

BIBDUINO

PROJET ARDUINO



POLYTECH[®]
NICE-SOPHIA



Année : 2017-2018
Réalisé par : CARLES Claire-Victoria
SAUTEL Emma, SCHROPFF Solène

Table des matières

INTRODUCTION	2
Présentation et problématique.....	2
Objectifs initiaux.....	2
DÉVELOPPEMENT DU PROJET.....	4
Matériel.....	4
Schéma	5
Difficultés et solutions	6
Électronique	6
Informatique	6
Assemblage	7
CONCLUSION	9
État final du projet : objectifs atteints ?.....	9
Améliorations possibles.....	9
Synthèse	10
Remerciements.....	10
ANNEXE	11
Code	11

INTRODUCTION

Présentation et problématique

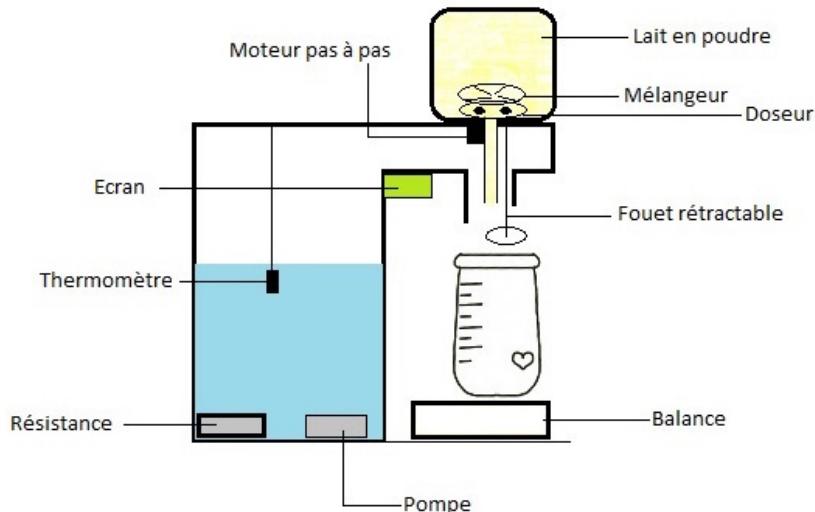
Durant ce projet, nous cherchions à automatiser et donc faciliter la préparation d'un biberon pour bébé.

D'une part, malgré le fait que ce type de machine existe déjà, elle n'est pas modulable (ex : le *BABYNES®* de *Nestlé*, qui utilise des capsules spécifiques à la marque) : cette machine répondrait donc à une réelle demande du consommateur.

D'autre part, cette automatisation permettrait de "faciliter" la vie de l'utilisateur : il aurait seulement avoir à lancer à distance la préparation du biberon, se lever au dernier moment et récupérer un biberon tout prêt économiserait de précieuses minutes de sommeil.

Notre principale problématique était donc de créer une machine modulable, où on pouvait choisir la quantité de poudre à mettre, et tout ça contrôlable depuis une application mobile connectée.

Objectifs initiaux



Ce schéma représente le Bibduino comme nous l'aurions idéalement vu.

Lors du lancement de l'application (non représentée), l'utilisateur choisit sa quantité de poudre. Cette valeur est envoyée à l'Arduino via un module Wifi, qui permet une plus grande portée. Par exemple, la chambre du parent n'est pas forcément adjacente à la cuisine, où pourrait se trouver le Bibduino : un Bluetooth ne serait pas assez performant.

Dès son lancement, un écran LCD indique l'état de l'avancement. De l'eau est chauffée via une résistance chauffante dans un récipient à part. Le thermomètre assure que l'eau soit à la bonne température.

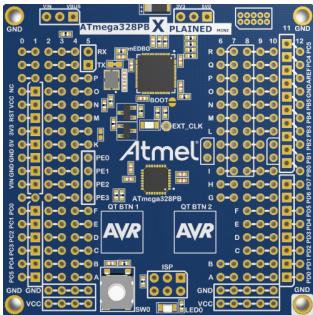
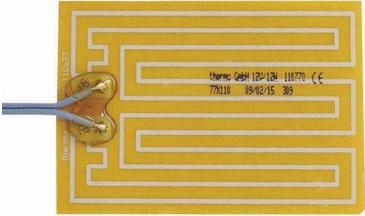
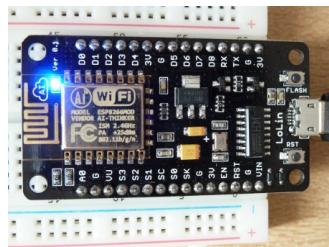
Après que l'eau ait été aspirée via une pompe vers le biberon, un moteur pas-à-pas se déclenche, ouvre la trappe d'un doseur et délivre la quantité voulue de poudre de lait. La balance permet de stopper le processus au moment où la quantité est atteinte.

Enfin, un fouet rétractable vient mélanger la poudre et l'eau, et un message s'affiche sur l'écran LCD.

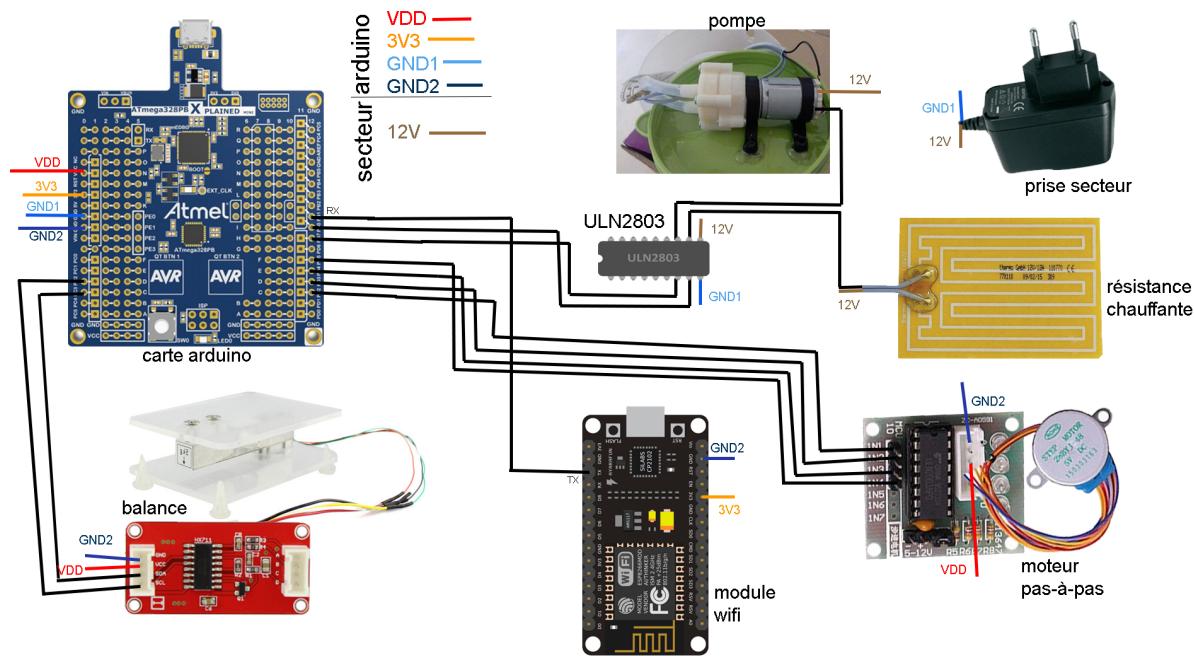
Servez chaud, c'est prêt.

DÉVELOPPEMENT DU PROJET

Matériel

<p>Carte Arduino ATmega328PB Xplained mini</p>  <p>⇒ Gère l'ensemble de l'électronique</p>	<p>Distributeur de poudre pour biberon + pièce imprimée en 3D</p>  <p>⇒ Squelette du Bibduino, contient l'électronique</p>	<p>Balance électronique HX711</p>  <p>⇒ Vérifie le poids du biberon</p>	
<p>Résistance chauffante Thermo Folienheizung 110X77MM</p>  <p>⇒ Chauffe l'eau</p>	<p>Moteur pas-à-pas ULN2804</p>  <p>⇒ Fait tourner le doseur</p>	<p>Pompe</p>  <p>⇒ Aspire l'eau jusqu'au biberon</p>	
<p>ULN2803</p> 	<p>RELAIS 5V</p>  <p>⇒ Transforme le 12V en 5V (provisoire)</p>	<p>Module Wifi ESP8266MOD</p>  <p>⇒ Connecte l'application à l'Arduino</p>	<p>AppInventor MIT</p>  <p>⇒ Interface de l'application</p>

Schéma



(1) Schématisation du montage électronique



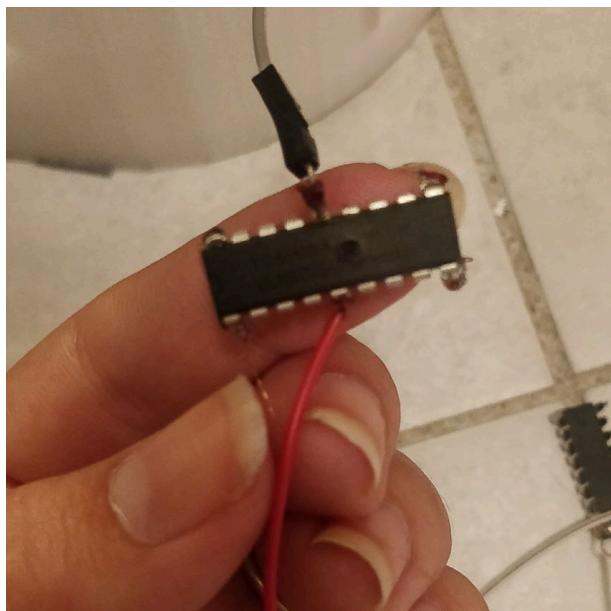
(2) La réalité du montage

Difficultés et solutions

Électronique

Pour la partie électronique, il y a eu des difficultés de communication entre le module Wifi et l'Arduino, avec la communication RX-TX. Les seules solutions furent patience et persévérance.

La puce ULN2803 a quant à elle fait quelques surtensions, dont une fatale.



RIP

Nous avons donc dû la remplacer au dernier moment (c'est-à-dire la veille) par une module relai 5V. Or à cause de cela, la puissance d'une seule carte Arduino n'était plus suffisante pour alimenter le moteur. Nous avons donc ajouté une seconde carte Arduino.

Avant cette surtension, jusqu'à la veille de la présentation, le Bibduino fonctionnait très bien. Nous nous sommes rendu compte d'un fil défectueux qui affaiblissait la pompe, l'avons remis en place et la puce a littéralement explosé. Depuis ça, nous avons tenté de faire fonctionner le relai 5V, puis ajouté la carte, mais malheureusement aucune solution n'a été effective. Il est donc important de retenir que le schéma présenté précédemment était fonctionnel, c'est pour cela que nous ne l'avons pas modifié.

Informatique

Pour l'interface de l'application, nous nous sommes premièrement dirigées vers Android Studio. Or pour créer une application de A à Z, il faut assimiler de nombreuses nouvelles notions de Java et de XML. C'est pour cela que nous nous sommes finalement tournées vers AppInventor du MIT. Par la suite, il a fallu comprendre comment fonctionnaient en parallèle l'interface de l'utilisateur et les *blocks* de code, mais ce fut déjà plus simple.

Ensuite, le nouveau gros problème était d'envoyer les données entrées par l'utilisateur au module Wifi. Or, pour cela il faut récupérer son adresse IP et la rentrer dans l'application. Cela signifie qu'à chaque changement de réseau Wifi, l'adresse IP change elle aussi, et il faut donc la modifier dans l'application même. Malheureusement, aucune solution n'a été trouvée. Or retenons qu'une fois placé dans une maison, le Bibduino ne va pas être spécialement déplacé et son réseau Wifi ne changera donc pas.

Assemblage

Premièrement, la pièce imprimée n'est pas tout à fait de la bonne taille : en effet, nous avons utilisé des parties d'une machine déjà pré-faite, que nous avons démonté. Après avoir modélisé une base sur *Autodesk Inventor* en essayant de respecter au mieux les dimensions, quelques millimètres ont fait la différence : les deux pièces vont ensemble mais pas parfaitement. Nous avons collé les deux avec un pistolet à colle.



(1) Percage de la pièce imprimée



(2) Fixage de la pièce avec le sommet du squelette

Ensuite, le récipient à poudre qui ne veut plus se fixer sur son support d'origine, très probablement à cause d'un coup de chaud. Pourtant, nous avons récupéré et démonté une machine d'origine où tout s'emboitait parfaitement (*cf les photos ci-dessous*). Pour y remédier, nous avons encore utilisé notre fidèle ami, le pistolet à colle.



(1) Squelette original

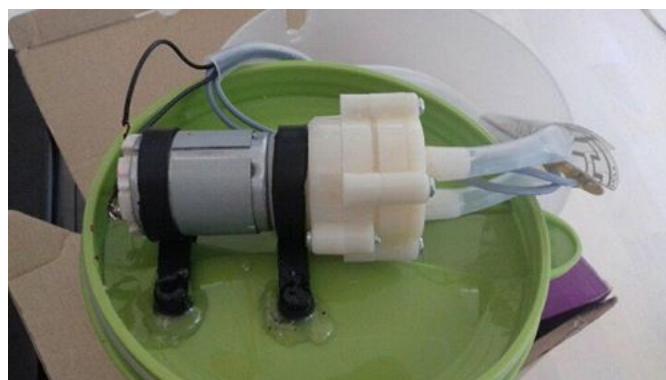


(2) Sommet démonté



(3) Le tout recollé ensemble

Enfin, nous avons dû fixer la pompe ainsi que les fils électriques en évitant qu'ils soient en contact avec la poudre. Pour se faire, nous les avons collés sous le couvercle de la machine, amovible afin de mettre de la poudre dans le biberon.



(1) Collage de la pompe sous le couvercle

CONCLUSION

État final du projet : objectifs atteints ?

Reprenez l'introduction : qu'est-ce qui a été *réalisé*, *pas réalisé* ?

Lors du lancement de l'application :

- *L'utilisateur choisit la quantité de poudre.*
- *L'application et l'Arduino se connectent pour échanger les informations via Wifi.*

Lors de la préparation du biberon :

- ~~*L'eau est chauffée dans un récipient à part via une résistance chauffante.*~~
 - ➔ Le parent utilise sa propre bouteille d'eau (Évian par exemple) où il introduit un tube : en passant dans ce tube, l'eau est chauffée grâce à la résistance.
 - ➔ Ce choix a été réalisé car chauffer plusieurs fois la même eau peut être nocif, encore plus pour un enfant en développement.
- ~~*Un thermomètre vérifie la température de l'eau.*~~
 - ➔ Pas de thermomètre : le parent vérifie la température lui-même.
 - ➔ De plus, la mode est au biberon à température ambiante, la résistance n'a donc pas à chauffer beaucoup pour satisfaire le client.
- *L'eau est pompée vers le biberon.*
- *Le moteur pas-à-pas contrôle un doseur et délivre la quantité voulue de poudre.*
- *La balance vérifie la quantité dans le biberon.*
- ~~*Un fouet mélange l'eau et la poudre.*~~
 - ➔ Encore une fois, on remet le rôle au parent de mélanger le biberon lui-même.
- ~~*Un écran LCD affiche des messages sur l'état d'avancement.*~~
 - ➔ Pas d'écran LCD, remplacé par un suivi sur l'application.

Améliorations possibles

Premièrement, nous aimerais remplacer la puce qui a explosé. Malheureusement, cela est arrivé en dehors des heures de cours. Malgré le fait que nous nous soyons rendues au FabLab afin de la remplacer, la deuxième que nous avons soudé ne fonctionne pas, et nous n'en n'avions pas d'autres. Notre tentative d'utilisation du relai s'est malheureusement soldée par un échec. Il est donc primordial de remplacer cette puce défectueuse par deux séparées pour éviter toute surchauffe, et le Bibduino sera comme neuf. En effet, jusque-là il fonctionnait. Nous n'avons pas mis d'écran LCD par manque de temps, mais nous aurions vraiment aimé le brancher afin d'avoir un suivi à même l'appareil.

De plus, il nous faut absolument un moteur pas-à-pas plus puissant afin d'éviter tous les problèmes qui y sont liés : arrêt au moindre frottement, rotation très lente...

Au niveau de l'assemblage, l'étanchéité du doseur est à revoir, il y a quelques fuites de poudre. Aussi, nous aurions voulu modéliser un cache fonctionnant comme pour le boîtier d'une télécommande afin de rendre l'arrière de notre machine plus esthétique.

Synthèse

Durant ces six mois de projet, nous avons pu découvrir des aspects de ce qu'était d'être un "ingénieur". Nous avons dû être autonomes, imaginer et créer un appareil électronique de A à Z (*ou plutôt de C à X*), et apprendre à gérer par nous-même nos problèmes. Bien que nos débuts aient été laborieux et un peu lents — ceci afin d'étudier au mieux le fonctionnement de notre machine — nous avons réussi à synthétiser nos connaissances et à les mettre en pratique.

L'Arduino fut une interface très utile pour concevoir notre machine : en effet, son environnement a été conçu pour être utilisé facilement, et nos connaissances d'électronique numérique/analogique et d'informatique ont enfin trouvé un sens pratique.

Comme nous l'avons dit précédemment, des améliorations sont envisageables, voire nécessaires, et la machine est encore perfectible. Encore aujourd'hui, certains composants nous font défaut mais en général nous sommes satisfaits du projet réalisé. Malgré de nombreuses difficultés, nous avons tout de même réussi à atteindre la majorité de nos objectifs.

Remerciements

Car ils nous ont aidé à réaliser ce projet, nous remercions :

- Atmel, de nous avoir fourni la carte Arduino,
- M. Masson et Ferrero, de nous avoir aidées et apporté leurs connaissances,
- Marc Former au FabLab, de nous avoir accueillies, aidées et conseillées durant le montage de la machine.

ANNEXE

Code

Code compilé pour faire fonctionner l'Arduino :

```

/** RX TX */
#include <SoftwareSerial.h>
SoftwareSerial mySerial(9,10);

/** BALANCE */
#include <hx711.h>
Hx711 scale(A2,A3);
const int poids_bibi = scale.getGram();

/** MOTEUR PAP */
#include <Stepper.h>
//broches moteur
const int NombrePas = 350;
const int pas_1=2;
const int pas_2=4;
const int pas_3=3;
const int pas_4=5;
Stepper moteur(NombrePas, pas_1, pas_2, pas_3, pas_4);

/** INITIALISATION PINS POMPE ET RESISTANCE */
const int pin_pompe = 7 ;
const int pin_resis = 8;

void setup() {
  Serial.begin(9600);
  mySerial.begin(4800);

  moteur.setSpeed(50); // On définit la vitesse du moteur pas à pas
  pinMode(pin_pompe,OUTPUT);
  pinMode(pin_resis,OUTPUT);
  pinMode(pas_1, OUTPUT);
  pinMode(pas_2, OUTPUT);
  pinMode(pas_3, OUTPUT);
  pinMode(pas_4, OUTPUT);

}

void loop() {

  int quantite = mySerial.read()-0; //quantite de poudre (g)
  int eau = quantite*6; // en ml

  Serial.println("Quantité désirée : ");
  Serial.println(quantite);

  if (scale.getGram() > 10){ //poids min biberon
    int poids_bibi = scale.getGram();
    if (quantite>0) {
      allume(pin_resis);           // allumage de la résistance
      fonctionnement_pap(1);       // ouverture du pap
    }
  }
}

```

```

while(scale.getGram()<poids_bibi+quantite){
    Serial.println("Versement de la poudre en cours");
    Serial.println(poids_bibi);
    Serial.println(scale.getGram());
}
fonctionnement_pap(-1); //fermeture du pap

while(scale.getGram()<poids_bibi+quantite+eau-10){
    Serial.println("Versement de l'eau en cours");
    Serial.println(poids_bibi);
    Serial.println(scale.getGram());
    allume(pin_pompe); // allumage de la pompe
}

eteint(pin_pompe); //eteint pompe
eteint(pin_resis); //eteint resistance

Serial.println("bibi terminé");

delay(5000000);
}
else { Serial.println("une erreur est survenue"); }
}

else { Serial.println("Erreur: bibi non présent"); }
}

/** FONCTIONS */
int balance() {
    int mesure=scale.getGram();
    return (mesure);
}

void fonctionnement_pap(int n) {
    for (int i=1; i<NombrePas; i++){ // boucle de defilement du nombre de pas
        moteur.step(n); // un pas en sens positif/negatif
    }
}

void allume(int pin_comp) {
    digitalWrite(pin_comp,HIGH);
}

void eteint(int pin_comp) {
    digitalWrite(pin_comp,LOW);
}

```

Code du module Wifi

```
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>

const char* ssid = "iPhone de Emma"; //Livebox-3e0c;//"AndroidAP";//"OnePlus 5 4241";
const char* password = "emmadamour";//3356931666F098A39845403845;//"cvcvcvcv"; //"sole

WiFiServer server(80);
SoftwareSerial mySerial(13,15); //RX TX

void setup() {
    mySerial.begin(4800);
    Serial.begin(9600);

    delay(10);
    pinMode(13, OUTPUT);

    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    // Start the server
    server.begin();
    Serial.println("Server started");

    // Print the IP address
    Serial.print("Use this URL to connect: ");
    Serial.print("http://");
    Serial.print(WiFi.localIP());
    Serial.println("/");

}

void loop() {
    // Check if a client has connected
    WiFiClient client = server.available();
    if (!client) {
        return;
    }
}
```

```

// Wait until the client sends some data
Serial.println("new client");
while(!client.available()) {
    delay(1);
}

// Read the first line of the request
String request = client.readStringUntil('\r');
Serial.println(request);
client.flush();

String poudre1="";
if (request.length()==18) {
    poudre1 = request.substring(5,9);
}
else if (request.length()==17) {
    poudre1 = request.substring(5,8);
}
else {
    poudre1 = request.substring(5,7);
}

int poudre=poudre1.toInt();

if (poudre>1000) {
    int temp=poudre/10;
    int temp2=temp+250;
    poudre=temp2/30;
}
int eau=poudre*6;
int temp=poudre;

while(true){
if (poudre==temp) {
    mySerial.write(poudre);
}
else {
    temp=poudre;
    mySerial.write(poudre);
}
// Return the response
// client.println("HTTP/1.1 200 OK");
// client.println("Content-Type: text/html");
// client.println(""); // do not forget this one
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<body>");
client.println("<style>");
client.println("body {background-color: rgb(246,241,199)} </style>");


```

```
client.print("<h2>Préparation du bibi suivant:</h2><br>");  
client.print("<strong>Quantité de lait en poudre: </strong>");  
client.print(poudre);  
client.print("g<br><br>");  
client.print("Volume d'eau: ");  
client.print(eau);  
client.print("ml</strong>");  
  
client.println("<br><br>");  
client.println("</body></html>");  
}
```