

# **Livrable de projet H (COVID-19): Mise à jour du prototype final**

Groupe FB 8- Pression de Piano

Alexandre Elie, Bamoussa Sacko, Johny Camara, Maxim Busatta, Nicolas Thivierge

Mars 29, 2020

# Table de Matière

<b>1. Introduction</b>	<b>3</b>
<b>2. Mise à jour de l'état du prototype final</b>	<b>3</b>
2.1 État final du prototype	3
2.1.1 Schéma du circuit final	3
2.1.2 Schéma digital du circuit final	4
2.1.3 Code arduino final	4
2.1.4 Logiciel pour graphique	6
<b>2.2 But et fonctionnement</b>	<b>6</b>
2.3 Interaction du client	6
2.4 Vidéo	7
2.5 Essais et analyse	7
<b>3. Plan d'exécution sous condition normale</b>	<b>7</b>
3.1 Description de l'état	7
3.2 Plan d'action	7
3.3 Plan d'essais et analyse	8
<b>4. Conclusion</b>	<b>8</b>
<b>5. Références</b>	<b>8</b>

# 1. Introduction

Malheureusement suite à la pandémie causé par le COVID-19, tous les cours sont maintenant enseigner électroniquement et les espaces de travail ont été fermés. Il est maintenant presque impossible de compléter le projet du cours GNG 1503 présenté par Gilles Comeau. Pour cette raison, le document suivant aura comme but de présenter l'état du prototype final et le plan d'exécution sous conditions normals.

## 2. Mise à jour de l'état du prototype final

### 2.1 État final du prototype

#### 2.1.1 Shéma du circuit final

Ci-dessous se trouve le plan du circuit pour le prototype final.

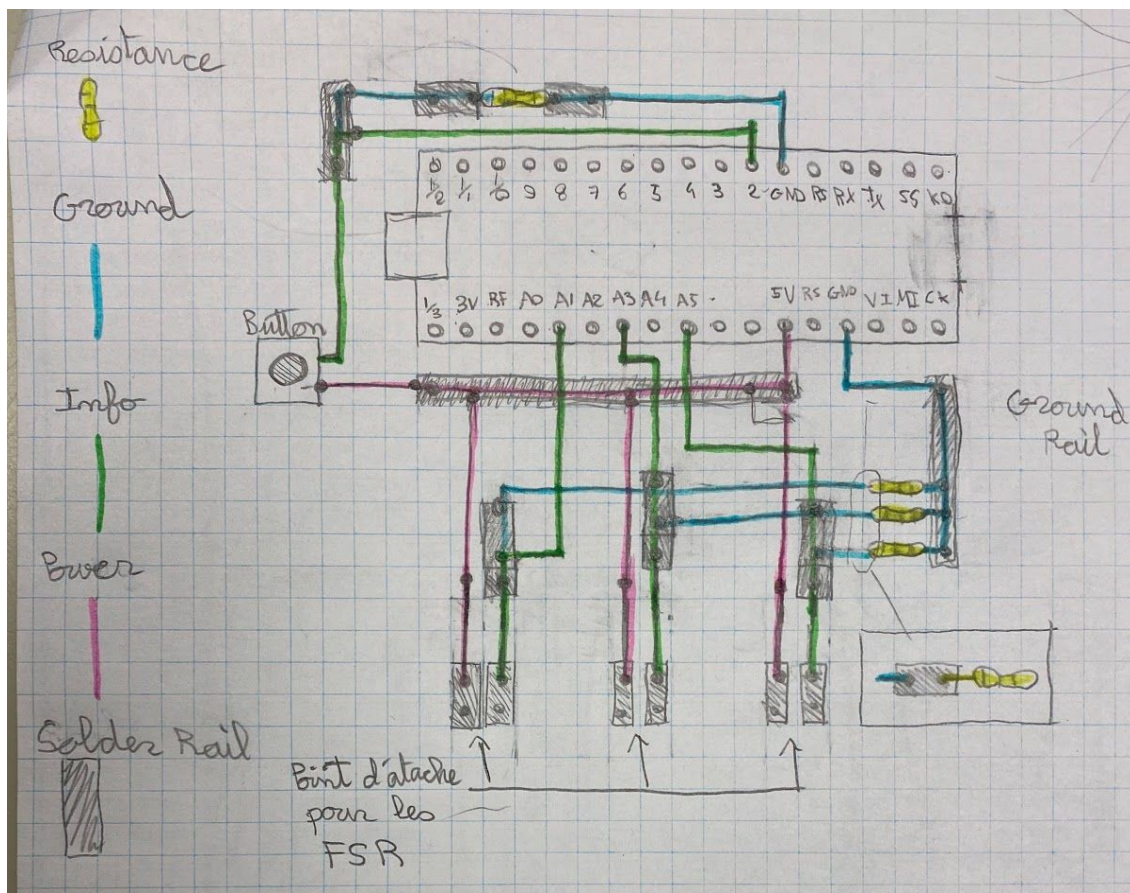


Figure 1: Schéma du circuit sur la perf board.

## 2.1.2 Schéma digital du circuit final

Ci-dessous se trouve un schéma fait avec tinkercad (protoboard pas disponible sur le logiciel).

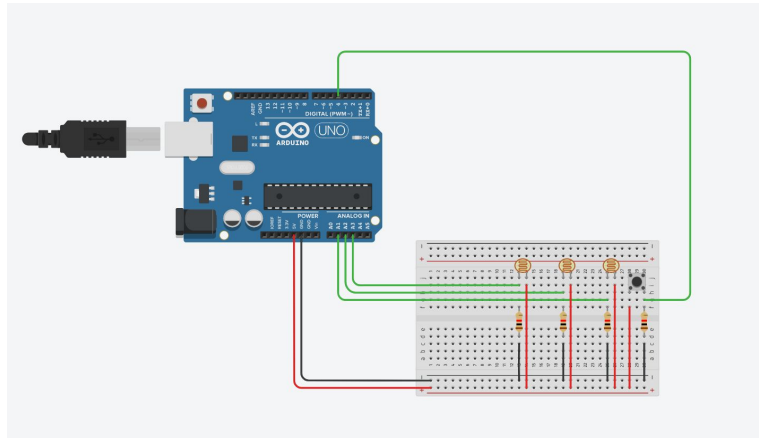


Figure 2: Schéma digital du circuit avec trois capteurs de pression (présenté par des photorésistances) et un bouton on/off fait avec l'aide de tinkercad.

## 2.1.3 Code arduino final

Ci-dessous se trouve le code pour le arduino qui permet de faire la lecture de la pression appliquée sur les senseurs et faire l'envoi des données (en Newtons) au logiciel MegunoLink et au serial moniteur.

```
#include "MegunoLink.h"
#include "ArduinoTimer.h"

ArduinoTimer TableRefreshTimer;
TimePlot MyPlot;

int counter = 0;
int counter2 = 1;
int previousState = 0;
double Temps = 0;
const int bouton = 2;

void setup()
{
  Serial.begin(9600);
  Serial.println("Temps, Sensor1, Sensor2, Sensor3");

  MyPlot.SetTitle("Pression vs. Temps");
  MyPlot.SetXLabel("Temps");
  MyPlot.SetYLabel("Pression");

  MyPlot.SetSeriesProperties("Sensor1", Plot::Blue, Plot::Solid, 2, Plot::Square);
  MyPlot.SetSeriesProperties("Sensor2", Plot::Red, Plot::Solid, 2, Plot::Square);
  MyPlot.SetSeriesProperties("Sensor3", Plot::Green, Plot::Solid, 2, Plot::Square);
}
```

```

void loop()
{
  int boutonstate;
  boutonstate = digitalRead(bouton);

  if(boutonstate > previousState)
  {
    counter2 = counter2+1;
  }

  if ((counter2 % 2) == 0){

double Sensor1, Sensor2, Sensor3, Voltage1, Voltage2, Voltage3, Resistance1, Resistance2, Resistance3, Conductance1, Conductance2, Conductance3;

Sensor1 = analogRead(A1);
Sensor2 = analogRead(A3);
Sensor3 = analogRead(A5);

Voltage1 = map(Sensor1, 0, 1023, 0, 5000);
Voltage2 = map(Sensor1, 0, 1023, 0, 5000);
Voltage3 = map(Sensor1, 0, 1023, 0, 5000);

Resistance1 = 5000 - Voltage1;
Resistance2 = 5000 - Voltage2;
Resistance3 = 5000 - Voltage3;

Resistance1 = Resistance1*10000;
Resistance2 = Resistance2*10000;
Resistance3 = Resistance3*10000;

Resistance1 = Resistance1/Voltage1;
Resistance2 = Resistance2/Voltage2;
Resistance3 = Resistance3/Voltage3;

Conductance1 = 1000000;
Conductance2 = 1000000;
Conductance3 = 1000000;

Conductance1 = Conductance1/Resistance1;
Conductance2 = Conductance2/Resistance2;
Conductance3 = Conductance3/Resistance3;

Sensor1 = Conductance1/80;
Sensor2 = Conductance2/80;
Sensor3 = Conductance3/80;

MyPlot.SendData(F("Sensor1"), Sensor1);
MyPlot.SendData(F("Sensor2"), Sensor2);
MyPlot.SendData(F("Sensor3"), Sensor3);

delay(1);

Temps = Temps + 0.001;

Serial.print(",");
Serial.print(Temps);
Serial.print(",");
Serial.print(Sensor1);
Serial.print(",");
Serial.print(Sensor2);
Serial.print(",");
Serial.println(Sensor3);
}

previousState = boutonstate;
}

```

Figure 3: Capture d'écran du code

## 2.1.4 Logiciel pour graphique

Ci-dessous se trouve le template du logiciel MegunoLink. Au bas se trouve le graphique.

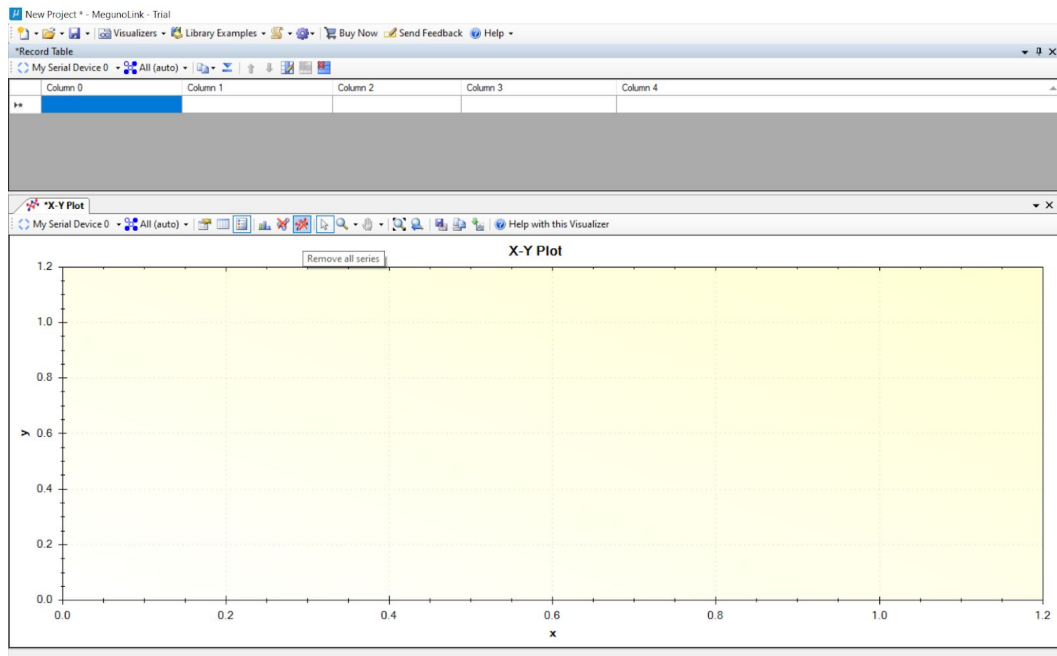


Figure 4: Capture d'écran du format MegunoLink.

## 2.2 But et fonctionnement

Le but et le fonctionnement du prototype final était d'avoir un produit fonctionnel qui permet à l'utilisateur de visualiser en temps réel la pression appliquée sur les clés de piano avec un graphique affiché sur le logiciel MegunoLink et le tableau de données avec Excel, le logiciel Tera Term. L'utilisateur aurait ensuite l'option de sauver le projet pour faire l'analyse des résultats à un autre moment.

## 2.3 Interaction du client

Après l'installation du produit dans le piano, le client n'a qu'à télécharger le logiciel MegunoLink et ouvrir le template du projet que l'équipe a déjà développé. Cela contient le code arduino ainsi que le template pour le graphique. Il doit ensuite confirmer la connection du arduino avec l'ordinateur et le logiciel. Puis pour recevoir les résultats sur Excel. il doit obtenir le logiciel Tera Term et faire la sélection du port. Ensuite, il doit sauver un fichier csv sous la section log du logiciel (le lien pour un vidéo d'instruction se trouve dans la Référence). Après, il n'a qu'à appuyer sur le bouton ON/OFF et jouer au piano.

## 2.4 Vidéo

Aucun vidéo du prototype final n'a pu être filmé pour plusieurs raisons.

## 2.5 Essais et analyse

Puisque nous étions pas en mesure de finaliser le prototype, un essai final du fonctionnement du prototype n'était pas possible. Par contre, nous avons pu faire des essais du code, qui inclut l'utilisation du logiciel MegunoLink. Durant les essais le logiciel a pu interpréter les données reçues par le arduino et de les afficher avec un graphique. Les données qui ont été affichées par le logiciel ont satisfait le groupe. Cela était la seule composante qui a pu être vérifié donc il y aucune nécessité d'utiliser un tableau qui démontre les résultats attendus vs. les résultats reçus puisqu'il a seulement un résultat reçu.

# 3. Plan d'exécution sous condition normale

## 3.1 Description de l'état

Notre prototype était en partie terminé et il restait quelques tâches nécessaires pour le finaliser, en particulier le soudage du circuit dessiné (voir 2.1.1). Par contre, nous étions en mesure de finaliser le code (voir 2.1.3) qui inclut la lecture et conversion en newton de la pression, l'affichage du graphique et du tableau de données (voir 2.1.4).

## 3.2 Plan d'action

Le soudage du circuit aurait été complété la journée du 17 et le code aurait été le complété entre le 16 et 19. L'assemblage du prototype aurait suivi et le produit aurait été présenté au client et ensuite toutes modifications nécessaires auraient suivies.

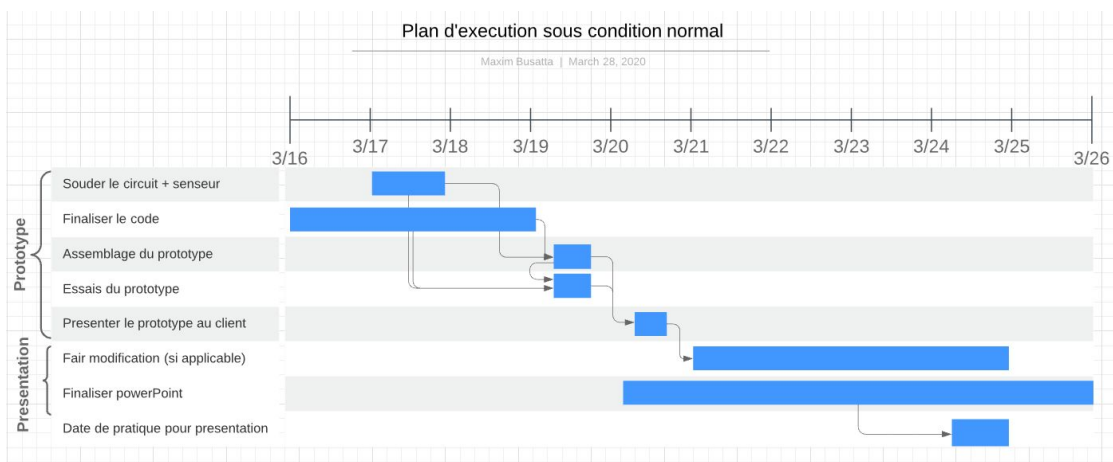


Figure 5: Le plan d'action pour finaliser le projet (diagramme Gantt).

### 3.3 Plan d'essais et analyse

Notre plan d'essais aurait consisté principalement de faire l'essai du prototype final avec un piano qui aurait été procuré par le cours. Cela nous permettrait de vérifier chaque aspect de notre prototype (le code, le circuit et les dimensions). Une rencontre avec le client aurait aussi été organisée pour lui présenter le prototype final et son fonctionnement.

## 4. Conclusion

Pour conclure, notre projet pour le cours GNG 1503 n'a pas pu être complété dû à la pandémie du covid-19. Le document ci-dessus a pu présenter les tâches du prototype final que nous avons réalisées et non réalisées avant la fermeture de l'université ainsi que le plan d'exécution sous condition normale. Nous avons par contre pu compléter le code final qui permet d'afficher nos données reçues du arduino avec le logiciel MegunoLink, Tera Term et Excel. Nous sommes confiants que nous aurions pu terminer notre prototype final (complètement fonctionnel) si les espaces de travail (Maker space...) n'auraient pas fermés. La fermeture des espaces de travail est la raison principale pourquoi le prototype n'a pas pu être complété.

## 5. Références

MegunoLink, "GET MORE FROM YOUR ARDUINO". <https://www.megunolink.com/>

Tera Term, "Free download".

[https://download.cnet.com/Tera-Term/3000-2094\\_4-75766675.html](https://download.cnet.com/Tera-Term/3000-2094_4-75766675.html)

Network of Excellence Robotic & Mechatronics, "Force sensor".

<https://home.roboticlab.eu/en/examples/sensor/force#theory>

YouTube, "Arduino Tutorial 26# How to take Arduino sensor data to Excel sheet in windows 10/8/7". <https://www.youtube.com/watch?v=T5g5U3JbmQc>

GitHub, "Arduino library for sending data to MegunoLink visualisers and useful components".

<https://github.com/Megunolink/MLP>