature);
  //écrire le résultat
  delay(1000);
}

// getVoltage() - retourne le voltage de l'entrée analogue pin

float getVoltage(int pin){
  return (analogRead(pin) * .004882814); //converti d'un 0
                                          //à 1023 digital
                                          //range de 0 à 5
                                          //volts (chaque 1
                                          //lecture égale ~ 5
                                          //millivolts
}
}
```



## Coder avec Tinkercad:

```
//TMP36 Pin Variables
int temperaturePin = A0; //brancher TMP36's Vout à A0

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  float temperature = getVoltage(temperaturePin);
  //lire la valeur en voltage reading du capteur temp
  temperature = (temperature - .5) * 100;
  //convertir 10 mv par degré avec 500mV offset
  //par degrés ((voltage 500mV) fois 100)
  Serial.println(temperature);
  //écrire le résultat
  delay(1000);
}

// getVoltage() - retourne le voltage de l'entrée analogue pin

float getVoltage(int pin){
  return (analogRead(pin) * .004882814); //converti d'un 0
                                          //à 1023 digital
                                          //range de 0 à 5
                                          //volts (chaque 1
                                          //lecture égale ~ 5
                                          //millivolts
}
}
```

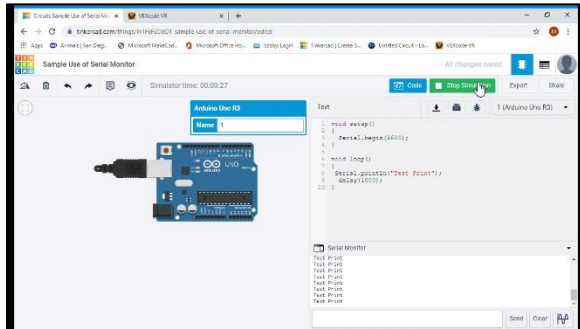
**ÉTAPE #2:** Revoir les fils et le code de sortie des DEL RGB de l'activité #2

**ÉTAPE #3:** Vous devez COMBINER les deux codes. (Note importante: Vous pouvez avoir seulement un void setup() fonction et UN void loop() fonction. Vous devez mélanger tous les codes de ces sections.

**ÉTAPE #4:** Circuit final – Votre circuit doit être une représentation visuelle d'une température. Le DEL RGB sera BLEU si la température est trop froide (en bas de 17°C), le DEL RGB sera VERT si la température est "Normale" (entre 17°C et 22°C), et le DEL RGB sera ROUGE si la température est trop chaude (au-delà de 22°C).

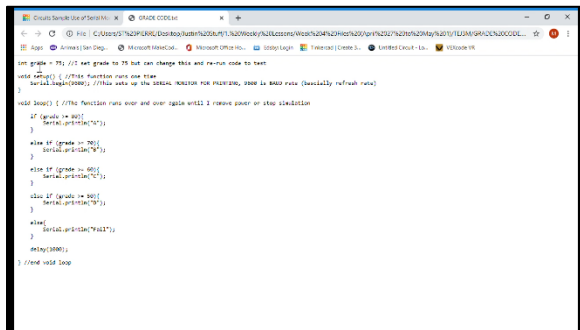
\*Voir les vidéos 3a. Utilisation du Serial Monitor dans Tinkercad et 3b. Utilisation du If Statement dans Tinkercad – Grade Example

Lien Web : Vidéo – [Use of Serial Monitor in Tinkercad](https://www.octe.ca/application/files/7415/9708/8223/Video_3a_Use_of_Serial_Monitor_in_Tinkercad.mp4)



[https://www.octe.ca/application/files/7415/9708/8223/Video\\_3a\\_Use\\_of\\_Serial\\_Monitor\\_in\\_Tinkercad.mp4](https://www.octe.ca/application/files/7415/9708/8223/Video_3a_Use_of_Serial_Monitor_in_Tinkercad.mp4)

Lien Web : Vidéo – [Using If Statements in Tinkercad - Grade Example](https://www.octe.ca/application/files/8215/9708/8065/Video_3b_Using_If_Statements_in_Tinkercad.mp4)



[https://www.octe.ca/application/files/8215/9708/8065/Video\\_3b\\_Using\\_If\\_Statements\\_in\\_Tinkercad.mp4](https://www.octe.ca/application/files/8215/9708/8065/Video_3b_Using_If_Statements_in_Tinkercad.mp4)

**ÉTAPE #5:** SAUVEGARDER le circuit combiné comme “Control de température RGB” sur Tinkercad et soumettre les captures d’écrans du **CIRCUIT et du CODE**

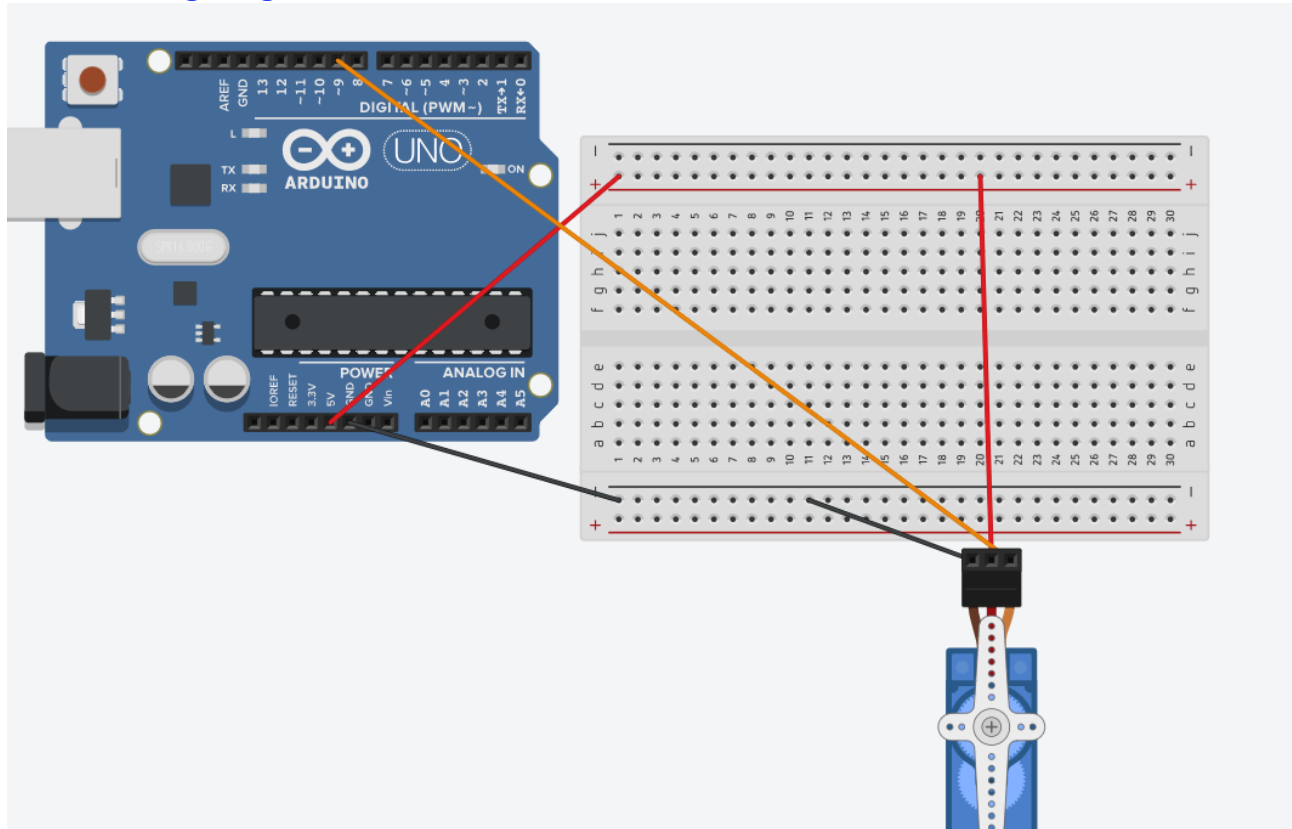
Liste de contrôle pour la note:

Itème	Note
TMP36 Câblage	___ / 3
RGB DEL Câblage	___ / 3
Code combiné	___ / 4

\* Un rappel d'utiliser les fils de couleur appropriée pour l'alimentation et la mise à la terre.

# Activité 4 –Micro servomoteur (avec librairies d’Arduino)

Servo Wiring Diagram:



## Servo Code:

```
#include <Servo.h> //Ceci importe la Servo Library Code

Servo myservo; //créer un servo object pour contrôler un servo

int pos = 0; // variable pour garder en mémoire la position
            //du servo

void setup()
{
  myservo.attach(9); // branche le servo à la pin 9 au
                    //servo object
}

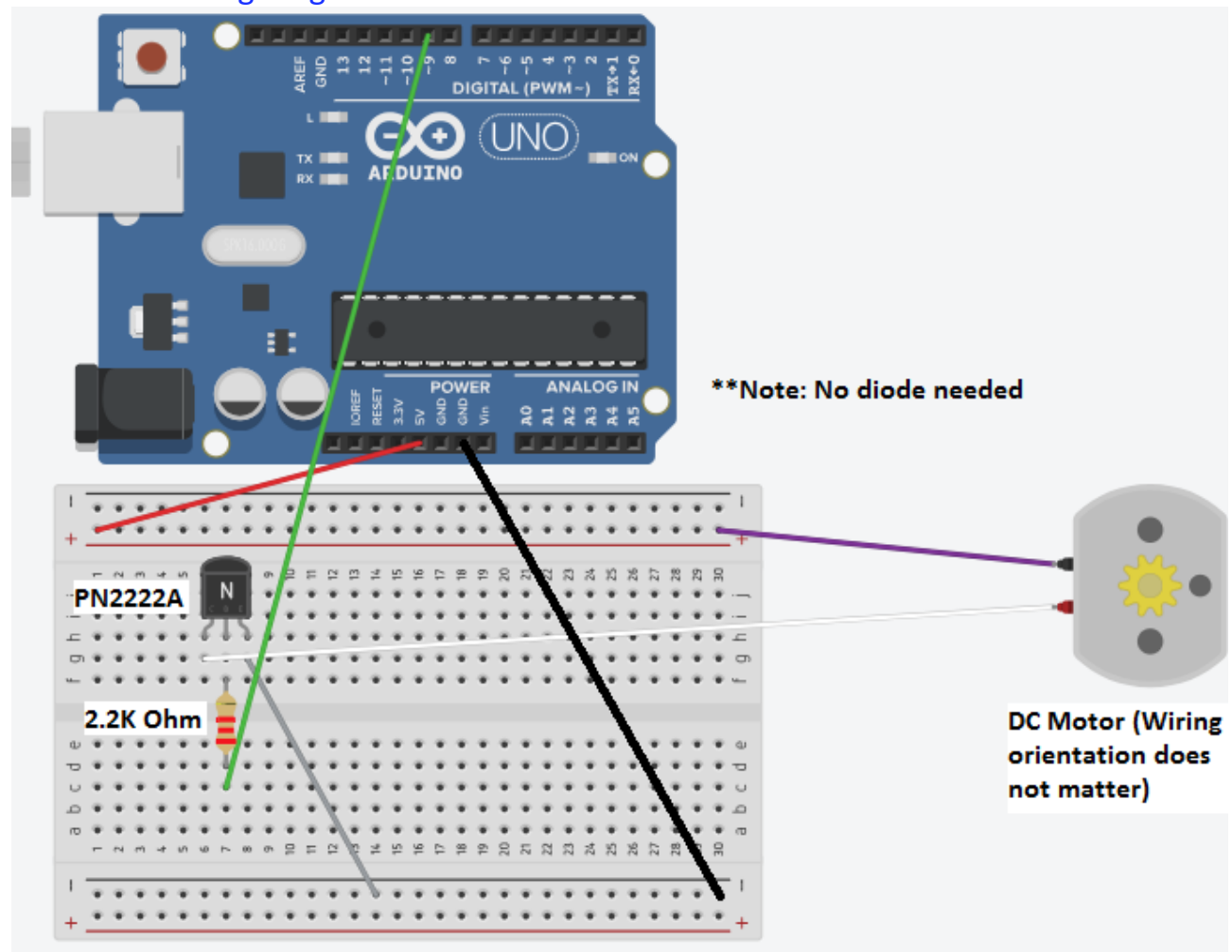
void loop()
{
  for(pos = 0; pos < 180; pos += 1) //va de 0 à 180 degrés
  {
    myservo.write(pos); // en étapes de 1 degré
                        // dit au servo d'aller à
                        //la position de la
                        //variable 'pos'

    delay(15); // attendre 15ms le servo
              //se rendre à sa position
  }
  for(pos = 180; pos>=1; pos--=1) //de 180 degrés to 0 deg
  {
    myservo.write(pos); // dit au servo d'aller à
                        //position de la var 'pos'

    delay(15); // attendre 15ms pour servo
              //se rendre à sa position
  }
}
```

## Activité 5 – Contrôle d'un moteur CC(DC) Arduino

DC Motor Wiring Diagram:



## DC Motor Code #1 (Basic Motor ON/OFF):

```
//Basic Motor ON and OFF

int motorPin = 9; // définir la pin du moteur est branché à 9
                  // (si tu utilises pin 9,10,11 ou 3 tu peux
                  //aussi contrôler la vitesse)

void setup()
{
  pinMode(motorPin, OUTPUT);
}

void loop()
{
  int onTime = 2500; //nombre de millisec moteur va tourner on

  int offTime = 1000;//nombre de millisec moteur ne tourne pas off

  digitalWrite(motorPin, HIGH); // met le moteur à On
  delay(onTime);                // attend le onTime milliseconds
  digitalWrite(motorPin, LOW);  // met le moteur à Off
  delay(offTime);               //attend le offTime milliseconds
}
```

## DC Motor Code #2 (Analog Speed):

```
//Basic Motor with Analog Speed

int motorPin = 9; // définir la pin branché au moteur

void setup()
{
  pinMode(motorPin, OUTPUT);
}

void loop()
{
  int fastSpeed = 200; // un nombre entre 0 (arrêter) et 255
                        //(pleine vitesse)
  int fastTime = 2500; //nombre en millisecondes pour haute vitesse

  int slowSpeed = 50; // un nombre entre 0 (arrêter) et 255
                      //(pleine vitesse)
  int slowTime = 1000; // nombre en millisecondes pour basse vitesse

  analogWrite(motorPin, fastSpeed); // tourne le moteur à vite
  delay(onTime); // attend pour fastTime ms
  analogWrite(motorPin, slowSpeed); // tourne le moteur à lent
  delay(offTime); // attend pour slowTime ms
}
```

### DC Motor Code #3 (Accélération):

```
//Basic Motor with Acceleration using For Loops

int motorPin = 9; // définir la pin branché au moteur

void setup()
{
  pinMode(motorPin, OUTPUT);
}

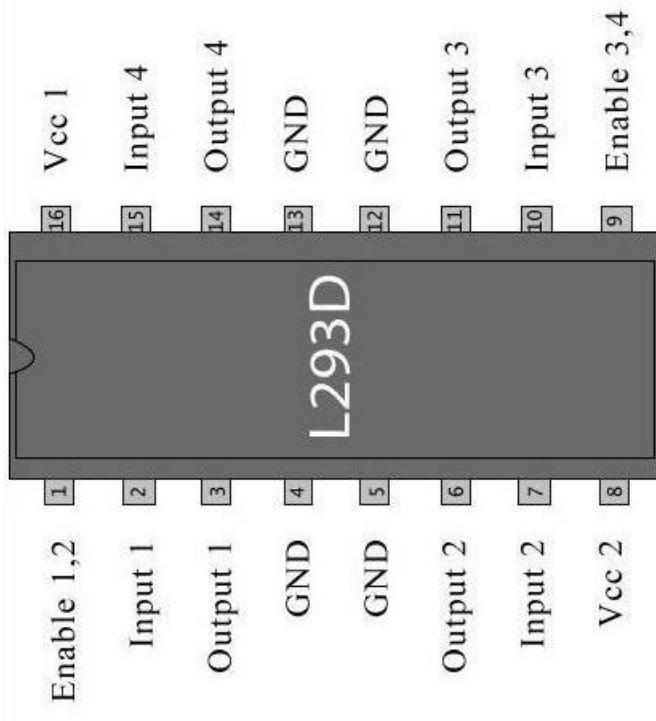
void loop()
{
  int delayTime = 50; //millisecondes entre chaque vitesse

  //Accelerates the motor
  for(int i = 0; i < 256; i++){ //passe à travers chaque vitesse
                                //de 0 à 255
    analogWrite(motorPin, i); //défini une nouvelle vitesse 'i'
    delay(delayTime);        //attend pour delayTime ms
  }

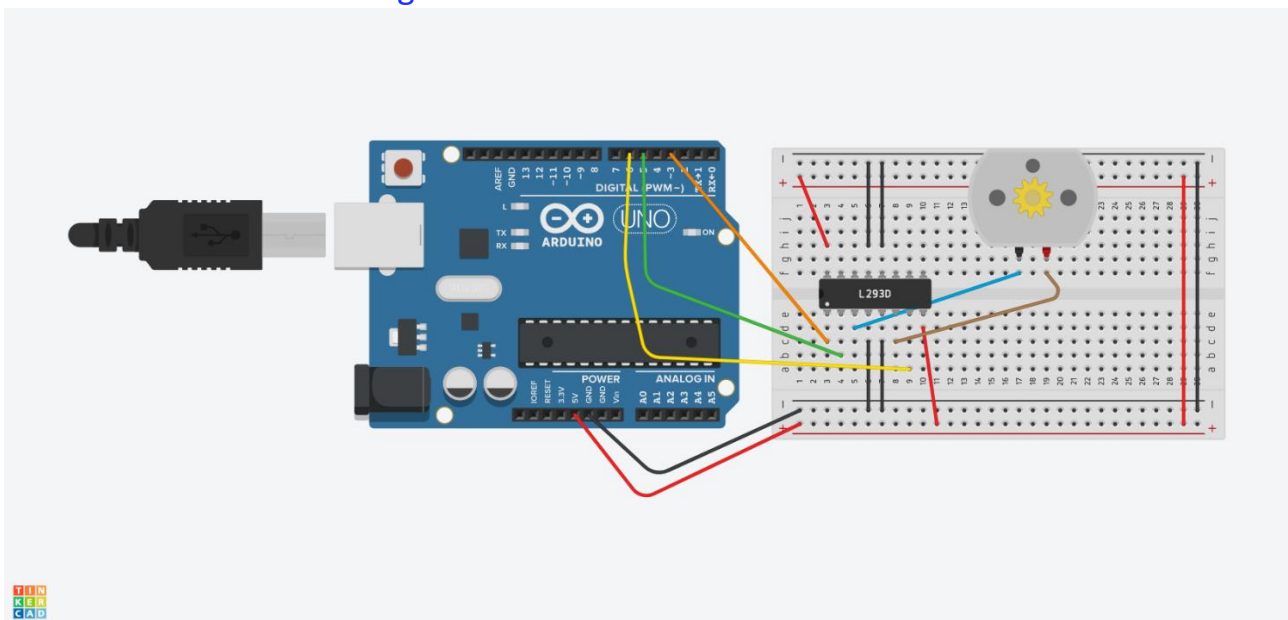
  //Decelerates the motor
  for(int i = 255; i >= 0; i--){ //passe à travers chaque vitesse
                                  //de 255 à 0
    analogWrite(motorPin, i); //défini une nouvelle vitesse 'i'
    delay(delayTime);        //attend delayTime ms
  }
}
```

## Activité 5 – Contrôle d'un moteur DC - Arduino L293D Motor Driver

L293D Pinout:



L293D Motor Driver Wiring:



TI  
X  
G  
CAD

## L293D Code:

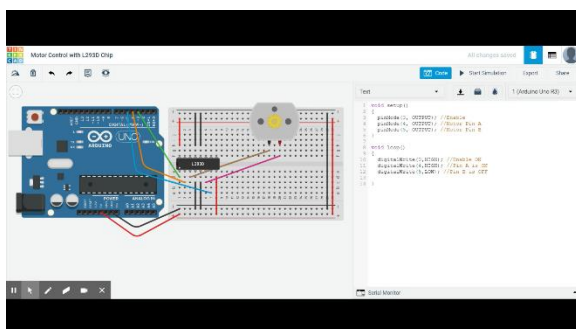
```
void setup()
{
  pinMode(3, OUTPUT); //ENABLE PIN
  pinMode(5, OUTPUT); //MOTOR PIN A
  pinMode(6, OUTPUT); //MOTOR PIN B
}

void loop()
{
  //MOTOR SPINS IN DIRECTION #1
  digitalWrite(3, HIGH); //ENABLE ON
  digitalWrite(5, HIGH); //MOTOR PIN A ON
  digitalWrite(6, LOW); //MOTOR PIN B OFF
  delay(2000); //delay 2 seconds

  //MOTOR SPINS IN DIRECTION #2
  digitalWrite(3, HIGH); //ENABLE ON
  digitalWrite(5, LOW); //MOTOR PIN A ON
  digitalWrite(6, HIGH); //MOTOR PIN B OFF
  delay(2000); //delay 2 seconds

  //MOTOR STOPS
  digitalWrite(3, LOW); //ENABLE OFF
  delay(2000); //delay 2 seconds
}
```

Lien Web : Vidéo – [Bi-directional Motor Control with L293D](#)



[https://www.octe.ca/application/files/7015/9692/6223/Video\\_5.\\_DC\\_Motor\\_Control\\_with\\_L293D.mp4](https://www.octe.ca/application/files/7015/9692/6223/Video_5._DC_Motor_Control_with_L293D.mp4)

## Activité 5 - Contrôle d'un moteur DC – Arduino - Questionnaire







Nom: \_\_\_\_\_

/ 17

### Section I – Composants électroniques/Interfaçage

/ 6

Utilisez la banque de noms ci-dessous pour identifier chaque élément

Composante	Nom de la composante
	
	
	
	
	
	

Banque de composantes:

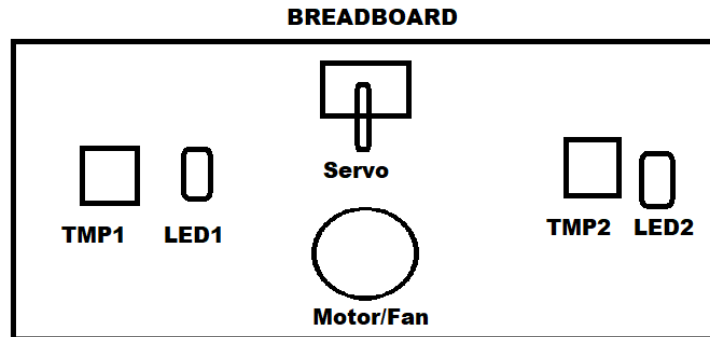
Lumière	Transistor variable	Diode électroluminescente	Capaciteur	Sonnette
Afficheur à 7 segments	Servo	Interface USB	Diode	Resistance
Transistor	Interrupteur à 2 voies	Bouton momentané	DEL RGB	Moteur DC



## Activité 6 – Projet d'échangeur d'air automatisé

Dans cette activité, vous combinerez tous les composants rencontrés à ce jour et créerez un système d'échangeur d'air automatisé.

Votre système sera composé de deux capteurs TMP36, de 2 DEL RGB, d'un moteur bidirectionnel (Ventilateur) avec 293D et d'un servo.



- Un capteur de température sera monté de chaque côté. Le moteur sera monté au milieu (supposons que le moteur ait des palettes (similaire à ce ventilateur de PC).



- Le DEL RGB de chaque côté indiquera quel côté est "plus chaud" (ROUGE) et "plus froid" (BLEU). Les deux peuvent être verts s'ils se trouvent à moins de 1°C l'un de l'autre. (le moteur doit s'arrêter à ce point également, et le bras du servo doit pointer vers le HAUT)
- Le servo indiquera le débit (imaginez qu'une flèche y soit attachée). Comme mentionné ci-dessus, il devrait pointer vers le haut si le moteur est éteint.
- La direction du moteur dans votre circuit n'est pas trop importante à ce stade (en réalité, en échangeant rapidement les deux fils du moteur à courant continu, nous pourrions assurer la bonne direction). Ce qui est important, c'est que votre moteur puisse tourner dans les deux sens et s'arrêter. (Possibilité d'extension : plus la différence de température est importante, plus vite le moteur tourne.)
- Vous devez faire votre circuit avec votre compte de la classe Tinkercad pour vous assurer que l'enseignant peut correctement effectuer le test. Vous pouvez également créer une capture d'écran vidéo de votre circuit en cours d'exécution en vous assurant de bien coder et tester tous les cas de différences de température. (Exemple :  $LS > RS$ ,  $LS = RS$ ,  $LS < RS$ )

## Grille d'évaluation pour circuits:

Design du circuit	Niveau 1	Niveau 2	Niveau 3	Niveau 4
Attentes : utiliser un processus de design pour concevoir, construire et tester en toute sécurité des circuits d'interface ou de robotique	Le design des circuits n'intègre pas tous les composants et le codage requis.	Le design des circuits comprend la plupart des composants nécessaires et une portion du code fonctionne.	Le design des circuits est bon avec des erreurs mineures dans le câblage et/ou le code.	Le design des circuits intègre tous les composants nécessaires et répond à toutes les exigences de la tâche en utilisant le câblage et le code appropriés.
Attentes : utiliser des programmes informatiques pour simuler le fonctionnement des circuits	La simulation du circuit ne fonctionne pas correctement.	Certaines parties du circuit fonctionnent correctement en simulation.	Le circuit fonctionne bien en simulation avec des problèmes mineurs.	La simulation de circuit fonctionne très bien lors de tous les essais.

## Journal de solution des circuits :

Problèmes	Solutions essayées	Solution finale
Exemple : Le TMP36 ne produit pas une tension/température appropriée	* Vérification du code pour s'assurer que la broche d'entrée analogue était correcte, vérification du câblage pour s'assurer que la bonne broche analogue était branchée au TMP36	* J'ai réalisé que j'avais confondu Vout et Alimentation lors du câblage du TMP36



4. Explorez un peu plus Tinkercad et voyez quels autres capteurs ou modules pourraient être utilisés lors de projets avec Arduino.

Capteur / Modules	Description

## Références

Compétences du 21e Siècle: Document de Réflexion. Phase 1: Définir les Compétences du 21e Siècle pour l'Ontario. Édition de l'automne, 2016.  
[https://pedagogienumeriqueenaction.cforp.ca/wp-content/uploads/2016/02/Ontario-21st-century-competencies-foundation-FINAL-FR\\_AODA\\_EDUGAINS\\_Feb-19\\_16.pdf](https://pedagogienumeriqueenaction.cforp.ca/wp-content/uploads/2016/02/Ontario-21st-century-competencies-foundation-FINAL-FR_AODA_EDUGAINS_Feb-19_16.pdf)

4-Pin RGB LED (Image), 2020 <https://www.digikey.com/product-detail/en/kingbright/WP154A4SUREQBFZGC/754-1615-ND/3084119#images>

Arduino Cheat Sheet – Jeroen Doggen, AP University College Antwerp  
[https://raw.githubusercontent.com/jeroendoggen/Arduino-cheat-sheet/master/examples/Arduino-Cheat-Sheet\\_v0.1.png](https://raw.githubusercontent.com/jeroendoggen/Arduino-cheat-sheet/master/examples/Arduino-Cheat-Sheet_v0.1.png)

Arduino (Website), 2020 <https://www.arduino.cc/>

Autodesk Tinkercad (Website), 2020 <https://www.tinkercad.com/>

Codes des cours de spécialisation : Éducation Technologique, 11e et 12e année, édition révisée,(2009)

<http://www.edu.gov.on.ca/eng/curriculum/secondary/techedemphasiscourses.pdf>

DC Motors (Image), 2020 <https://abra-electronics.com/electromechanical/motors/dc-motors/mot-dc-160-6v-high-speed-35000-rpm-dc-motor.html>

Faire croître le succès : Évaluation et communication du rendement des élèves fréquentant les écoles de l'Ontario. Première édition, 1re–12e année. 2010.

<http://www.edu.gov.on.ca/fre/policyfunding/growSuccessfr.pdf>

Hitec HS-55 Servo Motor (Image), 2020 <https://hitecrad.com/products/servos/micro-and-mini-servos/analog-micro-and-mini-servos/hs-55/product>

Images for Quiz (Images), 2020 <https://www.ardx.org/src/guide/2/ARDX-EG-SOLA-WEB.pdf>

L293D Pinout (Image), 2020 <https://www.engineersgarage.com/contributions/dc-motor-interfacing-with-beaglebone-black-part-5-15/>

L'apprentissage pour tous : Guide d'évaluation et d'enseignement efficaces pour tous les élèves de la maternelle à la 12e année, 2013

<http://www.edu.gov.on.ca/fre/general/elemsec/speced/LearningforAll2013Fr.pdf>

PC Fan (Image), 2020 <https://www.digikey.ca/en/products/detail/sunon-fans/MF60201V1-1000U-A99/7652230?s=N4lgTCBcDa4KwE4C0BGAGHMSByAREAugL5A>

Red LED (Image), 2020 <https://www.digikey.com/product-detail/en/kingbright/WP7113ID14V/WP7113ID14V-ND/3084219#images>

Resistor (Image), 2020 <https://www.digikey.com/product-detail/en/ohmite/OX101KE/OX101KE-ND/823903>

The Differentiated Instruction Scrapbook

<http://www.edugains.ca/resources/DI/EducatorsPackages/DIEducatorsPackage2010/2010DIScrapbook.pdf>

Le curriculum de l'Ontario, 9e et 10e année, Éducation technologique, 2009 (révisé)  
<http://www.edu.gov.on.ca/fre/curriculum/secondary/teched910curr09.pdf>

Le curriculum de l'Ontario, 11e et 12e année, Éducation technologique, 2009 (révisé)  
<http://www.edu.gov.on.ca/fre/curriculum/secondary/2009teched1112curr.pdf>

TMP36 Pinout (Image), 2020 <https://microcontrollerslab.com/tmp36-temperature-sensor/>

Transistor (Image), 2020 <https://www.digikey.com/product-detail/en/n-te-electronics-inc/PN2907A/2368-PN2907A-ND/11655347#images>

Site Web d'OCTE <https://www.octe.ca/fr/resources/resource-folder/tej-3m-4m-computer-engineering-automated-air-exchange-system>