**Mini projet char : Alan PHAM, Elouan CALLEC, Erwan FAVE**

Planifier le travail :

|  |  |
| --- | --- |
| **DATES** | **AVANCEMENT DU PROJET** |
| 22/11/2019 (3h) | Finalisation de la phase 1 et Commencement de la phase 2 |
| 25/11/2019 (1h) | Phase 2 |
| 26/11/2019 (2h) | Phase 2 |
| 27/11/2019 (3h) | Phase 2 |
| 29/11/2019 (3h) | Phase 2 |
| 02/12/2019 (1h) | Phase 2 |
| 03/12/2019 (2h) | Phase 3 |
| 04/12/2019 (3h) | Phase 3 |
| 11/12/2019 (3h) | Finalisation e la phase 3 et présentation |

Tableau de répartitions des tâches :

|  |  |  |
| --- | --- | --- |
| Erwan | Elouan | Alan |
| Arduino :  Piloter les moteurs  Acquérir les informations de la Raspberry afin de piloter la vitesse et la direction. | Raspberry :  Récupérer l’information transmise par la télécommande puis la transmettre à la carte Arduino | Télécommande :  Envoyer une information a la carte Raspberry (Mouvement, Vitesse). |

Tâches critiques (indispensables à la réussite du projet) :

Transmission des données de la télécommande au char, pouvoir communiquer à distance.

Faire avancer, reculer, tourner le char.

Les Solutions :

Matériel :

-Une batterie 12V

-Deux Moteurs 12V

-Cavalier

-Batterie 5V

-Pont en H

-Arduino

-Antennes RF 433MHz

A qui sert-il et à quoi sert-il ?

Le char répond au besoin de personnes cherchant contrôler un engin télécommandé. Il sert de jouet à la personne souhaitant l’utiliser. Il peut se déplacer dans différents environnements tels que sur la route, la terre et le sol.

Répartition des tâches :

La première personne doit acquérir les informations des boutons de la télécommande et les envoyer à la 2e personne. La 2e personne doit récupérer ces informations pour les envoyer à la 3e personne qui devra les traiter pour adapter les mouvements du char.

Limite d’étude :

Le projet final doit être fonctionnel selon le cahier des charges défini. Il ne sera pas effectué de modifications de structure, d’organisation, d’usage, de procédures opérationnelles et de composantes du système.

Performances envisagées :

Le système devra avoir des performances de vitesse conformes aux demandes du client à travers le cahier des charges. Il devra pouvoir se déplacer conformément à ces demandes également (avant, arrière, gauche, droite).

Choix des solutions :

|  |  |  |
| --- | --- | --- |
| **Solution :**  **Pilotage moteur :** | **Interface:** | **Communication:** |
| -Batterie (Choix n1) :  Peut faire avancer le char à une vitesse élevé et peut permettre d’avoir en variation de vitesse plus élevé qu’avec des piles.  -Piles (Choix n2) :  Ne va pas à plus de 5 v donc le char sera lent et prendra du temps pour tourner.  Il faut donc en mettre plusieurs ce qui reviens à un cout plus élevé et a un montage avec une marge d’erreur plus élevé.  -Pont en H :  C’est une des solutions qui permet de faire avancer,reculer,tourner et changer la vitesse.  On peut faire varier la vitesse avec un pont en H  Plus couteux que l’utilisation de deux relais.  -Deux relais :  C’est la deuxieme solution qui permet de faire avancer,tourner,reculer mais ne peut pas faire varier la vitesse.  On ne peut pas changer la Vitesse avec le pont en H. | **Interface graphique ou Interface physique**  L’interface graphique requiert un codage plus compliqué du fait de sa comptabilité avec l’appareil (La tablette ou le téléphone). De plus, elle sera plus gourmande en énergie et aura donc une durée de fonctionnement réduite par rapport à une interface physique, qui serait alimentée par des piles (AA ou AAA). Cependant, une interface graphique apparait comme évolutive : Si de nouvelles fonctionnalités comme des nouveaux mouvements ou des mises à jour sont proposées après la mise en service du produit, il est plus simple de les rajouter via l’application que sur une télécommande réelle, ou cela serait impossible d’ajouter des boutons.  **Etant donné que notre projet ne sera pas évolutif et qu’il doit consommer peu d’énergie, Une interface physique apparaît comme plus intéressante. Notre projet comportera donc une interface physique de type Manette/Télécommande.** | **-Récepteur radio :**  La communication par récepteur radio permet moins de chose que la communication Bluetooth mais est donc plus simple à mettre en place et revient moins cher.  **-Bluetooth :**  La communication Bluetooth à un portée de 9 à 10 mètres. Cette communication est compliquée à mettre en place notamment à cause de l’appairage mais permet de faire plus de chose que les autres communications tels que la transmission de sons. Elle est également bien adaptée pour réaliser des projets de voitures télécommandé mais est principalement utilisé pour piloter l’objet avec son smartphone.  **Sachant que nous utiliserons une interface physique et non virtuelle et que nous devons mettre en place la solution la moins couteuse, nous utiliserons une communication radio 433Mhz.** |

Différentes fonctions disponibles et choix de solutions :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FONCTION | TYPE D’INFO | SOLUTIONS POSSIBLES | ERGONOMIE | VITESSE DE TRANSMISSION | CHOIX |
| Arrêt et démarrage du moteur | Tout ou rien | -Bouton Poussoir  -Interrupteur  -Capteur pression  -Potentiomètre | -  +  -  - | +  +  -  - | --  X  --  -- |
| Tourner à gauche | Tout ou rien | -Bouton Poussoir  -Interrupteur  -Capteur pression  -Potentiomètre | +  -  +  - | +  +  -  - | X  --  --  -- |
| Tourner à droite | Tout ou rien | -Bouton Poussoir  -Interrupteur  -Capteur pression  -Potentiomètre | +  -  +  - | +  +  -  - | X  --  --  -- |
| Avancer | Tout ou rien | -Bouton Poussoir  -Interrupteur  -Capteur pression  -Potentiomètre | +  -  +  - | +  +  -  - | X  --  --  -- |
| Reculer | Tout ou rien | -Bouton Poussoir  -Interrupteur  -Capteur pression  -Potentiomètre | +  -  +  - | +  +  -  - | X  --  --  -- |
| Choisir la vitesse | Analogique | -Capteur pression  -Potentiomètre | -  + | -  + | --  X |

Combinaisons de moteur possibles par fonction :

Moteur à l’arrêt = (0)

Moteur dans le sens horaire = (1)

Moteur dans le sens anti-horaire = (-1)

Moteur côté Gauche = MG

Moteur côté Droite = MD

Vitesse minimale = Vmin

Vitesse maximale = Vmax

Potentiomètre = Pot avec des valeurs comprises entre 0 et 1024

|  |  |  |
| --- | --- | --- |
| FONCTION | Etat des moteurs | Etat du potentiomètre |
| Arrêt du moteur | /// | Arrêt (0) : Pot = [0 ; 30] |
| Arrêt du véhicule | MD = (0)  MG = (0) | /// |
| Tourner à gauche | MD = (-1)  MG = (-1) | /// |
| Tourner à droite | MD = (1)  MG = (1) | /// |
| Avancer | MD = (-1)  MG = (1) | /// |
| Reculer | MD = (1)  MG = (-1) | /// |
| Choisir la vitesse | /// | Vmin (1) : Pot = [11 ; 512]  Vmax (2) : Pot = [513 ; 1024] |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Choix de la batterie de la télécommande :**

On dispose d’une Arduino Uno, qui consomme d’après sa Datasheet 5V pour une capacité de 50 mA, soit 50 mAh. On prend en compte une légère marge afin d’alimenter l’antenne radiofréquence ainsi que les composants de la télécommande (bp, pot).

On dispose également de deux batteries de caractéristiques 3.7V-2000mAh

2000 mAh = 40 \* 50 mAh. On aura une autonomie d’environ 40h.

La capacité étant largement suffisante, on s’intéresse à la tension nécessaire.

Deux batteries branchées en série additionnent leur tension afin de fournir la somme des deux tensions au système. Ici on cherche une tension de 5V. Etant donné que les deux batteries ont une tension de 3.7V, on aura 7.4V si on les branche en série, ce qui est suffisant.

On utilisera donc ces deux batteries afin d’alimenter la télécommande du char.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mesures de la télécommande :**

On mesure tout d’abord les différents composants afin d’avoir une idée de la taille de la télécommande.

Arduino :

Largeur : 5,5cm

Longueur : 10cm

Hauteur (on prend en compte les fils électriques qui dépassent) : 5cm

Batterie (x2) :

Largeur : 5cm

Longueur : 7.3cm

Hauteur : 0.5cm

Bouton poussoir (x4) :

Largeur : 2cm

Longueur : 0.5cm

Hauteur : 1.7cm

Potentiomètre :

Largeur : 2cm

Longueur : 4cm

Hauteur : 4cm

Breadboard :

Adaptable en fonction de la place restante.

**Gestion de l’espace et répartition des éléments :**

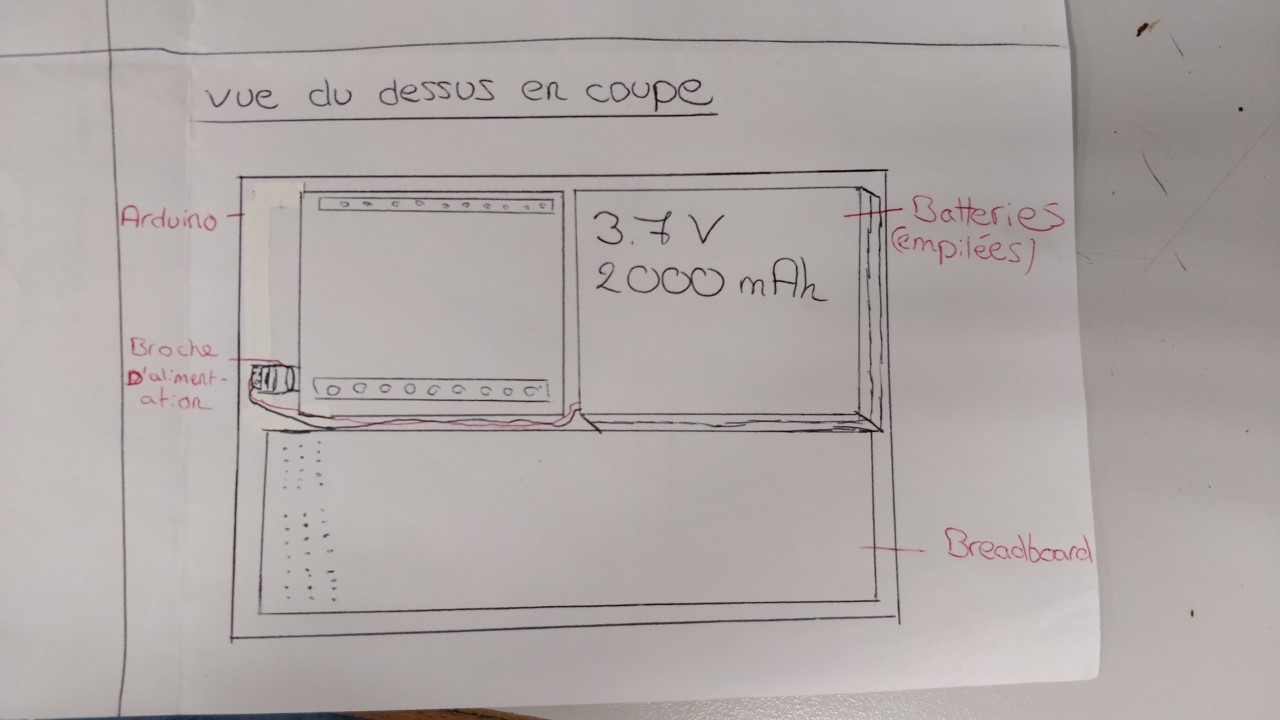
Pour fixer l’espace de la batterie : Système de porte + petite vis pour le socle batterie

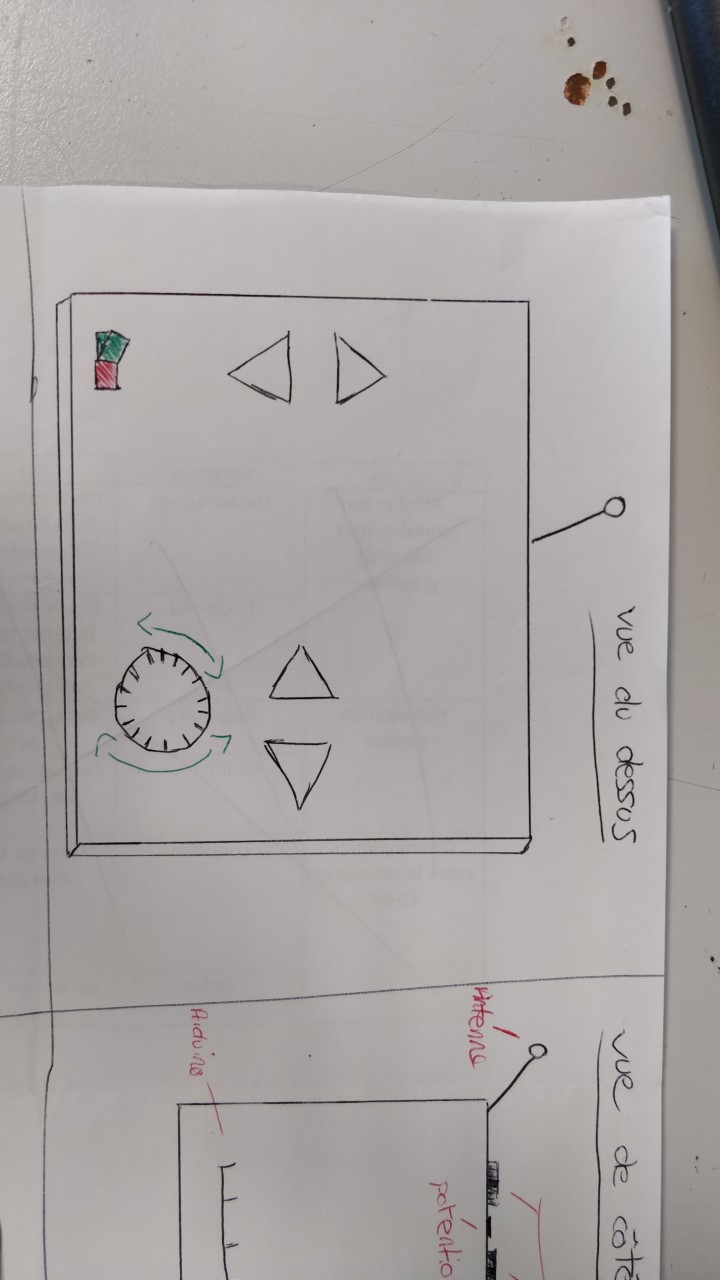
Pour les boutons : Petits trous avec socle (à construire).

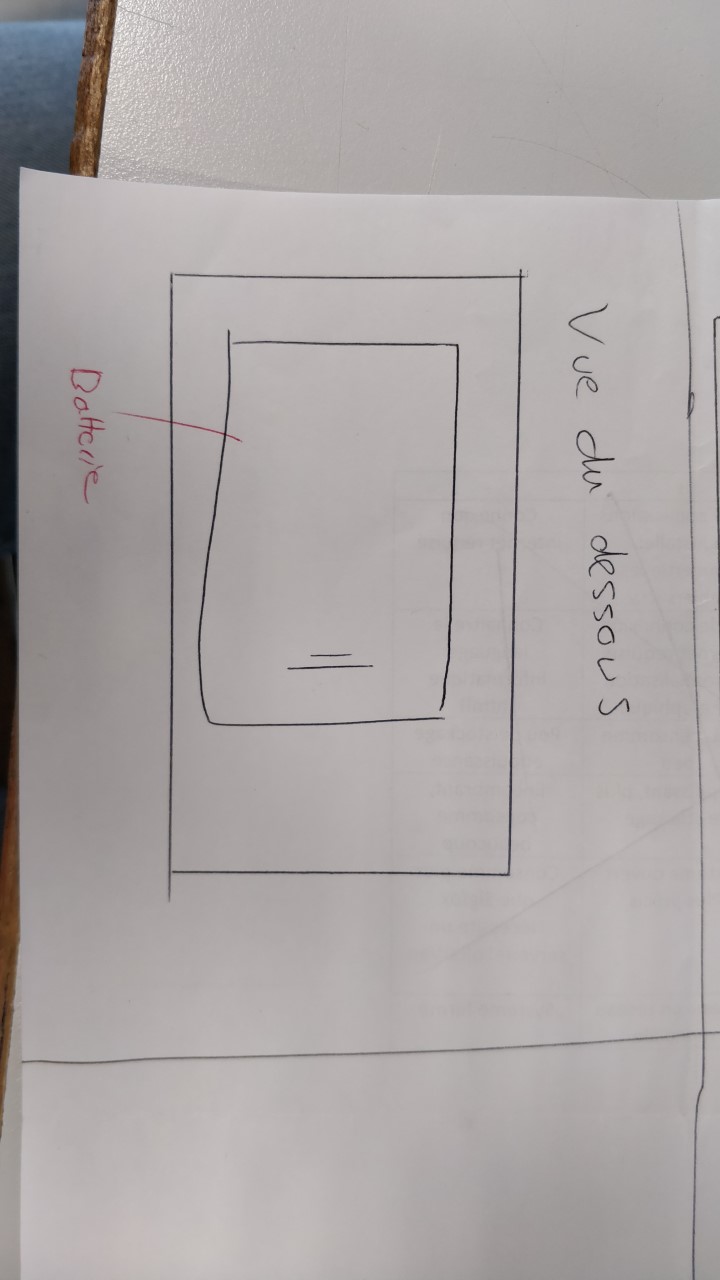
Pour le pot : 1- Laisser passer le bout du potentiomètre à tourner

2- Si y’a le temps : fabriquer un socle pour faciliter le tournage + marques sur la télécommande (0,1,2)

Voici plusieurs schémas de la télécommande :







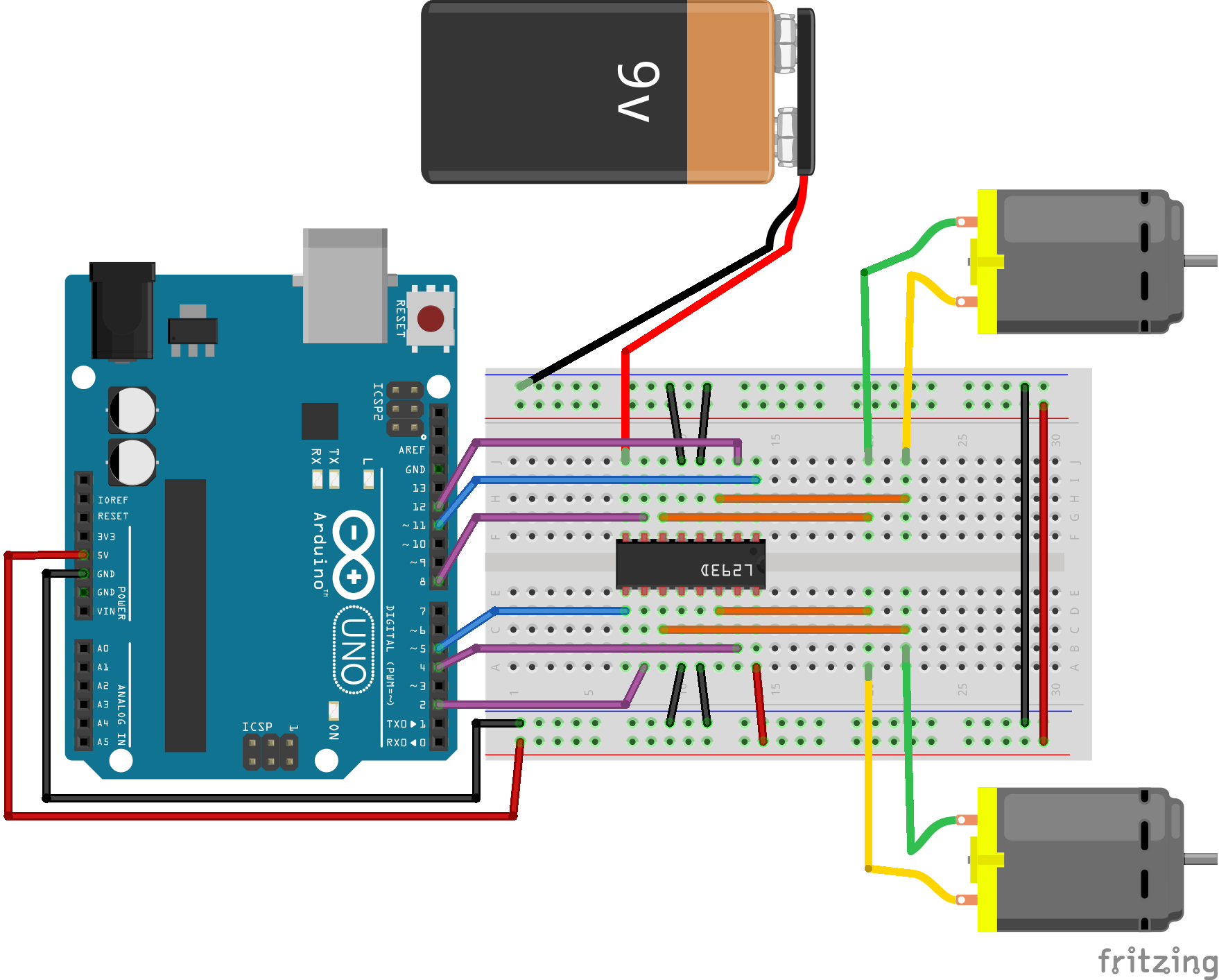
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

PWM : Pour contrôler la vitesse de rotation d’un moteur, on peut utiliser un PWM simuler une valeur analogique en numérique. Valeur numérique lu en valeur analogique.

Pont en H :

Les ponts en H Un Pont en H est un circuit électronique qui permet d’appliquer un voltage a une charge dans 2 directions différentes. Ce circuit est couramment utilisé en robotique et d’autres applications pour permettre aux moteurs DC de tourner dans deux sens. Le nom < pont en H > est vient de la position des éléments de commutation dans le circuit, places dans les quatre branches d’un H.

Moteur connectique :



Moteur : 12 V / 0,14A

P=U\*I = 12\*0,14=1.68W

1.68W \* 2H = 3.36Wh

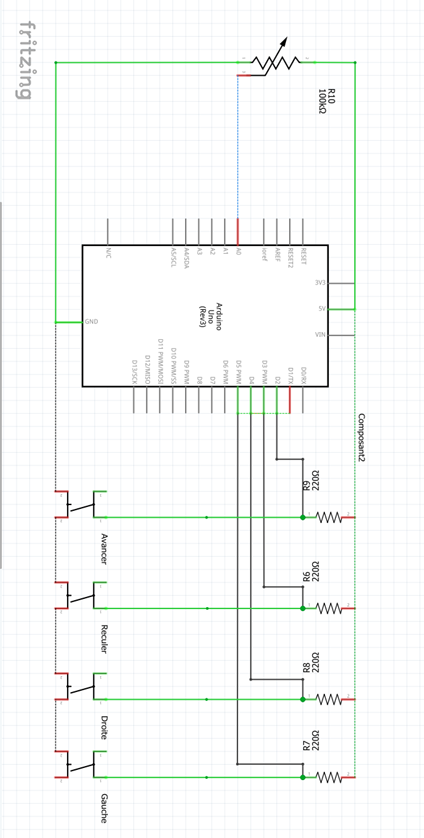
E=P\*S=1.68\*7200=12096 J

J en Wh = 3.36 Wh

Ah=Wh/U=3.36/12=0.28Ah

Moteur : JH25-310

Schéma du montage de la télécommande :

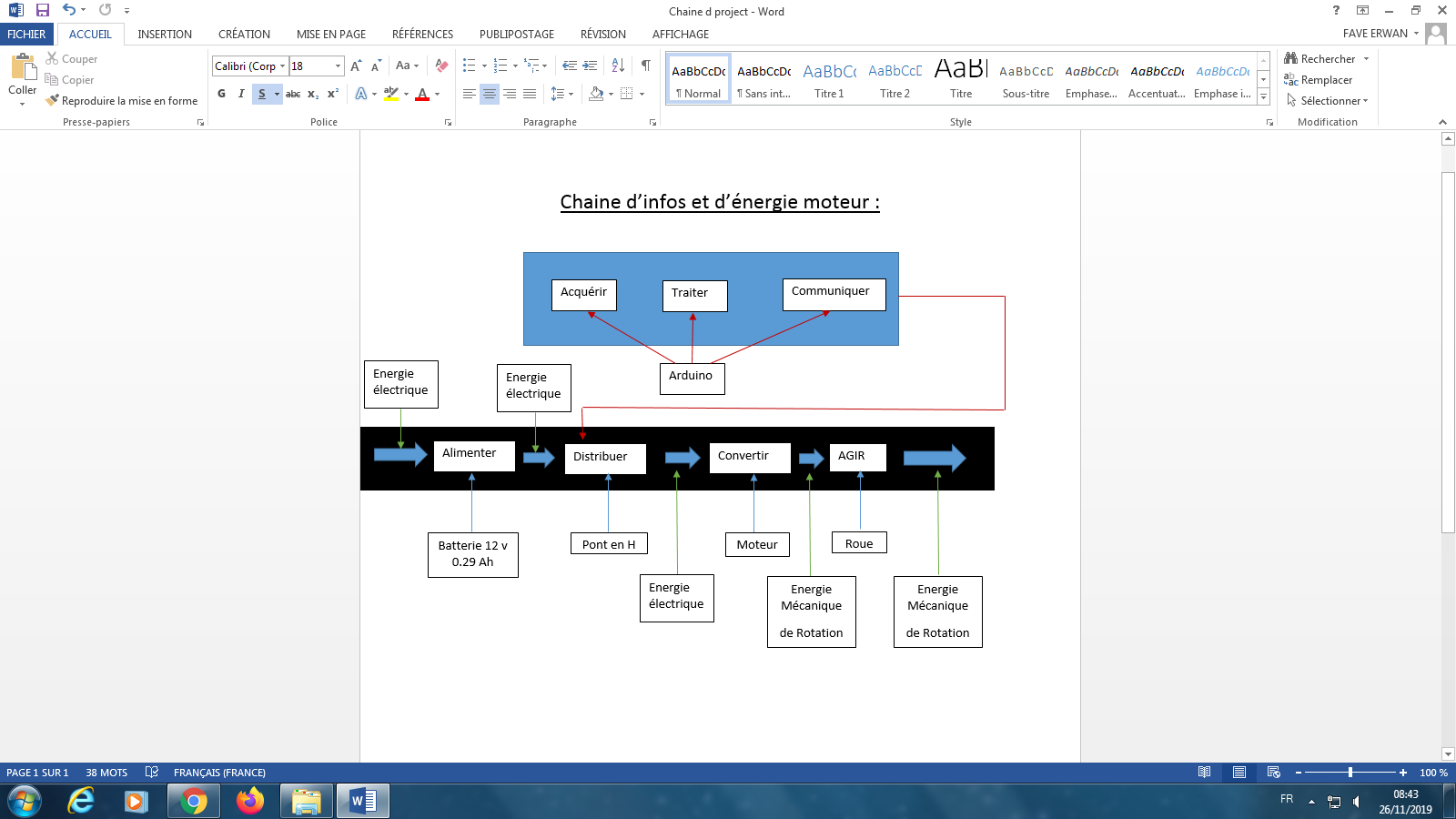


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FONCTION (COMPOSANT)** | **E/S** | **NOTATION ARDUINO** | **BROCHE ARDUINO** | **RÔLE** |
| Avancer (BP) | ENTREE | Int=avancer | Broche 4 | Faire avancer le char |
| Reculer (BP) | ENTREE | Int=reculer | Broche 5 | Faire reculer le char |
| Aller à gauche (BP) | ENTREE | Int=gauche | Broche 2 | Faire pivoter le char vers la gauche |
| Aller à droite (BP) | ENTREE | Int=droite | Broche 3 | Faire pivoter le char vers la droite |
| Adapter la vitesse (potentiomètre) | ENTREE | Int=pot | Broche A0 | Choisir la vitesse à laquelle on veut faire rouler le char |
| Transmettre l’info (Antenne RF d’entrée) | SORTIE | Int=erf | Broche 6 | Recevoir l’information de la télécommande pour l’envoyer à l'autre antenne |
| Recevoir une info (Antenne RF de sortie) | SORTIE | Int=srf |  | Recevoir l’information de la première antenne pour la transmettre au char |
| Avancer/ Reculer |  | Int=avancer | Arduino :  Pont en H:  Broche E1 et E2 | Faire avancer/reculer le char |
| Aller à gauche |  | Int=gauche | Pont en H:  Broche M1 | Faire pivoter le char vers la gauche |
| Aller à droite |  | Int=droite | Pont en H :  Broche | Faire pivoter le char vers la droite |
| Adapter la vitesse |  | Int=pot | Sur une sortie PWM | Choisir la vitesse à laquelle on veut faire rouler le char |

Définition des variables globales :

A COMPLETER AVEC LE DIAMETRE DE LA TETE DE CEDRIC

Chaine d’info et d'énergie de la partie moteur :

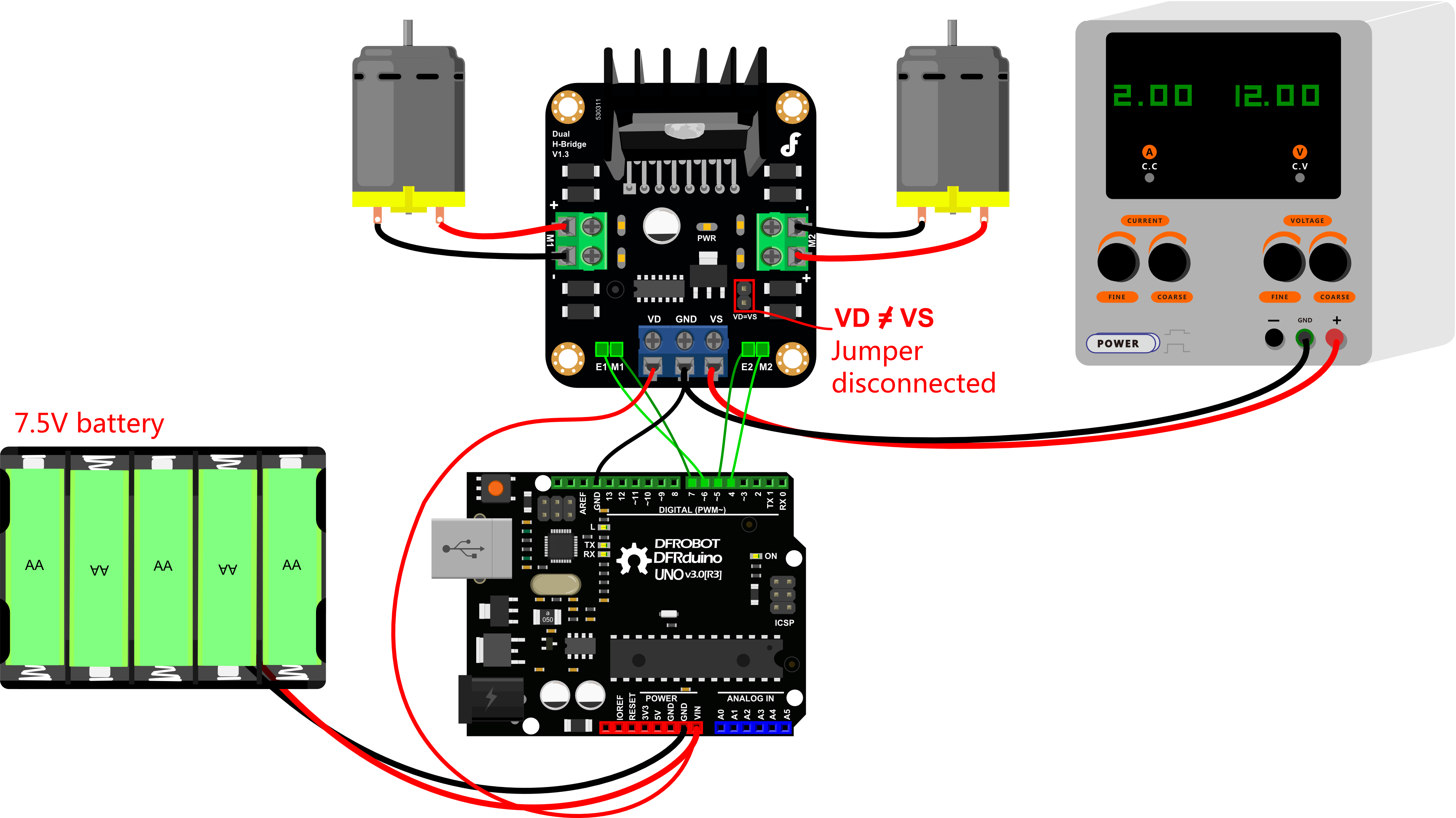


Matériel nécessaire :

|  |
| --- |
| **Partie Alan :** |
| * 4 Boutons poussoir |
| * 1 Potentiomètre |
| * 1 Carte Arduino |
| * Plaque d’essai |
| * 2 Batteries 3.7V-2000mAh |
| * 4 résistances |

|  |
| --- |
| **Partie Erwan :** |
| * 1 Batteries 12 V / 0,28 Ah (ou 3 batteries 4V) |
| * 1 Arduino |
| * Pont en H |
| * 2 Moteur 12 V / 0.14 A / 1.68 W |
| * Câbles électriques |

|  |
| --- |
| **Partie Elouan :** |
| * 1 Antenne radio fréquence 433 MHz |
| * Câbles électriques |



Code moteur :

int E1 = 12;//moteur 1

int M1 = 5;//

int E2 = 13;//moteur 2

int M2 = 6;//

int avancer = 1;

int reculer = 0;

int tournerIdroite = 0;

int tournerIgauche = 0;

void setup() {

pinMode(M1,OUTPUT);

pinMode(M2,OUTPUT);

}

void loop() {

if (tournerIgauche=1){

int value1 =0;

int value2=255;

digitalWrite(M1,HIGH);

digitalWrite(M2,HIGH);

analogWrite(E1,value1); //PWM Speed Control

analogWrite(E2,value2);//PWM Speed Control

delay(30);

}

if (tournerIdroite=1){

int value1 =255;

int value2=0;

digitalWrite(M1,HIGH);

digitalWrite(M2,HIGH);

analogWrite(E1,value1); //PWM Speed Control

analogWrite(E2,value2);//PWM Speed Control

delay(30);

}

if(reculer=1){

int value1 =0;

int value2=255;

digitalWrite(M1,HIGH);

digitalWrite(M2,HIGH);

analogWrite(E1,value1); //PWM Speed Control

analogWrite(E2,value2);//PWM Speed Control

delay(30);

}

if (avancer=1) {

int value1 =255;

int value2=0;

digitalWrite(M1,HIGH);

digitalWrite(M2,HIGH);

analogWrite(E1,value1); //PWM Speed Control

analogWrite(E2,value2);//PWM Speed Control

delay(30);

}

}

