RAPPORT AFB/AFT

Intitulé du projet : La Magic Box

Numéro de l'équipe : 106

Date de rédaction : 27/01/2023

Membres de l'équipe

Nom	Prénom
Bastian	Sévane
Passelaigue	Aurélien
Bassereau	Adrien
Bouanane	Aya
Branchereau	Thomas

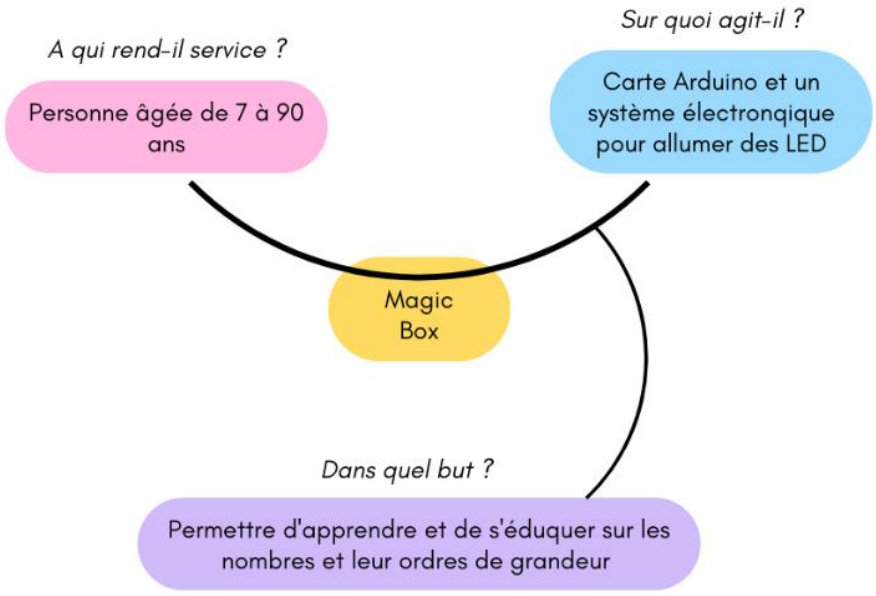
TABLE DES MATIERES

1 Objet

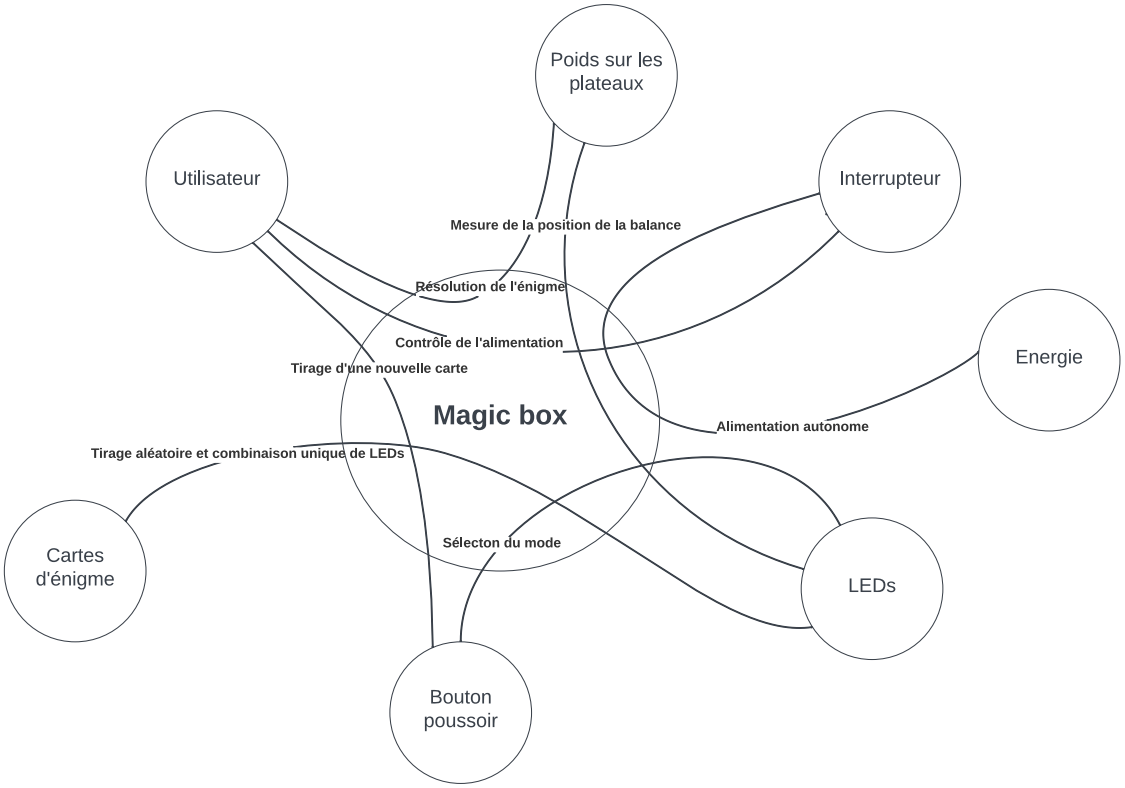
La Magic Box est un jeu éducatif sur les mathématiques, principalement le calcul mental, destinée aux personnes de 7 à 90 ans. C'est une boîte où se trouve à l'intérieur une balance cachée, qui n'est pas visible par l'utilisateur. Cet objet permet de coupler les chiffres avec une grandeur physique : la masse. Le but est d'équilibrer cette balance à travers le calcul des poids de chaque côté de l'objet. Les poids auront des masses différentes mais conserveront le même volume et la même forme cubique afin d'accroître le niveau. Pour les enfants, ce projet leur permettra de faire un premier lien entre les nombres et les poids ainsi qu'une première approche des différents ordres de grandeur. Des LEDs de différentes couleurs, placées au-dessus de la boîte, permettront au joueur de savoir si la balance est équilibrée ou non.

Ce rapport contient la bête à corne qui permet d'analyser l'objectif du produit. Il contient également l'analyse fonctionnelle sous la forme d'un diagramme FAST, listant les fonctions de service, les fonctions techniques ainsi que les solutions techniques associées. Enfin, nous avons réalisé la liste des différents composants nécessaires à la réalisation de la Magic Box.

2 Bête a corne

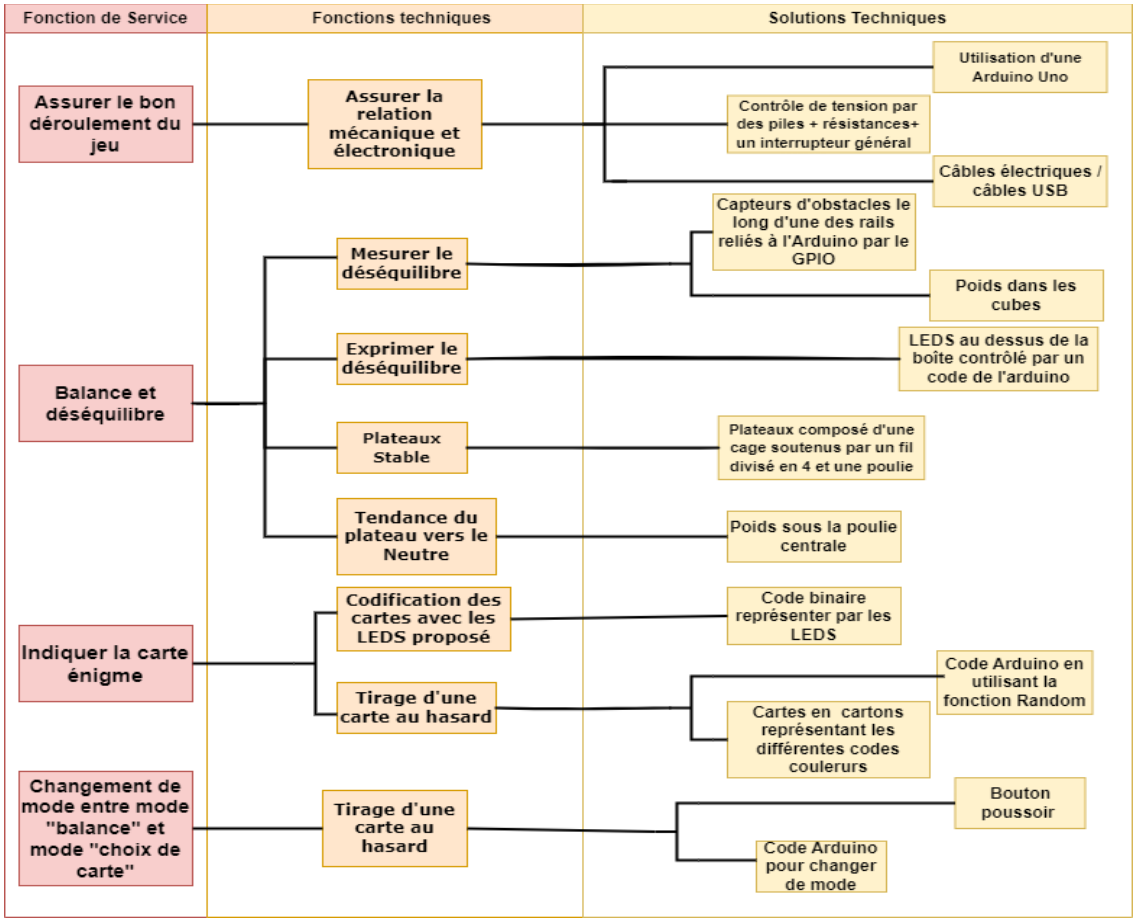


3 Diagramme pieuvre



4 Analyse fonctionnelle technique / Étude structurelle

4.1 Schéma



4.2 Explications :

- Piles : 6 piles de 1,5V seront utilisées en série, pour une tension de 9V (correspondant à la tension attendue en DC par l'Arduino).
- Résistances : Les résistances de 510 Ohms permettent d'alimenter les LEDs par les pins 5V du GPIO de l'Arduino sans risquer de les abimer (I = U/R = 0,01 A = 10 mA courant accepté par les LEDs).
- Capteurs : Les capteurs infrarouges permettent d'observer dans quelle « zone » la balance est.
- Arduino Uno : L'Arduino Uno contrôle tout le projet, à travers son GPIO, par du software disponible en annexe.
- Plateau : une cage sera créée sur chaque plateau qui sera soutenu par un fil divisé en 4 et une poulie pour permettre d'éviter tout frottement de la corde
- Poids sous la poulie centrale : Afin d'assurer que la balance retourne vers la position centrale quand elle est équilibrée, et qu'elle ne penchera pas complètement d'un côté dès qu'elle sera déséquilibrée, un petit poids sera ajouté au centre entre les poulies (plus bas que celles-ci).

- Bouton Poussoir : Le bouton poussoir permettra de passer du mode du “choix de carte” (les LEDS émettront un code binaire généré aléatoirement) au mode de balance (une seule LED sera allumée en fonction de l’équilibre de la balance) grâce au software de l’ARDUINO
- Partie Arduino (code en annexe) :
    - Fonction “balance” : prend en compte le capteur (situé le long des rail) qui est activé et retourne une LED spécifique qui s'active
    - Fonction “random” : prends un nombre aléatoire entre 1 et le nombre de cartes disponibles (5 LEDs nous donnent 32 affichages possibles, 31 pour éviter un affichage vide) pour le retourner en code binaire avec les LEDs, une représentation de cette affichage sera disponible sur les cartes.
    - Fonction “changement” : est capable de passer de “balance” à “random” selon l’état du bouton poussoir

La structure du système est obtenue en associant des composants aux fonction techniques et en définissant les interactions entre ces composants. D'accord donc un indice fonctionnel. J'ai commencé à agrémenter cette analyse fonctionnelle et éventuellement des types d'informations. Qui vont sortir des différentes fonctions où qui vont rentrer dans les fonctions principales.
Et donc je vais pouvoir justement, définir les interactions entre les composants. Vous devez donc reprendre l'analyse fonctionnelle, indiquer les grandeurs physiques, choisir et dimensionner les composants qui vont permettre le traitement de ces grandeurs. Éventuellement, modifier l'analyse fonctionnelle pour ajouter certaines fonctions dans le cahier des charges et l'analyse fonctionnelle n'auraient pas été pensées.
Gérer l'ordre de traitement des informations. Vous allez devoir créer des algorithmes ou des organigrammes pour par exemple si j'ai un robot qui se déplace, s'il rencontré un obstacle. Eh bien ce traitement est prioritaire sur le fait de rouler tout droit.
Donc, définir des algorithmes clairs et précis en analysant et en hiérarchisant toutes les informations qui vont transiter dans les fonctions principales et en gérant les priorités.

5 Annexes au rapport

5.1 Liste de commande

Nom du projet	Désignation de l'article	Fournisseur	Lien fournisseur	Référence fournisseur	Quantité	Prix	Commentaires
Magic Box	Led jaune diffusante 5mm	LEXTRONIC	https://www.lextronic.fr/led-jaune-diffusante-5mm-3563.html	LED5YLN	5	0,15€	
Magic Box	Led rouge diffusante 5mm	LEXTRONIC	https://www.lextronic.fr/led-rouge-diffusante-5mm-3562.html	LED5RLN	5	0,14€	
Magic Box	Led verte diffusante 5mm	LEXTRONIC	https://www.lextronic.fr/led-verte-diffusante-5mm-3561.html	LED5GLN	3	0,15€	
Magic Box	Module capteur d'obstacle	LEXTRONIC	https://www.lextronic.fr/module-capteur-d-obstacle-51712.html	OPENST1081	5	4,50€	
Magic Box	Bouton-poussoir OFF - (ON) rouge	LEXTRONIC	https://www.lextronic.fr/bouton-poussoir-off-on-rouge-3155.html	LEX-1634627	1	2,90€	
Magic Box	Carte Arduino UNO Rev 3 A000066 version originale fabriquée en Italie - Base ATMega328 @ 16 MHz	LEXTRONIC	https://www.lextronic.fr/carte-arduino-uno-dip-rev3-a000066-2474.html?fast_search=fs	A000066	1	23,80€	
Magic Box	Coupleur pour 6 piles AA pour Arduino®	LEXTRONIC	https://www.lextronic.fr/coupleur-pour-6-piles-aa-pour-arduino-39958.html	LEX-CON3	1	4,44€	
Magic Box	Plaque de connexions sans soudure 400 trous	LEXTRONIC	https://www.lextronic.fr/plaque-de-connexion-sans-soudure-19804.html	LEX-BREAD01	1	6,00€	
Magic Box	10 piles alcalines AA 1,5V	LEXTRONIC	https://www.lextronic.fr/10-piles-alcalines-aa-15v-64305.html	PCA9002	2	5,90€	
Magic Box	Interrupteur à bascule ON-OFF	LEXTRONIC	https://www.lextronic.fr/interrupteur-a-bascule-on-off-64711.html	LEX-2913883	2	1,95€	
Magic Box	Cordon USB A mâle - USB B mâle (50 cm)	LEXTRONIC	https://www.lextronic.fr/cordon-usb-a-male-micro-usb-b-male-50cm-54284.html	CABLE-USB-50	1	3,00€	
Magic Box	Résistance couche carbone 1 W	LEXTRONIC	https://www.lextronic.fr/resistance-couche-carbone-1-w-46297-63825.html#/1825-resistance_couche_carbone-510_	RC510E0	10	0,10€	Variante 510 Ohms
Magic Box	Jeu de 20 straps flexibles F/F (50 cm) transformables	LEXTRONIC	https://www.lextronic.fr/jeu-de-20-straps-flexibles-ff-50-cm-transformables-62003.html	RB-CB3-050	2	5,40€	
Magic Box	Jeu de 10 roulements à billes 8 mm	LEXTRONIC	https://www.lextronic.fr/jeu-de-10-roulements-a-billes-19408.html?fast_search=fs	MAK87310	1	7,80€	

5.2 Code

5.2.1 Version langage naturel

```
Initialisation des variables d'entiers correspondant aux pins de l'Arduino
utilisées :
liste d'entiers pinsCapteurs
liste d'entiers pinsLEDs
entier pin_switch_signal
entier pin_input_switch
entier loop_delay

entier loop_delay
entier nb_cartes

Initialisation :
      Pour chaque element de pinsCapteurs :
            Mettre la pin correspondant en mode recepteur
      Pour chaque element de pinsLEDs :
```

```
            Mettre la pin en mode emetteur

      Mettre la pin de pin_switch_signal en mode emetteur
      Commencer a emettre avec la pin de pin_switch_signal un signal
positif
      Mettre la pin de pin_input_switch en mode recepteur

      Creer la seed aleatoire basee sur un capteur non utilise en analogue

Boucle principale :
      Si le pin de pin_input_switch recoit un signal positif :
            Generer un nombre aleatoire et l'encoder en base 2 sur une
liste de 5 bits
            Tant que le pin recoit un signal positif :
                  Allumer les LEDs correspondant aux bits du nombre (par
leur pin respectif)
            Attendre loop_delay

      Pour chaque pin de LED, commencer envoyer un signal positif si la
pin du capteur correspondant recoit un signal positif, commencer a envoyer
un signal negatif sinon

      Attendre loop_delay
```

5.2.2 Version Arduino

```
int pinsCapteurs[5] = {5,6,7,8,9}; // Pins correspondants aux capteurs
(dans l'ordre des capteurs, la taille doit etre le nbCapteurs)
int pinsLEDs[5] = {10,11,12,13,14}; // Pins des LEDs indicatrices ()
int pin_switch_signal = 15; // Pin alimentant l interrupteur
int pin_input_switch = 16; // Retour de l'interrupteur
int loop_delay = 250;
int nb_cartes = 6; // Le nombre de cartes dans le deck d enigmes
bool presence(int capteur){return digitalRead(pinsCapteurs[capteur])==1;}
void AfficherPresence(int i){ // Pour faire des tests
  Serial.print ("Sensor ");
  Serial.println(i);
  Serial.print (" (pin ");
  Serial.println(pinsCapteurs[i]);
  Serial.print (") : ");
  Serial.println (pinsCapteurs[i]);
}

void setup() {
```

```
  Serial.begin(9600); // Pour debug
  for(int i = 0; i < 5; i++){
    pinMode (pinsCapteurs[i], INPUT);
    pinMode(pinsLEDs[i], OUTPUT);
  }
  pinMode(pin_switch_signal, OUTPUT);
  digitalWrite(pin_switch_signal, HIGH);
  pinMode(pin_input_switch, INPUT);
  randomSeed(analogRead(0));
}

void loop() {
  if(digitalRead(pin_input_switch)==1){ // Si on est en mode affichage de
code d'énigme

    int rnd = random(1,nb_cartes+1); // + 1 pour eviter tout eteint, qui
n'est pas lisible

    for(int i = 0; i < 5; i++){
      if(rnd%(pow(2,i)==1)){
        digitalWrite(pinsLEDs[i], HIGH);
      } else{
        digitalWrite(pinsLEDs[i], LOW);
      }
    }
    delay(100); // Pour eviter de clignoter
    while(digitalRead(pin_input_switch)==1){ // Tant qu'on reste appuye,
pour ne pas recalculer la carte a chaque tick
      delay(loop_delay);
    }
  }
  for(int i = 0; i < 5; i++){
    if(presence(i)){
      digitalWrite(pinsLEDs[i], HIGH);
    } else{
      digitalWrite(pinsLEDs[i], LOW);
    }
  }
  delay(loop_delay);
}
```