

GNG 1503

Livrable G : Prototype II et rétroaction du client

Par

Abdoul Abdoulaye

Philippe Plante

Franck Sokoudjou



uOttawa

Université d'Ottawa - University of Ottawa

© Ottawa, Canada, 14 mars 2021

Introduction

Dans un problème de conception, le prototype est utilisé pour apprendre davantage ou mieux comprendre le problème; communiquer avec le client et obtenir une rétroaction; réduire le risque associé à un aspect particulier ou mesurer les performances par des essais. Cependant un prototype en lui-même ne constitue pas une source suffisante de satisfaction ou ne peut pas servir d'indicateur pour un concepteur car après sa confection, le prototype doit subir des essais suivant un plan de test bien précis, en d'autres termes le prototype et les essais sont deux parties indissociables et intégrantes dans une conception.

Ainsi, si le dernier livrable consistait à mettre au point le premier prototype du concept final, prototype qui était non fonctionnel et donnait une vue globale du concept, dans ce livrable il sera question de développer un prototype plus avancé. Pour ce faire, l'équipe va tout d'abord évaluer à établir un plan d'essai, puis par la suite produire un modèle 3D du concept et trouver ou écrire le code source de chaque composante électronique. L'équipe fera aussi part de la rétroaction du client ainsi qu'une analyse objective sur le nouveau prototype.

Plan de test

Les tests que l'équipe perçoit essayer pour le développement du prototype deux est strictement basée sur le fonctionnement des logiciels utilisés dans le concept. Pour ce faire, l'équipe vise à faire fonctionner chaque composant individuellement ainsi que leur code attribué pour enfin les connecter en un circuit inter-intégré. Chaque test sera important d'être fait avec précision et efficacité pour atteindre les objectifs par rapport aux critères et besoins du client. Bref, le prototype deux permet à l'équipe de construire le code ainsi de faire fonctionner chacun des logiciels utilisés dans notre prototype final.

Tableau : Plan de test

N° de Test	Objectif du test (Pourquoi)	Description du Prototype Utilisé et de la Méthode de Test de Base (Quoi)	Description des Résultats à Documenter et Comment ces Résultats seront Utilisés (Comment)	Durée Estimée du Test et Date Prévues du Début du Test (Quand)
1	<i>Trouver le code pour le prototype pour chacun des composants</i> <ul style="list-style-type: none">- Ce test nous permet de mieux comprendre les logiciels comme tel	<i>Prototype ciblé</i> <ul style="list-style-type: none">- Nous allons tester chaque code avec l'aide du logiciel Arduino dans l'ordinateur ainsi que leur bibliothèque.	Les informations que l'équipe mesure est le code final de chacun des codes. Les codes doivent obtenir de bons résultats pour qu'ils puissent bien passer à la prochaine	<i>Début: 7 mars Fin: 14 mars Durée: 2 jours et demi Dépendance: Aucune</i>

	<ul style="list-style-type: none"> - Les codes arduino de chacun des logiciels est important pour le fonctionnement de notre concept. - Chaque code se comporte différemment par rapport à son objectif qu'il veut faire. - Les codes doivent être précis pour qu'ils retournent les bonnes valeurs par rapport aux critères demandés. - La raison du test, est de trouver le meilleur code pour chacun des composant utilisée dans le logiciel 	<ul style="list-style-type: none"> - Aucun coût pour ce test. - En utilisant les codes de bases trouvés en ligne, l'équipe veut les modifier en fonction de ce que le composant est supposé faire dans le prototype. - L'équipe veut être en mesure de développer un code pour chaque composant utilisé. Soit le lcd, ultrasonic, switch(on/off), potentiel mètre et laser. - Chaque composant est unique avec une utilité différente dans le prototype et donc, le code doit s'ajuster par rapport aux besoins et critères demandés. - De la recherche ainsi que la méthode d'essais erreurs est utilisé comme travail effectué. 	<p>étape. Pour bien les enregistrer, un membre de l'équipe les enregistra sur un fichier dans leur ordinateur du logiciel Arduino dans l'ordinateur. Les données ramassées devraient être les 5 codes des 5 composants utilisés dans le logiciel ainsi qu'un code final permettant de connecter chacun de ses composants en un circuit inter-intégré. Ces résultats sont important en prendre compte puisqu'ils nous aidera en équipe de mieux comprendre le comportement que le prototype prend ainsi qu'un avancement pour l'utilisation de ces composants dans le concept.</p> <p><u>Critère d'arrêt:</u> L'équipe arrêtera le test lorsque les codes retourneront les bonnes valeurs par rapport à leur fonction et objectifs.</p>	<p>Le test s'intègre dans le fonctionnement des codes de notre prototype et projet.</p>
<p>2</p>	<p><i>Tester chacun des composant du logiciel avec leur code associé</i></p> <ul style="list-style-type: none"> - Après avoir trouvé les codes, l'équipe teste maintenant chacun des logiciels avec leurs codes. - Les tests analyseront la fiabilité de notre logiciel dans notre prototype. - Le test est dédié à l'apprentissage de notre prototype. - La raison du test est de tester le code pour les logiciels pour s'assurer qu'il fonctionne correctement dans le 	<p><i>Prototype compréhensifs</i></p> <ul style="list-style-type: none"> - En utilisant les codes obtenus dans le test précédent, l'équipe veut maintenant tester les codes avec les logiciels appropriés pour comprendre leur fonctionnement dans le prototype. - Aucun coût pour ce test (Les logiciels et pièces sont déjà achetés par rapport à notre budget.) - Ce test nous permettra de comprendre la fiabilité du système de chaque pièce, lorsqu'il est en action. - Nous voulons que chaque pièce fonctionne 	<p>Après avoir trouvé le code, l'équipe teste chacun des composants avec leur code associé pour s'assurer que les pièces fonctionnent correctement lorsque le système est activé. Les information qui sont mesurée pour ce test est;</p> <ul style="list-style-type: none"> - Lcd: Le lcd lit les deux angles mesurés - Ultrasonic: permet de lire une distance verticale de 100 mm et le transmettre en une angle d'inclinaison dans le lcd - Potentiomètre: En lisant sa résistance interne, le potar retourne l'angle de direction par rapport à sa résistance/position. - Bouton: Le bouton ferme 	<p><i>Début:</i> 7 mars <i>Fin:</i> 14 mars <i>Durée:</i> 3 jour <i>Dépendance:</i> Test 1 a une dépendance sur le Test 2</p> <p>Le test s'intègre dans le fonctionnement des logiciels de notre prototype et projet.</p>

	prototype.	<p>correctement, donc un ajustement de code peut être nécessaire à cette étape.</p> <ul style="list-style-type: none"> - Chaque test est fait par un breadboard permettant de mesurer et tester chacun des circuits avec l'aide du MyDaq National Instrument. En utilisant ainsi l'arduino et son logiciel dans l'ordinateur avec le code implanté dans le système, l'équipe est en fonction de tester le logiciel et analyser les résultats. - La raison du test est de s'assurer que chaque code fonctionne correctement par rapport à la fonction que le composant est supposé de faire. 	<p>et allume complètement le circuit pour sauver le pouvoir dans la batterie.</p> <ul style="list-style-type: none"> - Laser: Le laser émet un light beam dans la position verticale pour que l'utilisateur soit en fonction de s'orienter avec la tige lorsqu'il manœuvre le bâton cylindrique. <p>Pour enregistrer les résultats, l'équipe décide de prendre des vidéos pour montrer visuellement le fonctionnement de chacun des tests. Ces données sont importantes à prendre en compte puisqu'elles doivent respecter les critères demandés par le client et retourner les bonnes mesures pour le concept final.</p> <p><u>Critère d'arrêt:</u> L'équipe arrêtera le test lorsque les logiciels retourneront les bonnes valeurs par rapport à leur fonction et objectifs.</p>	
3	<p><i>Tester le circuit en un circuit inter-intégré (i2c)</i></p> <ul style="list-style-type: none"> - L'équipe choisit de construire leur circuit pour leur logiciel en un circuit inter-intégré qui est simple et pratique à utiliser. - Ce test est la dernière étape pour que notre prototype puisse fonctionner proprement avec le logiciel enregistré à l'intérieur. - Ce test est dédié à construire un circuit final avec chacun des composant utilisé dans le logiciel pour notre concept final 	<p><i>Prototype compréhensifs</i></p> <ul style="list-style-type: none"> - Cet dernière étape consiste à tester chaque composant en un circuit inter-intégré (i2c) - Aucun coût pour ce test - Un i2c est un composant électronique reproduisant une ou plusieurs fonctions électroniques plus ou moins complexes, intégrant souvent plusieurs types de composants électroniques de base dans un volume réduit, rendant le circuit facile à mettre en œuvre. - Un ordinateur avec un logiciel Arduino ainsi que les composants du 	<p>Les informations qui sont mesurées pour ce test sont les mêmes que le test précédent. Par contre, le tout doit connecter maintenant en un circuit inter-intégré (i2c) pour être en fonction de retourner les mêmes résultats et être placé dans leur bonne position dans le prototype. Pour documenter les résultats, l'équipe décide de prendre des vidéos démontrant le fonctionnement de notre circuit i2c. Le code peut également être modifié pour cette étape pour faire fonctionner le circuit. C'est donner son importance puisqu'il seront les résultats finale de notre prototype. Pour que le prototype fonctionne, le concept doit être en fonction de connecter chaque composant du logiciel en un circuit.</p>	<p><i>Début:</i> 7 mars <i>Fin:</i> 14 mars <i>Durée:</i> 2 jour et demi <i>Dépendance:</i> Test 2 dépendance sur le Test 3</p> <p>Le test s'intègre dans le fonctionnement du circuit de notre prototype et projet.</p>

<ul style="list-style-type: none"> - La raison du test est de faire certain que le circuit i2c fonctionne correctement lorsqu'il est utilisé par l'utilisateur. 	<p>logiciel seront utilisés comme matériel pour ce test.</p> <ul style="list-style-type: none"> - Le travail effectué est d'ajuster chacun des composants dans sa position actuelle dans le prototype pour tester la fiabilité du circuit i2c comme système. - La raison du test est de connecter chacun des composants de notre logiciel en un circuit simple et non complexe pour rendre la facilité d'utilisation pour notre client. 	<p><u>Critère d'arrêt:</u> Lorsque chaque composant fonctionne correctement lorsqu'il est connecté en un circuit inter-intégré, l'équipe arrêtera le test.</p>	
--	---	--	--

Prototype

Pour le prototype deux, l'équipe démontre le concept en utilisant le logiciel Onshape qui est une plateforme de développement de produits "Software-as-a-Service" qui combine, CAO, des gestions de données intégrées, des outils de collaboration en temps réel et une analyse commerciale. Les membres illustrent le fonctionnement de chacun des composants ainsi qu'une base pour leur code Arduino utilisée. L'équipe trouvera le code final du prototype 2 avec l'aide des tests. Bref, l'équipe se concentre à bien faire fonctionner le logiciel du prototype pour être en mesure de bien calculer les angles pour le client.

En se basant sur le prototype 1, nous avons encore une boîte de forme d'un demi-cercle incluant une tige démontrant la balle de fusil. Nous avons un bâton cylindrique qui fait une rotation de 0 à 180 degré dans la boîte. Sur la surface du bâton, un ultrasonic ainsi qu'un laser est fixé sur le dessus pointant vers le haut verticalement. Le bâton est attaché sur le bout d'un potentiel mètre qui permet au mouvement rotationnel du bâton. Un levier triangulaire est également attaché à l'extrémité du bâton pour permettre accès au mouvement. Sur la surface du haut du demi-cercle, un LCD display ainsi qu'une bulle de niveau est à la disposition du client. Un bouton pour allumer et éteindre le circuit est situé sur le côté bas droit de la boîte. La source d'alimentation ainsi que le microcontrôleur est situé à l'intérieur de la boîte vers la gauche.

Ci-dessous, le concept est présenté sous format onshape pour mieux présenter le concept voulu dans le prototype 2. Le matériel utilisé est du bois MDF ainsi que la partie vide est composé d'une feuille acrylique.

Figure 1 : Prototype II

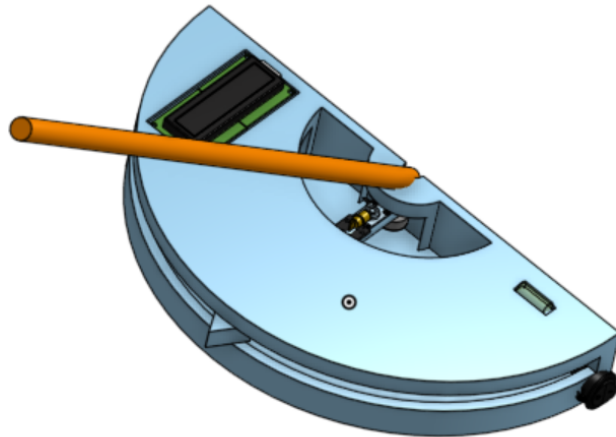


Figure 2 : Prototype II

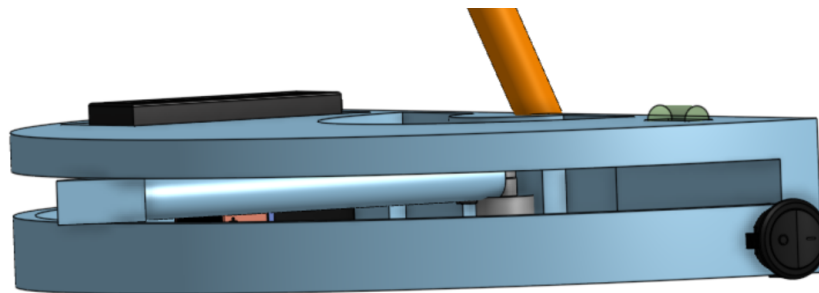


Figure 3 : Prototype II

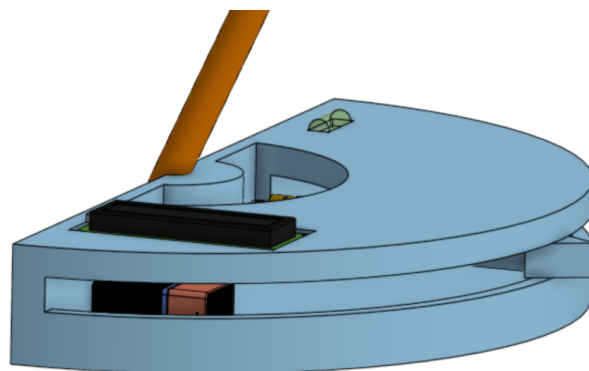
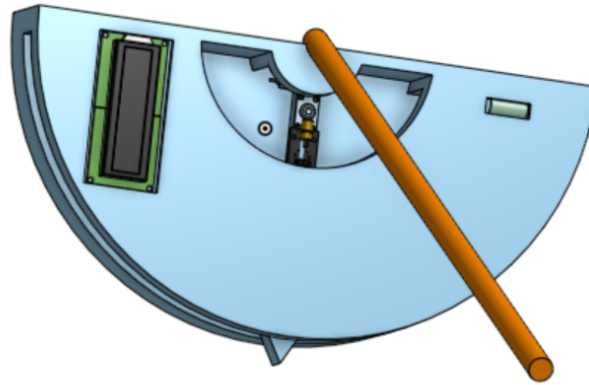


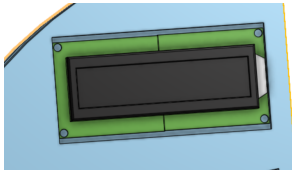
Figure 4 : Prototype II



Onshape

Les logiciels ainsi qu'une base de leur code et fonctionnement:

1. LCD Display



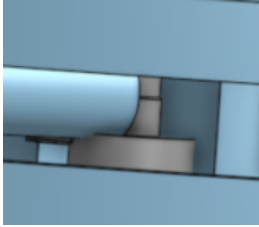
```
#include <Wire.h> // Library for I2C communication
#include <LiquidCrystal_I2C.h> // Library for LCD
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);
theta = atan(double(y/x))
x = 1.25
y = c1 // c1 is the value that the ultrasonic distance sensor will read
phi = c2 // the angle measured by the potentiometer due to its position

void setup() {
  // Initiate the LCD:
  lcd.init();
  lcd.backlight();
}

void loop() {
  lcd.setCursor(2, 0); // Set the cursor on the third column and first row.
  lcd.print("Angle 1", " ", theta); // Print the string "Hello World!"
  lcd.setCursor(2, 1); //Set the cursor on the third column and the second row (counting starts at 0!)
  lcd.print("Angle 2", " ", phi);
}
```

Le LCD est connecté directement par i2c communication dans le code Arduino et circuit. Le LCD lit directement les deux angles avec l'aide du ultrasonique et du potentiel mètre. Le client peut facilement lire c'est valeur du à son placement sur le demi-cercle.

2. Potentiomètre



Lorsqu'on bouge le bâton cylindrique on bouge ainsi le potentiomètre et donc change sa résistance. Nous voulons que le code lit cette résistance en forme d'angle de 0 à 180 degré et ensuite le transmettre sur le lcd.

```
#define ROTARY_ANGLE_SENSOR A0 //Use analog pin A0 for the Rotary Angle Sensor
#define B ADC_REFB 3.3 //Reference voltage of ADC is 3.3v
#define FULL_ANGLEB 300.0 //Full value of the rotary angle is 300 degrees

void setup()
{
  //Start the serial connection
  Serial.begin(9600);
}

void loop()
{
  //Read the value of the rotary angle sensor in degrees
  int degrees = getDegrees();

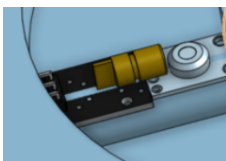
  //Output it to the serial monitor
  Serial.print("The angle between the mark and the start position: ");
  Serial.println(degrees);

  //The delay between readings
  delay(500);
}

int getDegrees()
{
  //Read the raw sensor value
  int sensor_value = analogRead(ROTARY_ANGLE_SENSOR);

  //Convert the sensor reading to degrees and return that value
  float voltage = (float)sensor_value * ADC_REF / 1023;
  float degrees = (voltage * FULL_ANGLE) / ADC_REF;
  return degrees;
}
```

3. Laser



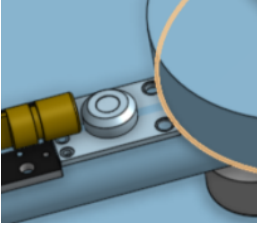
Le laser permet à l'utilisateur de bien orienter le ultrasonic avec la tige lorsqu'il bouge le bâton le cylindrique. Nous voulons donc un code qui permet au

```
int laserPin = 10;
void setup() {
  pinMode(laserPin, OUTPUT);
}

void loop() {
  digitalWrite(laserPin, HIGH);
  delay(100);
  digitalWrite(laserPin, LOW);
  delay(100);
}
```


laser de rester allumé jusqu'à temps que le circuit est fermé.

4. Ultrasonic



```
#include <Wire.h>
#include <SparkFun_VL6180X.h>
#define VL6180X_ADDRESS 0x29

VL6180xIdentification identification;
VL6180x sensor(VL6180X_ADDRESS);

void setup() {
  Serial.begin(115200); //Start Serial at 115200bps
  Wire.begin(); //Start I2C library
  delay(100); // delay .1s

  sensor.getIdentification(&identification); // Retrieve manufacture info from device memory
  printIdentification(&identification); // Helper function to print all the Module information

  if(sensor.VL6180xInit() != 0){
    Serial.println("FAILED TO INITIALIZE"); //Initialize device and check for errors
  };

  sensor.VL6180xDefaultSettings(); //Load default settings to get started.

  delay(1000); // delay 1s
}

void loop() {
  //Get Ambient Light level and report in LUX
  Serial.print("Ambient Light Level (Lux) = ");

  //Input GAIN for light levels,
  // GAIN_20 // Actual ALS Gain of 20
  // GAIN_10 // Actual ALS Gain of 10.32
  // GAIN_5 // Actual ALS Gain of 5.21
  // GAIN_2_5 // Actual ALS Gain of 2.60
  // GAIN_1_67 // Actual ALS Gain of 1.72
  // GAIN_1_25 // Actual ALS Gain of 1.28
  // GAIN_1 // Actual ALS Gain of 1.01
  // GAIN_40 // Actual ALS Gain of 40

  Serial.println( sensor.getAmbientLight(GAIN_1) );

  //Get Distance and report in mm
  Serial.print("Distance measured (mm) = ");
  Serial.println( sensor.getDistance() );
}
```

```

//Get Distance and report in mm
Serial.print("Distance measured (mm) = ");
Serial.println( sensor.getDistance() );

delay(500);
};

void printIdentification(struct VL6180xIdentification *temp){
Serial.print("Model ID = ");
Serial.println(temp->idModel);

Serial.print("Model Rev = ");
Serial.print(temp->idModelRevMajor);
Serial.print(".");
Serial.println(temp->idModelRevMinor);

Serial.print("Module Rev = ");
Serial.print(temp->idModuleRevMajor);
Serial.print(".");
Serial.println(temp->idModuleRevMinor);

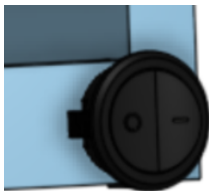
Serial.print("Manufacture Date = ");
Serial.print((temp->idDate >> 3) & 0x001F);
Serial.print("/");
Serial.print((temp->idDate >> 8) & 0x000F);
Serial.print("/");
Serial.print((temp->idDate >> 12) & 0x000F);
Serial.print(" Phase: ");
Serial.println(temp->idDate & 0x0007);

Serial.print("Manufacture Time (s)= ");
Serial.println(temp->idTime * 2);
Serial.println();
Serial.println();
}

```

Le ultrasonic lit la distance de la tige, en connaissant la position en x du ultrasonic et la hauteur de la tige y, nous pouvons retrouver l'angle d'inclinaison qui print sur le lcd display.

5. Rocker Switch



Le code ci-dessus est fermer une LED on and off ayant un button qui ferme et

```

int pin_LED = 10;
int pin_switch = 2;
void setup() {
Serial.begin(9600);
Serial.print("Sketch: "); Serial.println(__FILE__);
Serial.print("Uploaded: "); Serial.println(__DATE__);
Serial.println(" ");

pinMode(pin_LED, OUTPUT);
digitalWrite(pin_LED, LOW);

pinMode(pin_switch, INPUT)
}
void loop() {
if ( digitalRead(pin_switch) == HIGH)
{
digitalWrite(pin_LED, HIGH);
}
else
{

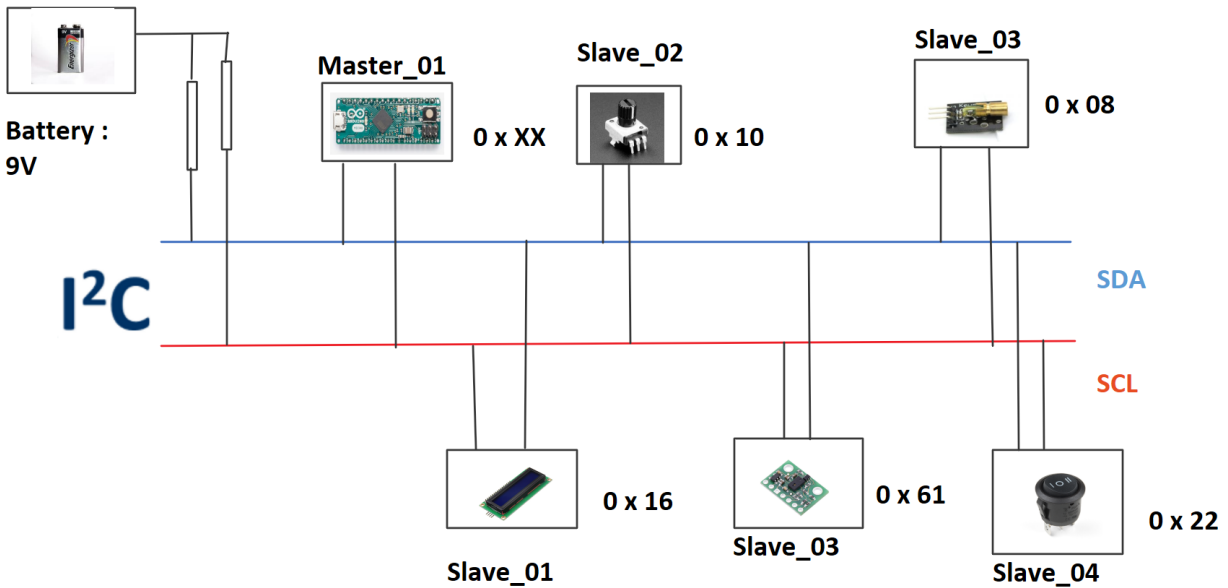
```

pour

allume. L'équipe modifie le code pour qu'il soit en fonction de fermer le courant au complet ainsi de le rallumer en fonction du mouvement du bouton.

La figure suivante démontre le système I2C en schéma. On peut voir que chaque composant est connecté directement sur deux fils soit SDA et SCL qui retournent à la source d'alimentation (Arduino) avec l'aide de la source d'alimentation. Le circuit I2C simplifié et facilite notre circuit et est facile à manœuvrer pour notre concept. Le code doit aussi être modifié pour donner accès à notre circuit d'utiliser ce système.

Figure 5 : I2C système



Mise à jour planification des tâches

Le tableau ci-dessous représente une mise à jour des plan de tâches pour chaque membres pour que l'équipe reste organisée tout au long du projet. Chaque tâche est spécifiée par rapport au test du prototype 2, ainsi que certaines tâches prévues à faire du prototype 3. La date et qui prendra la responsabilité ainsi que certaines dépendances si nécessaire seront associées à ces tâches.

Certaines tâches mineures que l'équipe prépare en avance par rapport au prototype 3 est le dessin Inkscape pour le découpage de laser du prototype avec mdf et acrylique. L'équipe planifie aussi les dernières ajustation nécessaires par rapport à la rétroaction du prototype 2 et des tests pour développer notre concept final.

Tableau : mise à jour du plan de projet

Tâches	Responsabilité	Dépendance	Date
Test 1	Abdoul	<i>Le test 1 doit être</i>	Début: 7 mars 2021

	Franck	<i>complété avant que le test 2 peut poursuivre</i>	Fin: 14 mars 2021 Durée du test: 2 jour et demi
Test 2	Philippe	<i>Le test 2 doit être complété avant que le test 3 peut poursuivre</i>	Début: 7 mars 2021 Fin: 14 mars 2021 Durée du test: 3 jour
Test 3	Abdoul Franck Philippe	<i>Seulement le test 2 dépend du test 3</i>	Début: 7 mars 2021 Fin: 14 mars 2021 Durée du test: 3 jour
Prototype 3	Philippe	<i>Aucune</i>	Début: 11 mars 2021 Fin: 21 mars 2021 Durée du test: 5 jour

Rétroaction et commentaires

Rétroaction :

La dernière rencontre avec le client a engendré quelques changements sur la planification des tâches. L'équipe fera plus de tests pour vérifier la fiabilité du prototype sur différents types de surfaces d'essai afin d'évaluer la fiabilité ainsi que la performance du prototype.

Commentaires :

Pour la mise au point de ce prototype quelques difficultés furent rencontrées pour trouver ou écrire une partie ou la totalité du code de chaque composante électronique. Ces codes seront revus et modifiés au besoin après les tests 1 et 2. Le laser peut se voir changer de place si les tests à venir confirment que dans certaines situations il n'arrive pas à être positionné directement au-dessous de la tige pour pouvoir l'atteindre.

Conclusion

En conclusion, dans ce livrable G il était question d'élaborer un second plan d'essai de prototypage et de développer notre prototype 2 en nous servant de la rétroaction que nous avons eu du client lors de notre présentation du prototype 1. Premièrement nous avons développé un prototype qui sera utilisé pour atteindre les objectifs comme décrit dans notre plan (c.-à-d.

répondre aux questions “pourquoi”, le “quoi” et le “quand” du prototypage). Ensuite nous avons inclus un modèle analytique, numérique ou expérimental de notre prototype. La rétroaction et les commentaires donnés globalisent ce que notre groupe a eu à modifier ou améliorer sur le prototype après notre rencontre avec le client qui était la présentation du prototype 1.

Dans le prochain livrable, il sera question d’élaborer un plan d’essai et de développer notre prototype 3 qui devra être une version complètement fonctionnelle de notre solution.

Wrike Snapshot

[Snapshot](#)