

Livrable de projet G: Prototype 2

Soumis par:

FA3, groupe 4

Moïse Batotele, 300124326

Jean-Marie Kabulekedi Kapiamba, 300109647

Haitam Nezhari, 300092858

Lyazid Sikouk, 300117380

5 novembre 2020

Université d'Ottawa

Table des matières

Table des matières	1
1 Introduction	2
2 Rétroaction des clients.....	3
3 Hypothèses.....	4
4 Conception détaillée.....	5-9
Prototype physique-logiciel.....	5-9
Analyse et Essais.....	10-11
5 Conclusion.....	12
6 Références.....	13

Introduction

Dans le cadre du cours de GNG 2501, nous poursuivons notre processus de conception d'un système facilitant la réanimation cardio-respiratoire en milieu médical. Suite au développement d'un premier prototype physique non-fonctionnel, le client nous fait part de ses inquiétudes concernant la direction dans laquelle le projet se dirige. En conséquent, en tant qu'équipe de conception, nous considérons avec extrême délicatesse toutes les angoisses que présente le client et, avec une grande souplesse, ajustons notre vision et concentrons nos efforts afin de satisfaire le chaland.

Ainsi, au cours de ce livrable, nous allons mettre en évidence la rétroaction que nous avons obtenu du client, présenter notre second prototype et faire la mise à jour de notre plan de projet.

Rétroaction du client

Suite à la troisième rencontre avec les clients, de grands changements ont été suggérés en termes d'analyse de données. En effet, Le système d'analyse de respiration du premier prototype consistait à interpréter les battements cardiaques par minute (bpm) d'un patient afin de faire un lien direct avec sa respiration et ultimement, déterminer une fréquence de compression adéquate de l'ambu bag.

Toutefois, la discussion ayant eu lieu le lundi dernier a remis en question l'efficacité d'un tel système. Les clients mentionnent la complexité de notre approche et suggèrent une démarche beaucoup plus réaliste. Ces derniers recommandent toujours l'utilisation d'un capteur de battements cardiaques, mais cette fois-ci déconseillent la conversion de bpm à rpm (respiration par minutes). Ils expliquent que lors de l'utilisation, lorsqu'une vie humaine en dépend, si la conversion de données n'est pas faite en temps réel, les résultats pourraient être fatales.

C'est ainsi que nous développons notre second prototype, qui fournit un aperçu du fonctionnement de chaque élément de notre produit. Celui-ci décrit la co-dépendance de chaque sous-système ainsi que le processus d'interprétation et d'analyse des données.

Il est à noter qu'en raison de la pandémie (COVID-19), les temps de livraison des matériaux sont étendus. Nous ne sommes donc pas à mesure de compléter le deuxième prototype avec un capteur de battements cardiaque comme nous l'aurions souhaité. En amenant cet aspect à l'attention des clients, ceux-ci proposent l'utilisation temporaire d'un capteur de pression afin de suppléer notre lacune de matériel.

Hypothèses

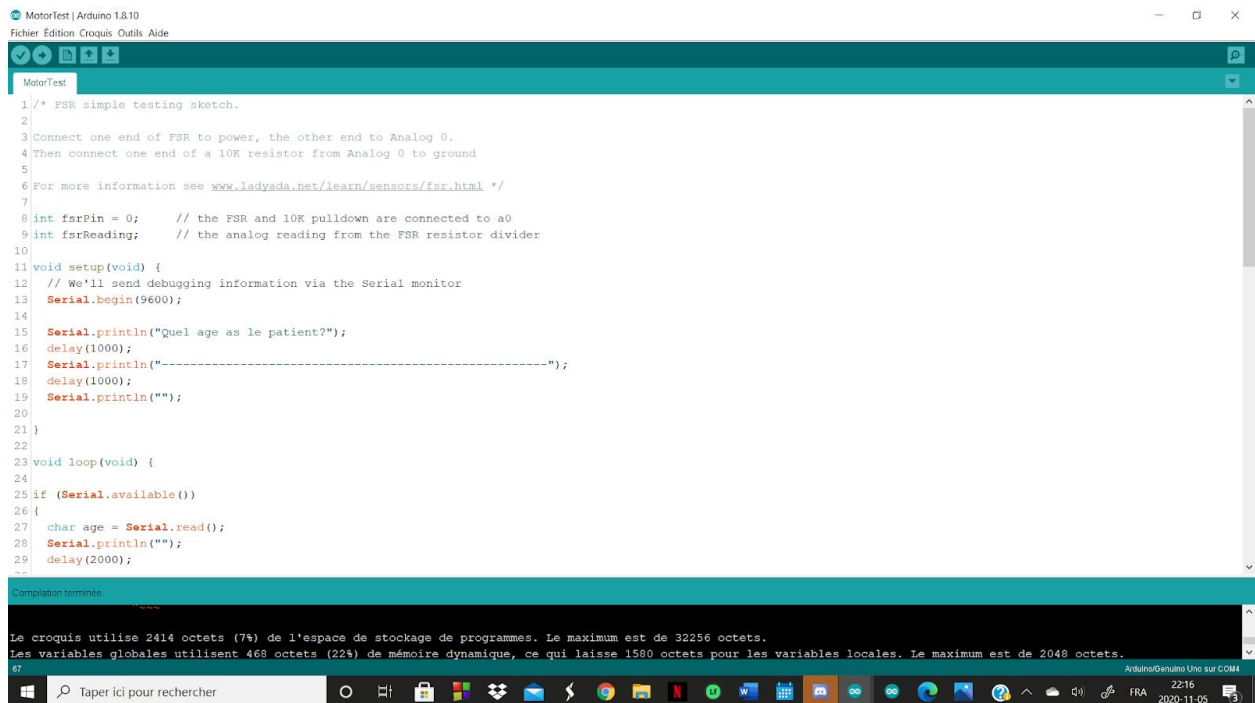
Dans notre cheminement vers le développement de notre second prototype, nous avons cultivé de nouvelles hypothèses et réévalué celles que nous avons développées lors de la création du premier prototype. Ainsi, nous avons pu former les hypothèses suivantes à l'égard de la conception du dispositif:

- Le code du prototype sera fonctionnel
- L'affichage des données aura une excellente fiabilité
- Le poids du dispositif ne sera pas très élevé
- Les dimensions du prototypes seront appropriées
- Le prix du produit ne sera pas élevé.
- L'esthétique du prototype ne sera pas exceptionnel.
- Le prototype aura une bonne mobilité
- Le prototype ne devrait pas prendre trop de temps à être installé
- Le prototype sera très sécuritaire
- La fiabilité du produit sera impeccable

Conception détaillé

Prototype physique et logiciel

Le deuxième prototype analyse la pression qu'un patient, qui a une irrégularité respiratoire, applique sur un capteur et en déduit si la réanimation cardio-pulmonaire est nécessaire. Suite à ceci, dépendamment du groupe d'âge dans lequel le patient se situe, un message sur L'écran LCD affiche si le patient hyperventile, hypoventile ou si la respiration est régularisé. L'utilisateur auras donc une approche respective à chaque cas.



```
MotorTest | Arduino 1.8.10
Fichier Édition Croquis Outils Aide

MotorTest
1 /* FSR simple testing sketch.
2
3 Connect one end of FSR to power, the other end to Analog 0.
4 Then connect one end of a 10K resistor from Analog 0 to ground
5
6 For more information see www.ladyada.net/learn/sensors/fsr.html */
7
8 int fsrPin = 0; // the FSR and 10K pulldown are connected to a0
9 int fsrReading; // the analog reading from the FSR resistor divider
10
11 void setup(void) {
12 // We'll send debugging information via the Serial monitor
13 Serial.begin(9600);
14
15 Serial.println("Quel age as le patient?");
16 delay(1000);
17 Serial.println("-----");
18 delay(1000);
19 Serial.println("");
20
21 }
22
23 void loop(void) {
24
25 if (Serial.available())
26 {
27 char age = Serial.read();
28 Serial.println("");
29 delay(2000);
30
31 }
32 }
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Compilation terminée

Le croquis utilise 2414 octets (7%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 468 octets (22%) de mémoire dynamique, ce qui laisse 1580 octets pour les variables locales. Le maximum est de 2048 octets.

67
Taper ici pour rechercher
Arduino/Genuino Uno sur COM4
22:16
2020-11-05
```

```
MotorTest
1 // -----
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31 fsrReading = analogRead(fsrPin);
32
33 Serial.print("Analog reading = ");
34 Serial.print(fsrReading); // the raw analog reading
35
36 if (age < '18')
37 {
38
39 // We'll have a few thresholds, qualitatively determined
40 if (fsrReading < 10) {
41 Serial.println(" - Le patient hypoventille niveau 3");
42 } else if (fsrReading < 100) {
43 Serial.println(" - Le patient hypoventille niveau 2");
44 } else if (fsrReading < 300) {
45 Serial.println(" - Le patient hypoventille niveau 1");
46 } else if (fsrReading < 700) {
47 Serial.println(" - La Respiration du patient est regulariser");
48 } else {
49 Serial.println(" - Le patient Hyperventille");
50 }
51 } else if (age < '60')
52 {
53 if (fsrReading < 10) {
54 Serial.println(" - Le patient hypoventille niveau 3");
55 } else if (fsrReading < 50) {
56 Serial.println(" - Le patient hypoventille niveau 2");
57 } else if (fsrReading < 200) {
58 Serial.println(" - Le patient hypoventille niveau 1");
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
```

Compilation terminée

Le croquis utilise 2414 octets (7%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 468 octets (22%) de mémoire dynamique, ce qui laisse 1580 octets pour les variables locales. Le maximum est de 2048 octets.

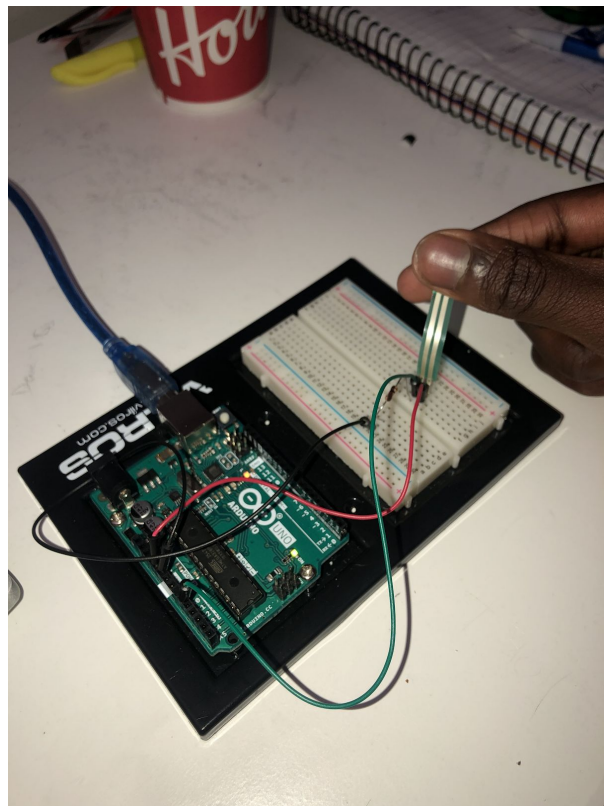
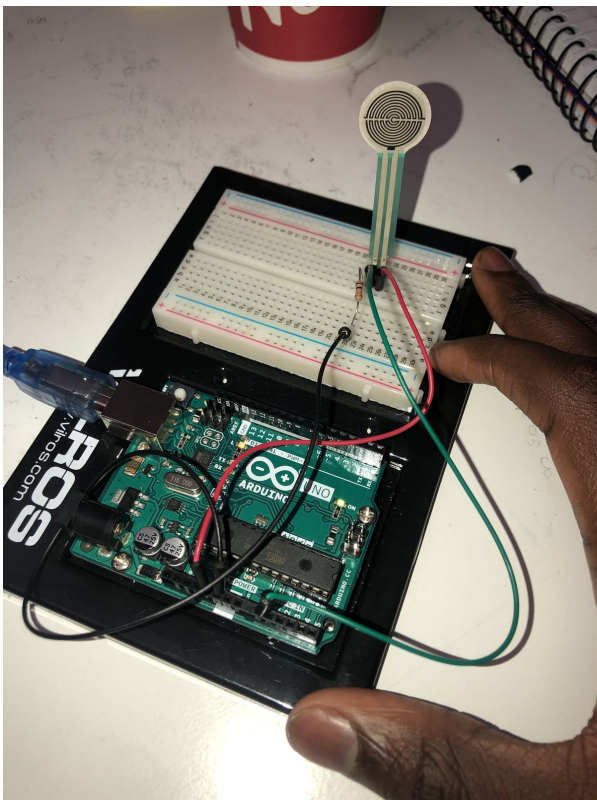
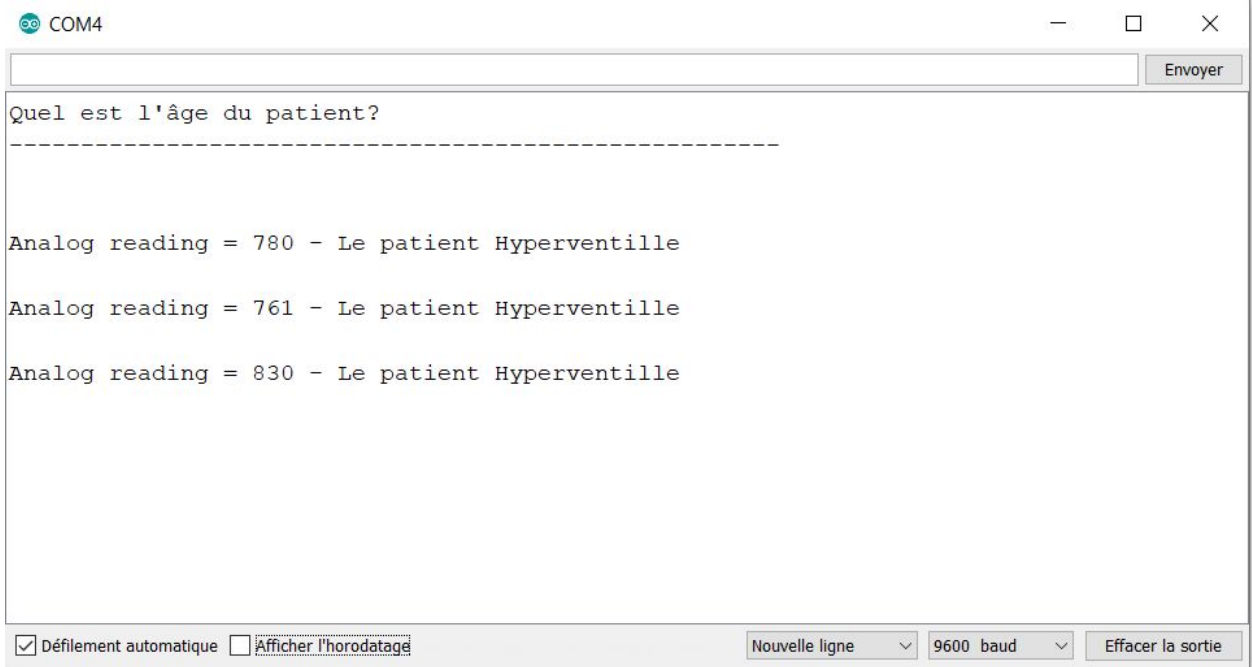
Arduino/Genuino Uno sur COM4
22:20
2020-11-05

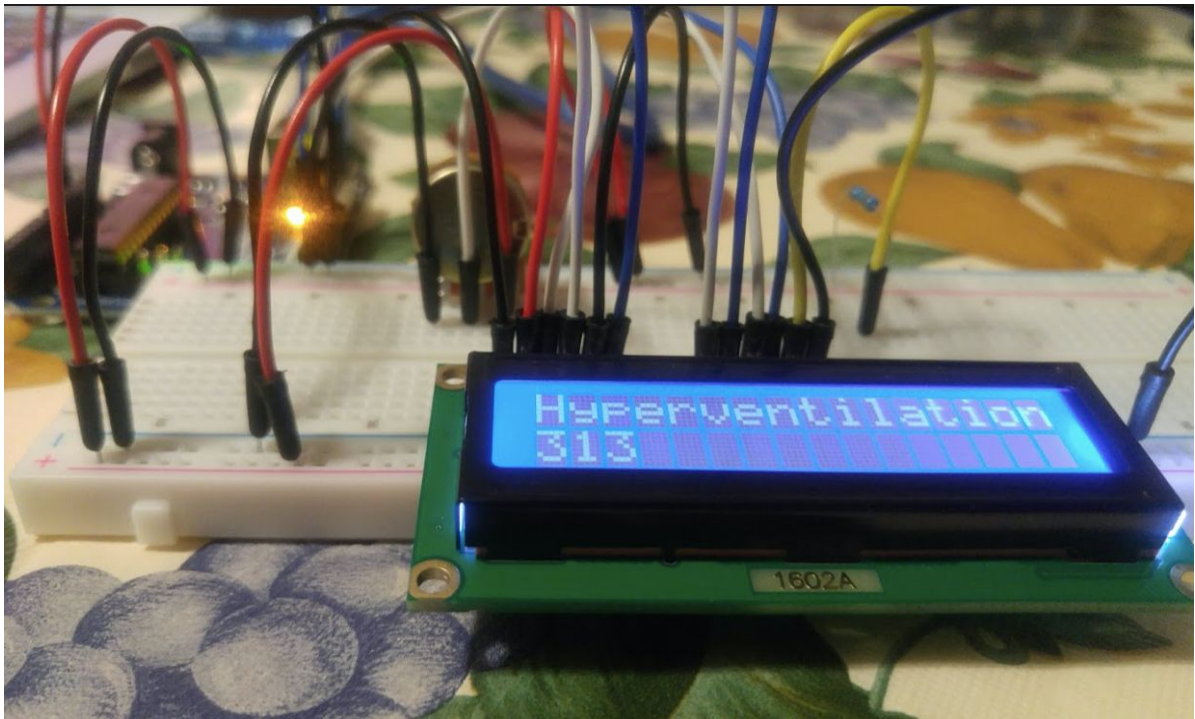
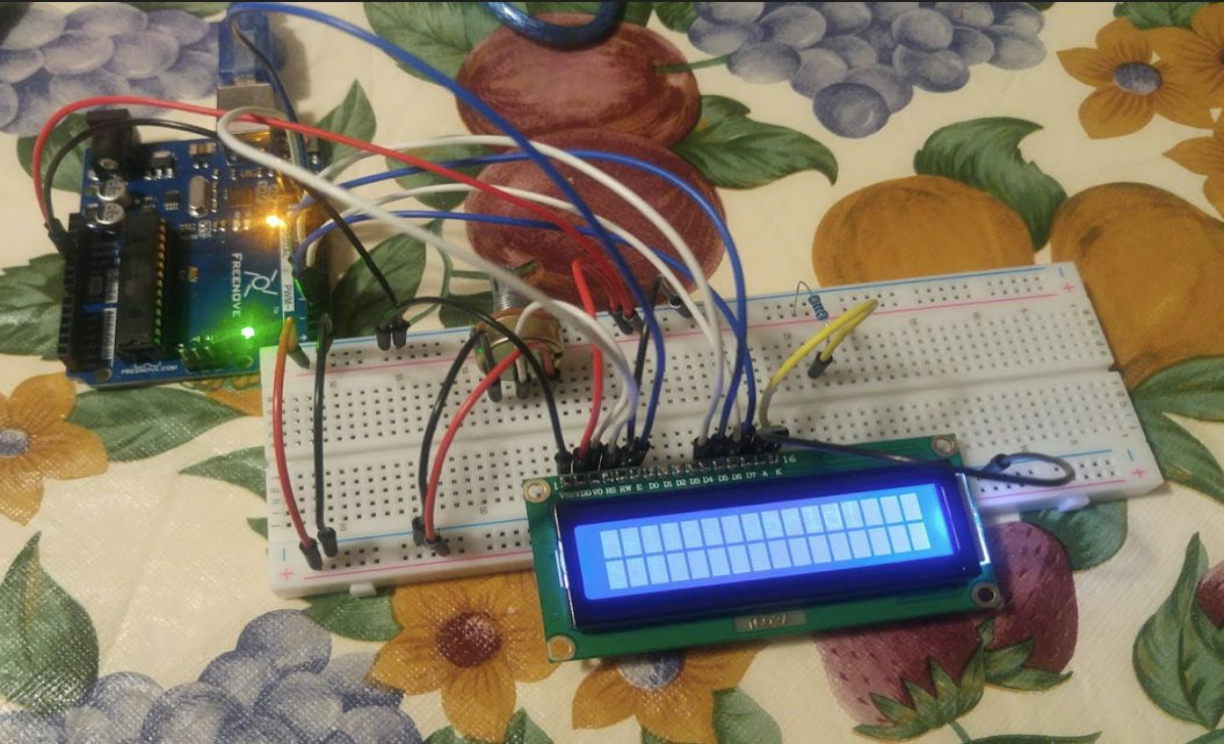
```
MotorTest
1 // -----
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56 Serial.println(" - Le patient hypoventille niveau 2");
57 } else if (fsrReading < 200) {
58 Serial.println(" - Le patient hypoventille niveau 1");
59 } else if (fsrReading < 500) {
60 Serial.println(" - La Respiration du patient est regulariser");
61 } else {
62 Serial.println(" - Le patient Hyperventille");
63 }
64 }
65 else if (age < 90)
66 {
67 if (fsrReading < 75) {
68 Serial.println(" - Le patient âgée hypoventille niveau 3");
69 } else if (fsrReading < 100) {
70 Serial.println(" - Le patient âgée hypoventille niveau 2");
71 } else if (fsrReading < 200) {
72 Serial.println(" - Le patient âgée hypoventille niveau 1");
73 } else if (fsrReading < 300) {
74 Serial.println(" - La Respiration du patient âgée est regulariser");
75 } else {
76 Serial.println(" - Le patient âgée Hyperventille");
77 }
78 }
79 }
80 delay(1000);
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
```

Compilation terminée

Le croquis utilise 2414 octets (7%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 468 octets (22%) de mémoire dynamique, ce qui laisse 1580 octets pour les variables locales. Le maximum est de 2048 octets.

Arduino/Genuino Uno sur COM4
22:24
2020-11-05





<p>Description - capteur de pression</p>	<p>Nous observons ici les divers éléments responsables au fonctionnement du système. Nous considérons tout d'abord les lignes de codes permettant la personnalisation de l'analyse des données. En un premier temps, le système demande à l'utilisateur l'âge du patient avec qui il traite. Par la suite, une série de ligne de codes s'effectuent basé sur la réplique respective. Le patient est ensuite invité à appliquer une force constante sur le capteur de pression sur un intervalle de 10s. En faisant appel à la méthode <i>forReading</i> le système détermine la condition cardio-respiratoire du patient et en conclut si des mesures de ressuscitation cardio-pulmonaire doivent être prise. Afin d'assurer la fiabilité des données récoltés par le capteur de pression, la véhémence de condition du patient se subdivise en trois niveaux d'hypoventilation; soit niveau 3, niveau 2 et niveau 1, étant en ordre d'intensité décroissante respective. Par exemple, si un patient ayant entre 0-18 ans applique une force inférieure à 10 udf (unité de force), le système interprète qu'il s'agit d'un patient qui hypoventile de niveau 3. Si ce même patient applique cette fois ci une force inférieure à 100 udf, le patient hypoventile de niveau 2. Si la pression est inférieure à 300 udf, il est dit que le patient hypoventile de niveau 1. Cependant, si la pression analysée par le système se situe entre 300 udf et 800 udf, la respiration du patient est considérée comme étant régularisé. Finalement, si la pression captée dépasse 800 udf, il est dit que le patient hyperventille et des mesures de réanimation cardio-respiratoire sont recommandés. Dans ce cas-ci, nous avons un patient ayant une pression moyenne de 790 udf. L'écran LCD affiche donc à l'usager que le patient hyperventille.</p>
<p>Composantes du système</p>	<ul style="list-style-type: none"> - Écran LCD - Breadboard - Arduino Uno - Fils électriques - Resistance (220 ohms) - Potentiomètre - Capteur de pression

Essais et Analyse

Critères de conception	Esthétique	Affichage	Sécurité	Fibabilité	Mobilité	Prix (\$CDN)	poids	Temps d'installation
Unité (1-5)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Résultats attendus (1-5)	3	5	4	5	5	5	5	4
Résultats réels (1-5)	3	3	3	4	4	5	5	4

Divers facteurs ont été pris en considération lors de la conception de notre deuxième prototype. De nombreux essais ont été réalisés afin de juger la performance du dispositif. Suite à cette période de tests, nous remarquons que le prototype respecte un grand nombre de critères établis par le client. En effet, le temps d'installation du prototype est d'environ 10 minutes, ce qui est légèrement inférieur aux exigences, soit un temps d'installation inférieur ou égal à 15 minutes. De plus, tous les matériaux dont nous avons eu recours pour la conception de notre dispositif ont été procurés à un prix abordable de 39.95\$. Le prototype respecte donc notre budget initial de 100\$. Aussi, le poids de notre appareil s'élève à environ 360g, incluant l'ambu bag, ce qui est inférieur à la contrainte de 400g.

Nos essais nous ont pareillement permis d'évaluer nos hypothèses de départ au sujet du prototype. Nous constatons ainsi que la majorité de ces hypothèses se sont révélés être exactes.

Lors de l'essai de notre code, nous avons remarqué que ce dernier n'est pas fonctionnel à 100%. En effet, l'analyse de respiration ne s'effectue que sur un intervalle d'environ 10 secondes, et non en continu tel que nous l'aurions souhaité.

De plus, l'éclairage de l'écran LCD se retrouve être tellement fort qu'il rend difficile la lecture des données. Prochainement, nous allons essayer de modifier le code afin de le rendre fonctionnel sur une plus longue période de temps. De plus, nous allons essayer de résoudre le problème d'éclairage de l'écran LCD. Nous pensons qu'il doit s'agir d'une erreur de connexion des fils électriques ou bien quelques lignes de code manquantes. Aussi, pour ce prototype, l'âge du patient a été entré manuellement en utilisant le clavier de l'ordinateur. Dans l'acheminement de notre produit final, nous allons implémenter un pavé numérique à notre prototype afin d'effectuer une telle action.

Conclusion

En conclusion, ce livrable nous a permis de développer un prototype compréhensif, muni de la majorité de ces fonctions et exigences, et nous assurer du bon fonctionnement de notre concept. L'essai de notre prototype nous a donné l'opportunité d'évaluer sa performance par rapport aux spécifications cibles du client et vérifier, par la même occasion, nos hypothèses de départ. Nous pouvons ainsi conclure que la conception du second prototype a été réalisée avec succès. Le prototype démontre efficacement le fonctionnement du dispositif.

Au cours des prochaines semaines, nous allons continuer de travailler sur le prototype afin d'implémenter toutes autres composantes nécessaires au dispositif et présenter un produit final de calibre supérieur lors de la journée du design.

Références

<https://www.makerguides.com/fsr-arduino-tutorial/>

<https://dronebotworkshop.com/lcd-displays-arduino/>

<https://www.tinkercad.com/things/g3TN0hMzZLj-shiny-waasa/editel?tenant=circuits>