Gate Project

IMERIR
INSTITUT MÉDITERRANÉEN
D'ÉTUDES ET DE RECHERCHE EN
INFORMATIQUE ET ROBOTIQUE

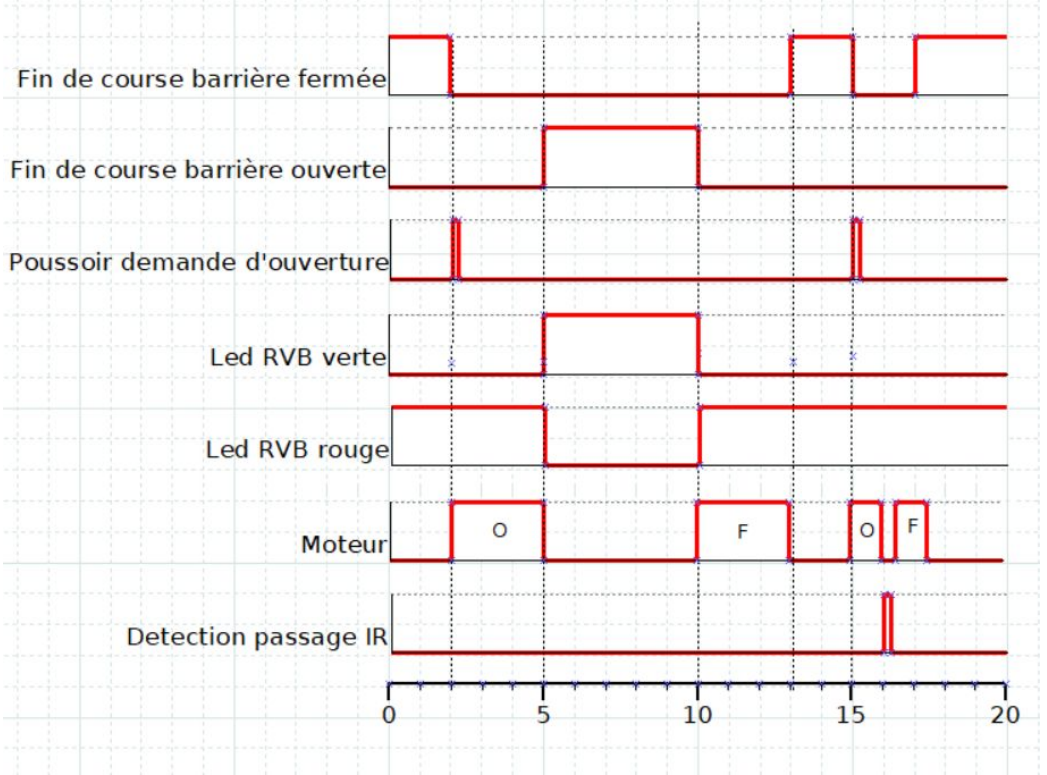
Summary

CDC

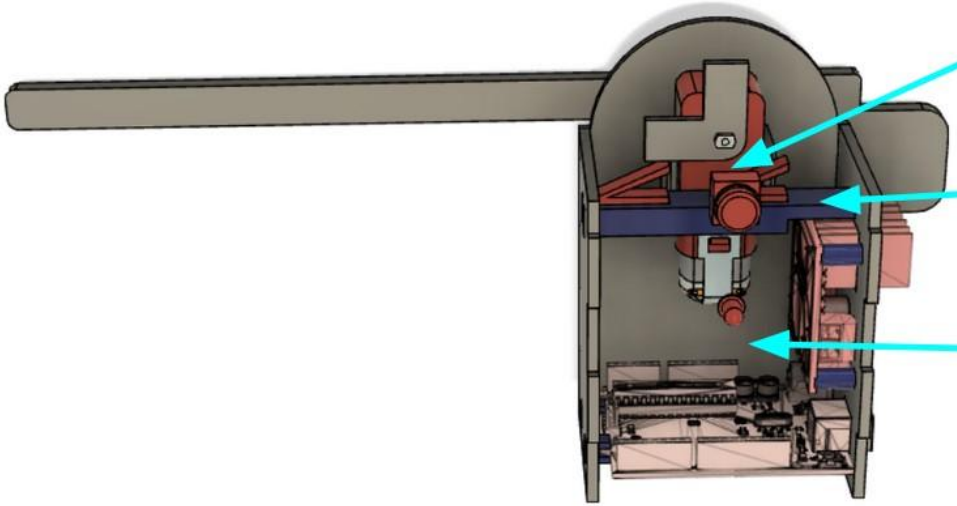
Make a gate for train crossroads

Mandatory material :
Arduino board
Motor driver
And/stop
RGB Led
Arduino yellow motor
Push button

Chronogram



Conception

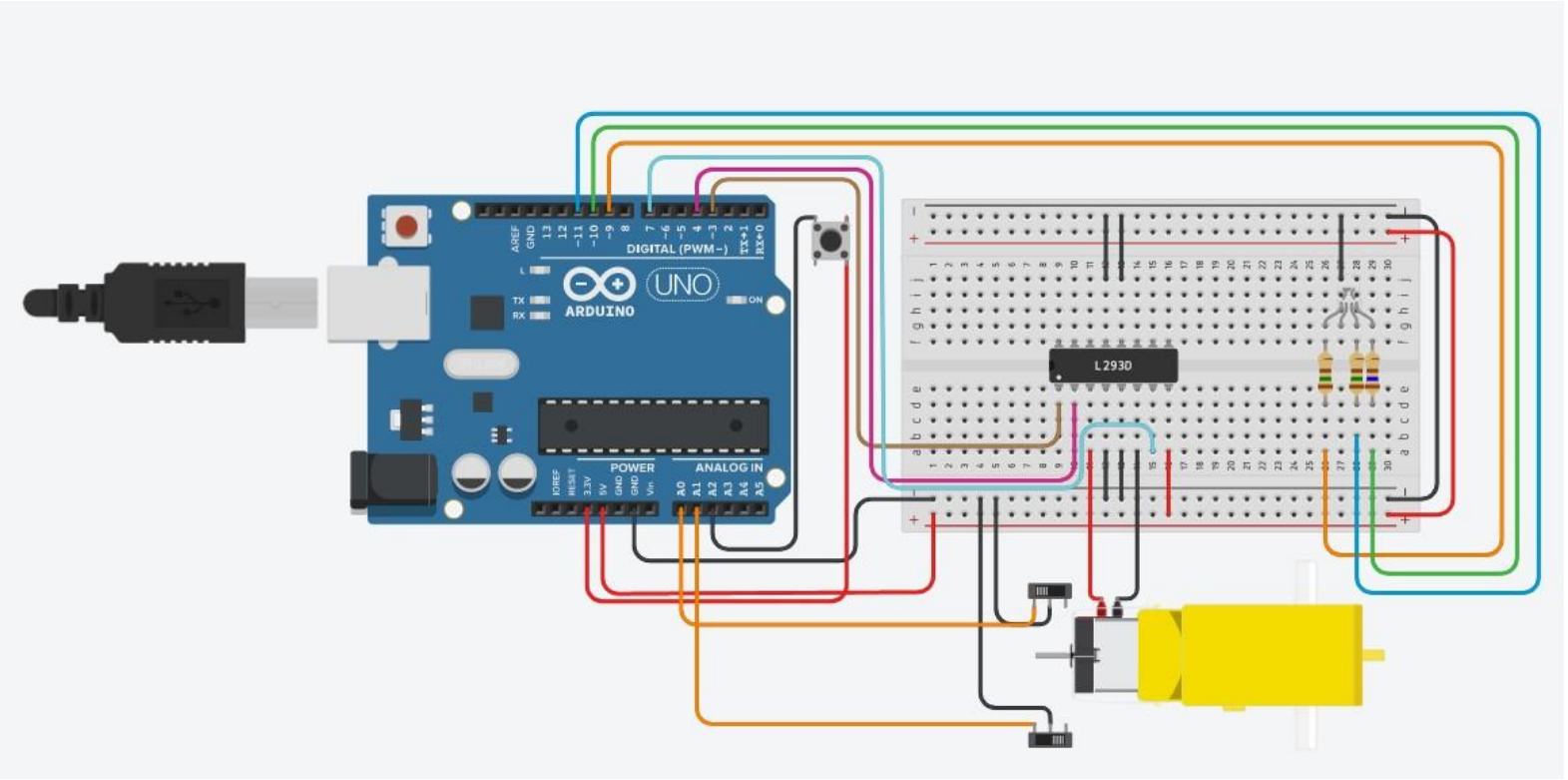


Red parts : Electronics

Blue parts : 3d printing

Grey parts : 3mm wood

Electric diagram



Coding

```
const int BP = A0; // broche 2 du micro-contrôleur se nomme maintenant : BP7
const int LedR=9;    //déclaration de la pin 9 pour led rouge
const int LedG=10;   //déclaration de la pin 10 pour led verte
const int LedB=11;   //déclaration de la pin 11 pour led bleue
const int L1 = 6; // broche 6 du micro-contrôleur se nomme maintenant : L1
const int StopUn = A1;
const int StopDeux = A2;
const int IN1=4;
const int IN2=7;
const int ENA=3;
int mouvement=0; //0 pour avant, 1 pour arriere
char directionB='arriere';
//int vitesse=255;
void setup() //fonction d'initialisation de la carte
{
//contenu de l'initialisation
  pinMode(BP, INPUT_PULLUP);
  pinMode(StopUn, INPUT_PULLUP);
  pinMode(StopDeux, INPUT_PULLUP);
  pinMode(LedR, OUTPUT);  //declare led rouge en sortie
  pinMode(LedG, OUTPUT);  //declare led verte en sortie
  pinMode(LedB, OUTPUT);  //declare led bleue en sortie
  pinMode(L1, OUTPUT); //L1 est une broche de sortie
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);
  autotest();             //test au demarage
  Serial.begin(9600);     //vitesse de comunication a 9600bauds
  delay(1000) ;            //delay
  Serial.println("test de com ");// ecriture
}
```

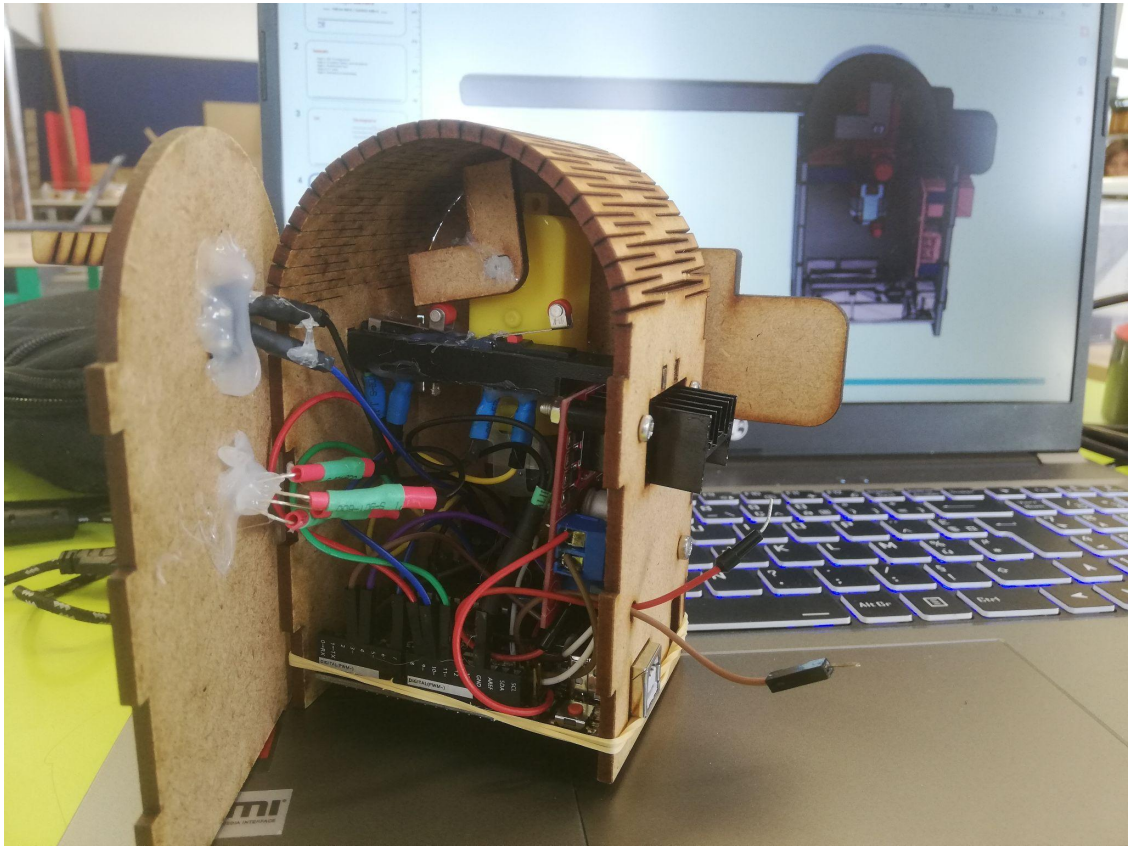
```
void loop() //fonction principale, elle se répète (s'exécute) à l'infini
{
//contenu du programme
int testActivation1 = digitalRead(BP); // Lecture de l'entree BP et sockage du résultats dans test
int testActivation2 = digitalRead(StopUn);
int testActivation3 = digitalRead(StopDeux);

if(testActivation2 == LOW) // Si test est à l'état bas
{
  FreinM();
  mouvement=1;
  couleur(255,0,0);
}
if(testActivation3 == LOW)
{
  FreinM();
  mouvement=0;
  couleur(0,0,255);
}
if (mouvement == 0) {
  delay(2000);
  couleur(255,0,0);
  avant();
  }

if (testActivation1 == LOW) {
  if (mouvement == 1 and testActivation2 == LOW) {
    arriere();
    }
  }
}
```

Coding

```
void avant(){
  digitalWrite(IN2,1);
  digitalWrite(IN1,0);
  analogWrite(ENA,70);
  delay(100);
  analogWrite(ENA, 10);
  delay(500);
  analogWrite(ENA, 60);
}

void arriere(){
  digitalWrite(IN2,0);
  digitalWrite(IN1,1);
  analogWrite(ENA,70);
  delay(100);
  analogWrite(ENA, 35);
}

void FreinM(){
  digitalWrite(ENA,0);
  digitalWrite(IN2,1);
  digitalWrite(IN1,1);
}

void couleur (int varLR,int varLG, int varLB){
analogWrite(LedR,varLR);
analogWrite(LedG,varLG);
analogWrite(LedB,varLB);
}

void stop(){
  analogWrite(LedB, 0);
  analogWrite(LedG, 0);
  analogWrite(LedR, 0);
```

```
void stop(){
  analogWrite(LedB, 0);
  analogWrite(LedG, 0);
  analogWrite(LedR, 0);
}

void autotest(){
  digitalWrite(LedR, HIGH);  //led rouge sur on
  delay(200);                // attente 200ms
  digitalWrite(LedR, LOW);   //led rouge sur on
  delay(200);                // attente 200ms
  digitalWrite(LedG, HIGH);  //led VERTE sur on
  delay(200);                // attente 200ms
   digitalWrite(LedG, LOW);  //led VERTE sur on
  delay(200);                // attente 200ms
  digitalWrite(LedB, HIGH);  //led BLEU sur on
  delay(200);                // attente 200ms
  digitalWrite(LedB, LOW);   //led BLEU sur on
  delay(200);                // attente 200ms
}
```

Building and assembly



Resume

Things that are successful

- Thing that can be improved