ATDTec
Association Tawassol
pour le Développement de la Technologie

الروبوت الإطفائي Robotech2018

مسابقة الروبوتيات التربوية دورة 2018

الملف التقني (بتصرف)

من إنجاز الفريق :

البروتي يحى محمد

الدريويش خولة

تأطير الأستاذ

عبدالجليل رواكدي

ثانوية فاطمة الفهرية الإعدادية

الأكاديمية الجهوية للتربية والتكوين جهة مراكش آسفي

المديرية الإقليمية بمراكش

Pour toute autre information, veuillez contacter Mr Abdeljalil Rwagdi

quermodi@hotamil.com

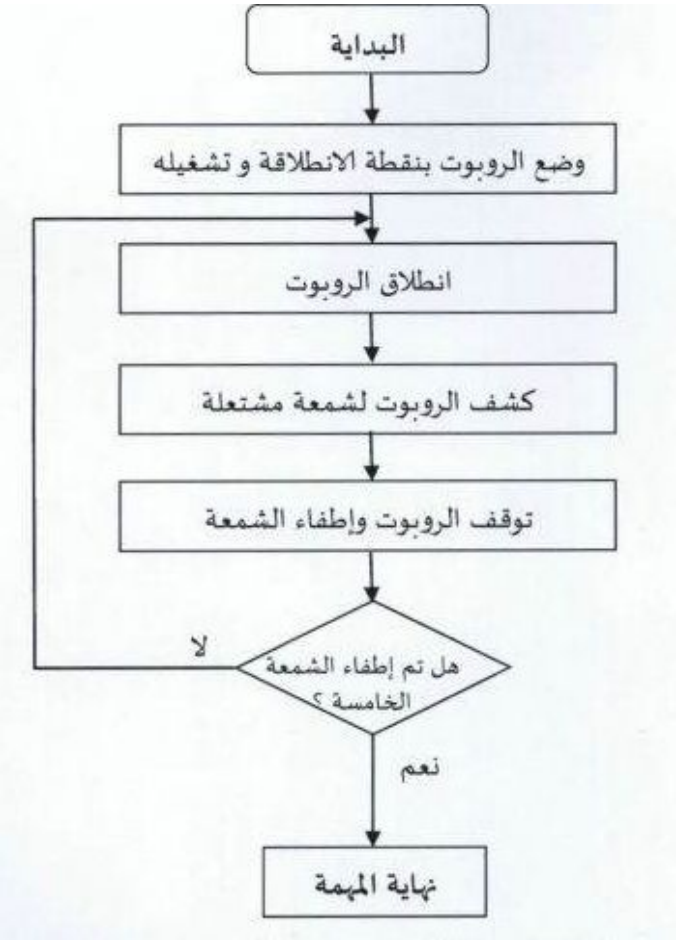
الرتبة الثالثة

ماي 2018

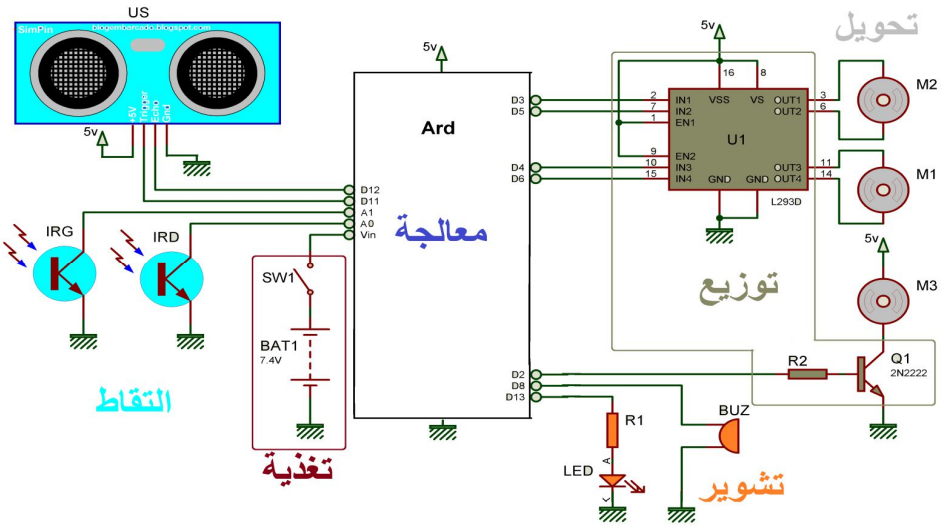
الروبوت الإطفائي

1. المقاربة العلمية و التقنية :

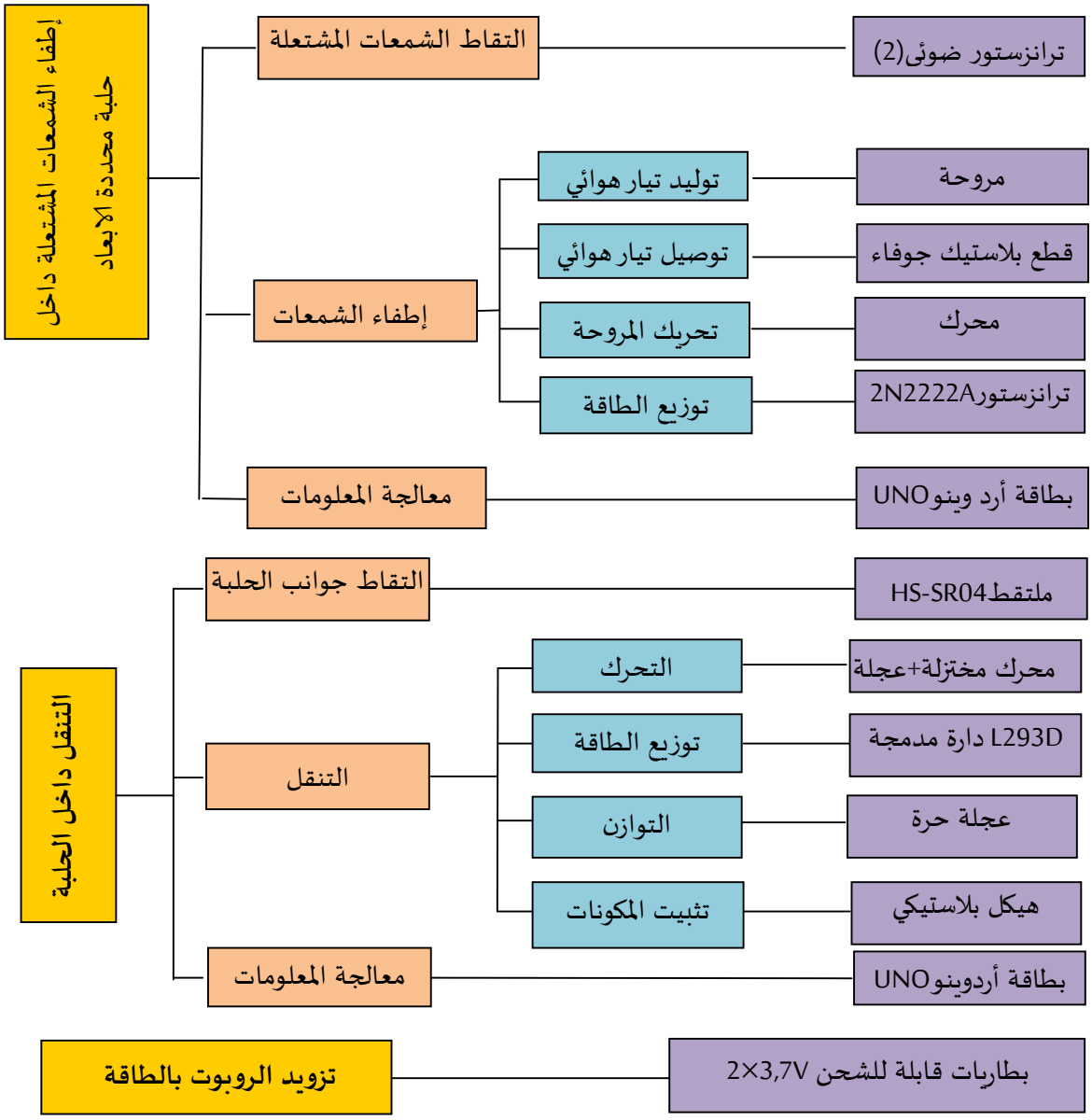
2.1 المبيان التنظيمي للروبوت الاطفائي :



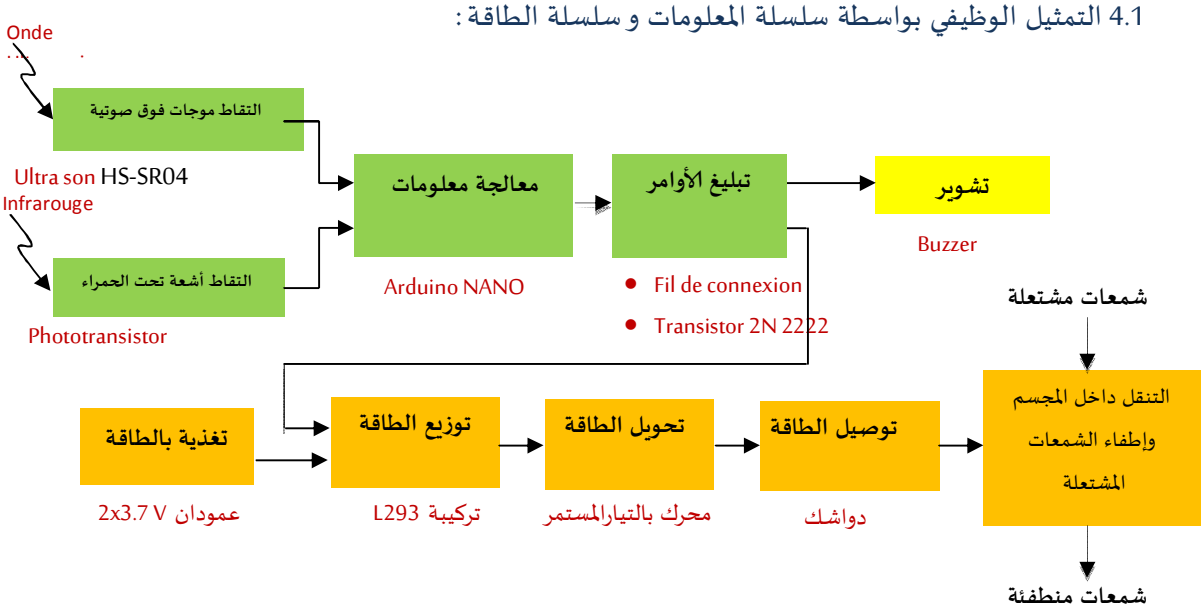
2.1 الرشم البنيوي لتركيبة التحكم في الروبوت :



3.1 فاست الوصفي Le FAST descriptif

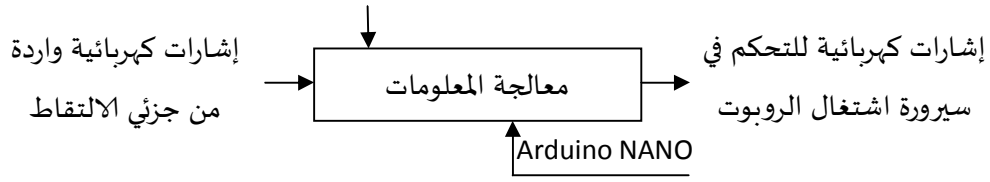


4.1 التمثيل الوظيفي بواسطة سلسلة المعلومات و سلسلة الطاقة :



5.1 دراسة الأجزاء المكونة للمنظم التقني

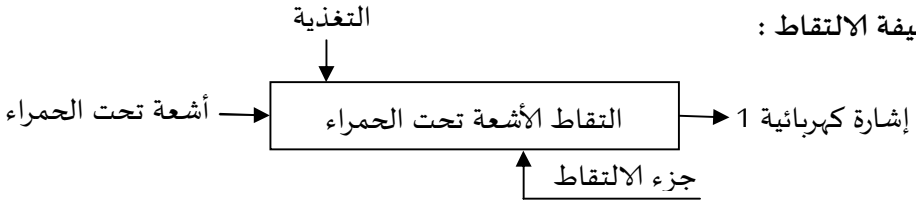
1. وظيفة المعالجة :



أسباب اختيار Arduino NANO :

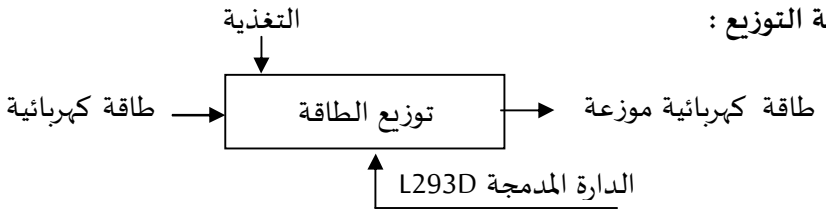
- **زحمة مختزلة ؛**
- **سهولة الاستعمال ؛**
- **إمكانية البرمجة بطرق مختلفة بسيطة بالنسبة للمتعلمة والمتعلم ؛**
- **ثمن معقول ومناسب .**

2. وظيفة الالتقاط :



أسباب اختيار الترانزيستور الضوئي :لالتقاط الشمعة المشتعلة ، اخترنا الترانزستور الضوئي كملتقط بفعل حساسيته المرتفعة للأشعة تحت الحمراء دون التأثر بإضاءة المحيط وسهولة تركيبه مع بطاقة الأردوينو باستغلال المقاومة الداخلية بالمربط التناظري للبطاقة la résistance interne'PULL UP -.

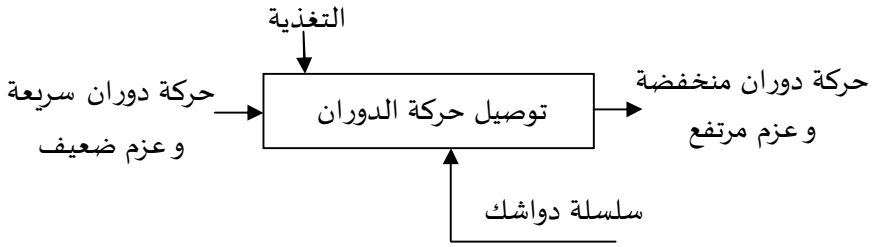
3. وظيفة التوزيع :



ملاحظة : مبررات اختيار تركيبة L293D لتوزيع الطاقة عوض المناب :

مناب6VRelais	الدارة المدمجة L293D
حل جيد . لكن ، لتطبيقه نحتاج مناب لكل محرك إضافة لترانزستور وثنبيل ذو وصل لتبليغ آمرة التوزيع للمناب مما سيزيد من زحمة التركيبة .	يوفر تيار كاف لتشغيل المحركين في منحيين مع إمكانية فصل تغذية المحرك عن تغذية باقي التركيبة .

4. توصيل الحركة



ملاحظة : خلال مرحلة التجارب ، اخترنا محرك-مختزلة(Moteur-Réducteur) الموجود بمعدات الإعدادية ، لكن المحرك يتطلب شدة تيار تصل الى 600mA عند وجود حمولة ، لذا قمنا بتعويضه بمحرك قارئ الاقراص المدمجة مع الاحتفاظ بالمختزلة الموجودة به وإدخال عليها بعض التعديلات .

5. إطفاء الشمعات المشتعلة :

لقد تم اعتماد مروحة لمنظم التبريد المستعملة في الحاسوب المحمول نظرا لانخفاض توتر اشتغالها(5V – 0,34A) . ولتوجيه التيار الهوائي نحو الشمعة المشتعلة ، اعتمدنا على قطع بلاستيكية فارغة من الداخل(أنظر الصورة جانبه) .

محرك التحكم في المروحة :

المميزات : 5v شدة التيار أثناء تدوير المروحة 340 mA

مميزات بطاقة الأردو ينو (مداخل/مخارج تناظرية) : I_{oh}=40mA و V_{oh}= 5v

مميزات الترانزستور KSP2222A : Ic_{max}=600mA - βmin=100 - Vce_{sat}=1v - $Vbe_{(Ic=500mA)}$=2v

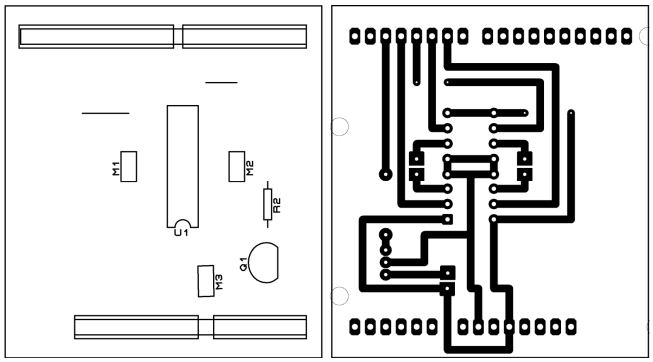
حساب قيمة مقاومة قاعدة الترانزستورQ1 (R2) :

في بداية اشباع الترانزستور تبقى العلاقة Ic=β.Ib صحيحة شريطة استعمال βmin بحيث تصبح Icsat=βmin.Ibsat ومنه: Ibsat=Icsat/βmin تطبيق: Ibsat=500/100=5mA

بالنسبة لدارة دخلة الترانزستور، لدينا V_{oh}=R2.Ib+Vbe و منه R2=(V_{oh}-Vbe)/Ib

تطبيق R2=(5-2)/(5.10^{-3})=600Ω اذن يمكن استعمال القيمة التالية : R2=680Ω

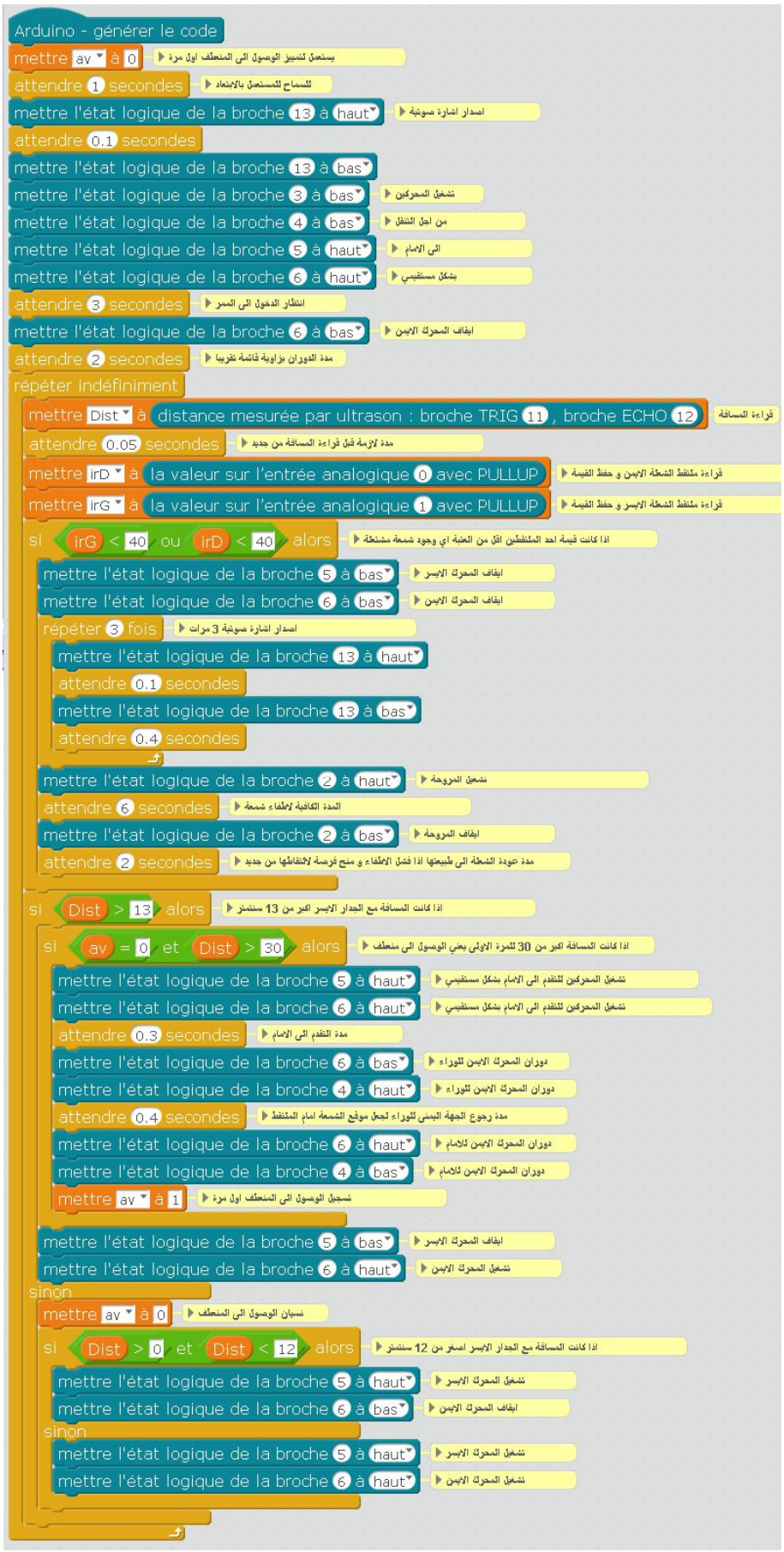
6.1 الدارة المطبوعة :

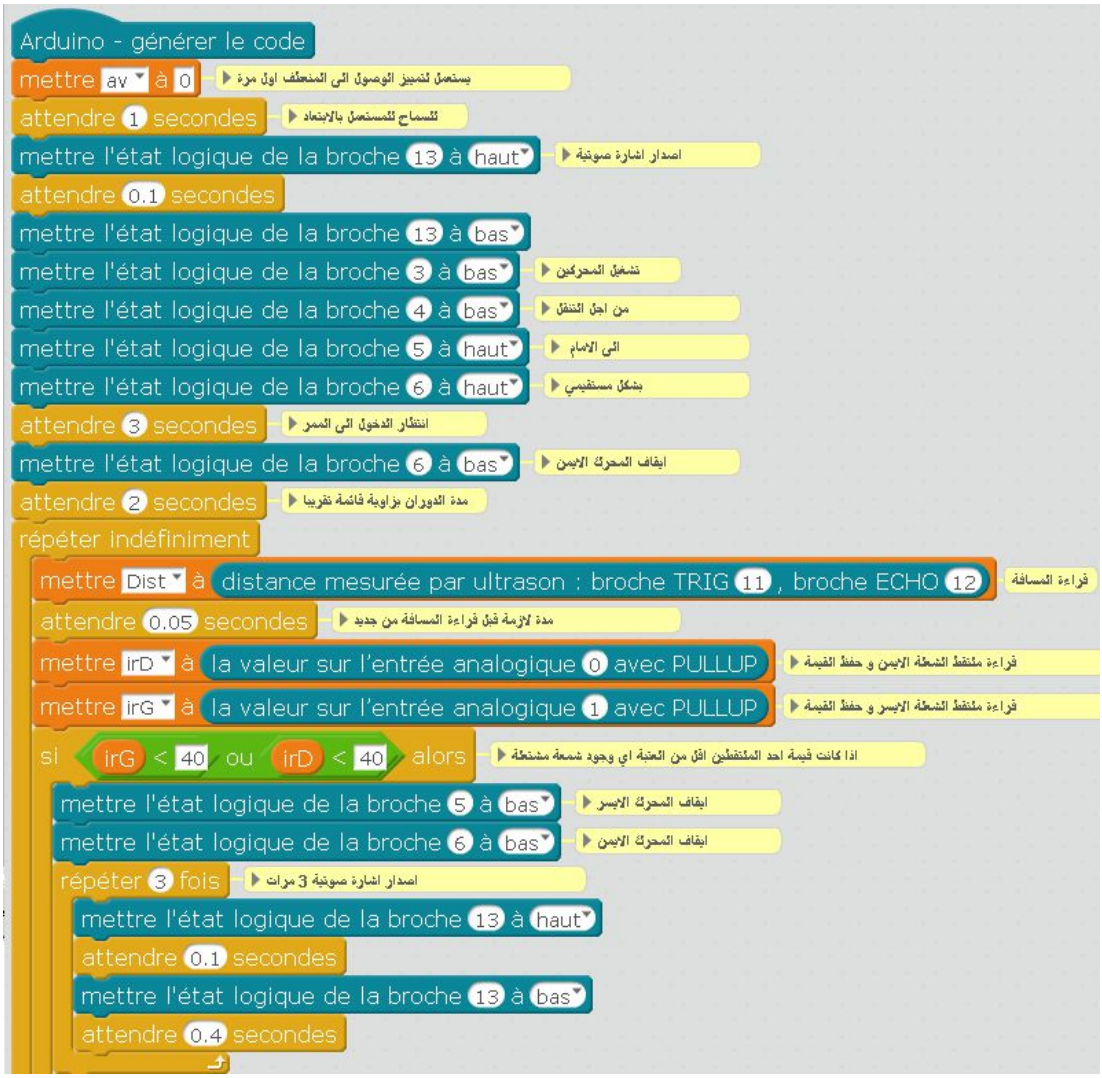


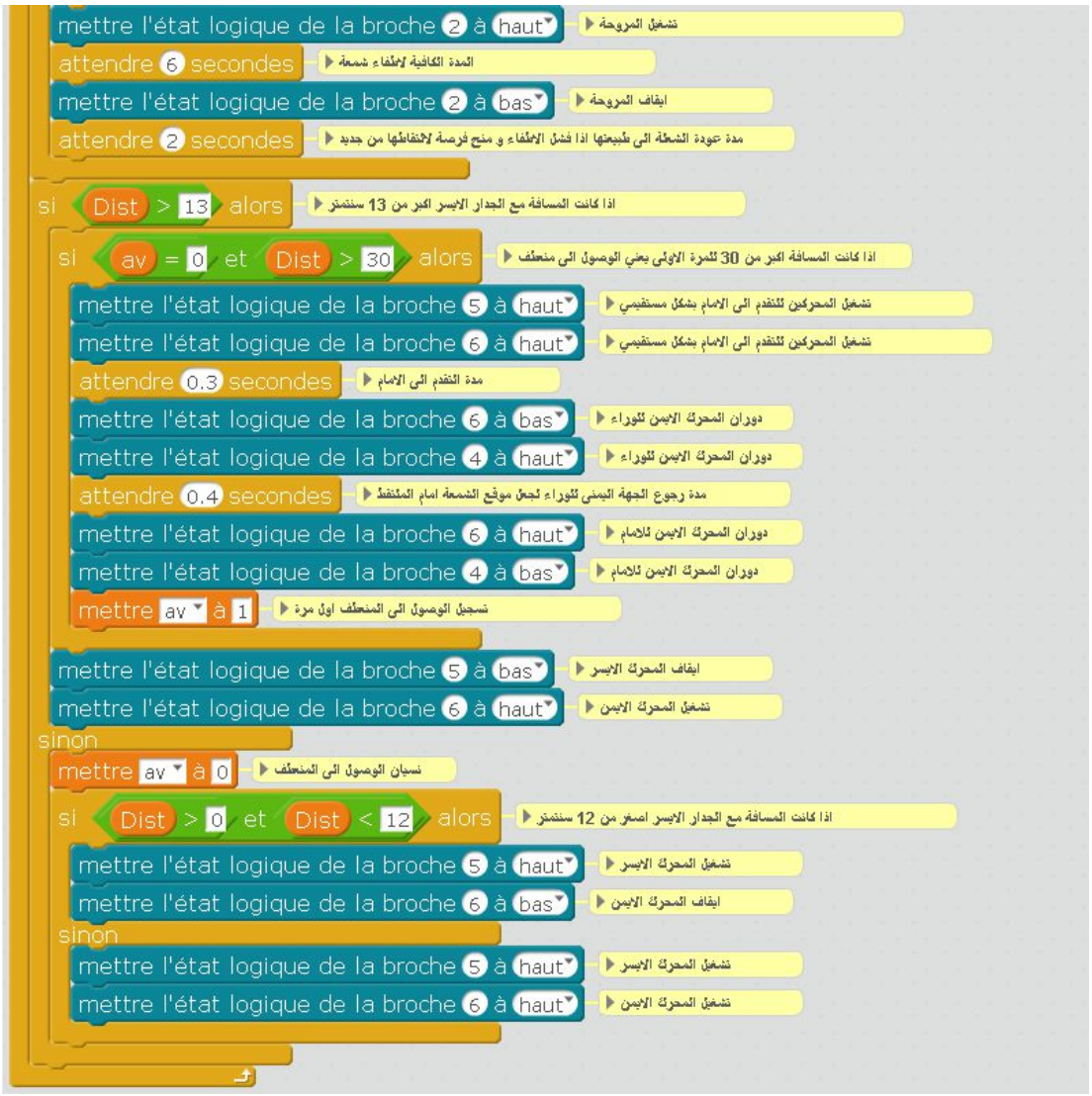
7.1 جدول المركبات :

الترتيب	الرمز	التعيين	العدد	المميزات
11	M1 ;M2	محرك كهربائي	2	5v- 120mA
10	M3	محرك . مروحة	1	5V-340mA
9	T1	ترانزستور	1	2N2222
8	R2	مقاومة	1	680 ohm 1/4w
7	R1	مقاومة	1	120 ohm 1/4w
6	U1	دارة مدمجة	1	L293D
5	BUZ	buzzer actif	1	9V
4	LED	ثنبيل متألق كهربائيا	1	5mm اصفر
3	ARD	بطاقة اردوينو	1	Arduino UNO
2	IRG;IRD	ترانزستور ضوئي	2	-
1	US	ملتقط بالموجات فوق الصوتية	1	HC-SR04

ملحق : البرنامج




Arduino - générer le code
mettre av à 0
يستعمل لتمييز الوصول الى المنعطف اول مرة
attendre 1 secondes
للسماح للمستعمل بالابتعاد
mettre l'état logique de la broche 13 à haut
اصدار اشارة صوتية
attendre 0.1 secondes
mettre l'état logique de la broche 13 à bas
mettre l'état logique de la broche 3 à bas
تشغيل المحركين
mettre l'état logique de la broche 4 à bas
من اجل التنقل
mettre l'état logique de la broche 5 à haut
الى الامام
mettre l'état logique de la broche 6 à haut
بشكل مستقيمي
attendre 3 secondes
انتظار الدخول الى الممر
mettre l'état logique de la broche 6 à bas
ايقاف المحرك الايمن
attendre 2 secondes
مدة الدوران بزاوية قائمة تقريبا
répéter indéfiniment
mettre Dist à distance mesurée par ultrason : broche TRIG 11 , broche ECHO 12
قراءة المسافة
attendre 0.05 secondes
مدة لازمة قبل قراءة المسافة من جديد
mettre irD à la valeur sur l'entrée analogique 0 avec PULLUP
قراءة مستقبل الضعثة الايمن و حفظ القيمة
mettre irG à la valeur sur l'entrée analogique 1 avec PULLUP
قراءة مستقبل الضعثة الايسر و حفظ القيمة
si irG < 40 ou irD < 40 alors
اذا كانت قيمة احد المستقبلين اقل من العتبة اي وجود شمعة مشتعلة
mettre l'état logique de la broche 5 à bas
ايقاف المحرك الايسر
mettre l'état logique de la broche 6 à bas
ايقاف المحرك الايمن
répéter 3 fois
اصدار اشارة صوتية 3 مرات
mettre l'état logique de la broche 13 à haut
attendre 0.1 secondes
mettre l'état logique de la broche 13 à bas
attendre 0.4 secondes


mettre l'état logique de la broche 2 à haut
تشغيل المروحة
attendre 6 secondes
المدة الكافية لإطفاء شمعة
mettre l'état logique de la broche 2 à bas
ايقاف المروحة
attendre 2 secondes
مدة عودة الشمعة الى طبيعتها اذا فشل الاطفاء و منح فرصة لاختفائها من جديد
si Dist > 13 alors
اذا كانت المسافة مع الجدار الايسر اكبر من 13 سنتمتر
si av = 0 et Dist > 30 alors
اذا كانت المسافة اكبر من 30 للمرة الاولى يعني الوصول الى منعطف
mettre l'état logique de la broche 5 à haut
تشغيل المحركين للتقدم الى الامام بشكل مستقيمي
mettre l'état logique de la broche 6 à haut
تشغيل المحركين للتقدم الى الامام بشكل مستقيمي
attendre 0.3 secondes
مدة التقدم الى الامام
mettre l'état logique de la broche 6 à bas
دوران المحرك الايمن للوراء
mettre l'état logique de la broche 4 à haut
دوران المحرك الايمن للوراء
attendre 0.4 secondes
مدة رجوع الجهة اليمنى للوراء لجعل موقع الشمعة امام المنتفظ
mettre l'état logique de la broche 6 à haut
دوران المحرك الايمن للامام
mettre l'état logique de la broche 4 à bas
دوران المحرك الايمن للامام
mettre av à 1
تسجيل الوصول الى المنحلف اول مرة
mettre l'état logique de la broche 5 à bas
ايقاف المحرك الايسر
mettre l'état logique de la broche 6 à haut
تشغيل المحرك الايمن
sinon
mettre av à 0
نسيان الوصول الى المنحلف
si Dist > 0 et Dist < 12 alors
اذا كانت المسافة مع الجدار الايسر اصغر من 12 سنتمتر
mettre l'état logique de la broche 5 à haut
تشغيل المحرك الايسر
mettre l'état logique de la broche 6 à bas
ايقاف المحرك الايمن
sinon
mettre l'état logique de la broche 5 à haut
تشغيل المحرك الايسر
mettre l'état logique de la broche 6 à haut
تشغيل المحرك الايمن

```ino
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>
double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
double av;
double Dist;
double irD;
double irG;
float getDistance(int trig,int echo){
    pinMode(trig,OUTPUT);
    digitalWrite(trig,LOW);
    delayMicroseconds(2);
    digitalWrite(trig,HIGH);
    delayMicroseconds(10);
    digitalWrite(trig,LOW);
    pinMode(echo, INPUT);
    return pulseIn(echo,HIGH,30000)/58.0;
}
void setup(){
    av = 0;
    _delay(1);
    pinMode(13,OUTPUT);
    pinMode(3,OUTPUT);
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);
    pinMode(6,OUTPUT);
    pinMode(A0+0,INPUT_PULLUP);
    pinMode(A0+1,INPUT_PULLUP);
    pinMode(2,OUTPUT);
    digitalWrite(13,1);
    _delay(0.1);
    digitalWrite(13,0);
    digitalWrite(3,0);
    digitalWrite(4,0);
    digitalWrite(5,1);
    digitalWrite(6,1);
    _delay(3);
    digitalWrite(6,0);
    _delay(2);
}
void loop(){
    Dist = getDistance(11,12);
    _delay(0.05);
    irD = analogRead(A0+0);
    irG = analogRead(A0+1);
    if(((irG) < (40)) || ((irD) < (40))){
        digitalWrite(5,0);
        digitalWrite(6,0);
        for(int __i__=0;__i__<3;++__i__)
        {
            digitalWrite(13,1);
            _delay(0.1);
            digitalWrite(13,0);
            _delay(0.4);
        }
        digitalWrite(2,1);
        _delay(6);
        digitalWrite(2,0);
        _delay(2);
    }
    if((Dist) > (13)){
        if((((av)==(0))) && ((Dist) > (30))){
            digitalWrite(5,1);
            digitalWrite(6,1);
            _delay(0.3);
            digitalWrite(6,0);
            digitalWrite(4,1);
            _delay(0.4);
            digitalWrite(6,1);
            digitalWrite(4,0);
            av = 1;
        }
        digitalWrite(5,0);
        digitalWrite(6,1);
    }else{
        av = 0;
        if(((Dist) > (0)) && ((Dist) < (12))){
            digitalWrite(5,1);
            digitalWrite(6,0);
        }else{
            digitalWrite(5,1);
            digitalWrite(6,1);
        }
    }
}
void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}
```

```ino
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>
double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
double av;
double Dist;
double irD;
double irG;
float getDistance(int trig,int echo){
    pinMode(trig,OUTPUT);
    digitalWrite(trig,LOW);
    delayMicroseconds(2);
    digitalWrite(trig,HIGH);
    delayMicroseconds(10);
    digitalWrite(trig,LOW);
    pinMode(echo, INPUT);
    return pulseIn(echo,HIGH,30000)/58.0;
}
void setup(){
    av = 0;
    _delay(1);
    pinMode(13,OUTPUT);
    pinMode(3,OUTPUT);
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);
    pinMode(6,OUTPUT);
    pinMode(A0+0,INPUT_PULLUP);
    pinMode(A0+1,INPUT_PULLUP);
    pinMode(2,OUTPUT);
    digitalWrite(13,1);
    _delay(0.1);
    digitalWrite(13,0);
    digitalWrite(3,0);
    digitalWrite(4,0);
    digitalWrite(5,1);
    digitalWrite(6,1);
    _delay(3);
    digitalWrite(6,0);
    _delay(2);
}
```

```
void loop(){
    Dist = getDistance(11,12);
    _delay(0.05);
    irD = analogRead(A0+0);
    irG = analogRead(A0+1);
    if(((irG) < (40)) || ((irD) < (40))){
        digitalWrite(5,0);
        digitalWrite(6,0);
        for(int __i__=0;__i__<3;++__i__)
        {
            digitalWrite(13,1);
            _delay(0.1);
            digitalWrite(13,0);
            _delay(0.4);
        }
        digitalWrite(2,1);
        _delay(6);
        digitalWrite(2,0);
        _delay(2);
    }
    if((Dist) > (13)){
        if((((av)==(0))) && ((Dist) > (30))){
            digitalWrite(5,1);
            digitalWrite(6,1);
            _delay(0.3);
            digitalWrite(6,0);
            digitalWrite(4,1);
            _delay(0.4);
            digitalWrite(6,1);
            digitalWrite(4,0);
            av = 1;
        }
        digitalWrite(5,0);
        digitalWrite(6,1);
    }else{
        av = 0;
        if(((Dist) > (0)) && ((Dist) < (12))){
            digitalWrite(5,1);
            digitalWrite(6,0);
        }else{
            digitalWrite(5,1);
            digitalWrite(6,1);
        }
    }
}
void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}
```