Association Tawassol
pour le Développement de la Technologie

مسابقة الروبوتيات التربوية دورة 2018

الملف التقني (بتصرف)

الأستاذ المؤطر : بومدين جابري

أعضاء الفريق التلاميدي : عماد العياشي وحمزة المختوم

ثانوية عبد الخالق الطريس الإعدادية

الأكاديمية الجهوية للتربية والتكوين بالجهة الشرقية

المديرية الإقليمية وجدة

السنة الدراسية 2017-2018

Pour toute information, veuillez contacter Mr Boumediane Jabri

cyjabritec@gmail.com

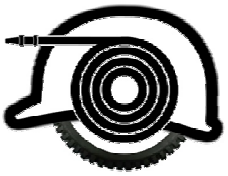
الروبوت الإطفائي Robotech2018

الرتبة الثانية

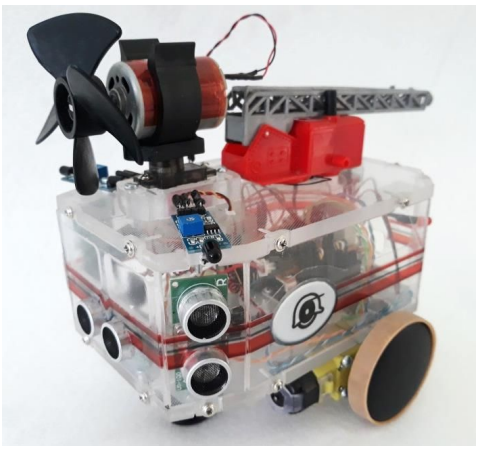
ماي 2018

تقديم :

تأتي هذه الدراسة الخاصة بمشروع "**الروبوت الإطفائي**" في إطار المسابقة الوطنية التلاميذية المؤطرة في الروبوتات التربوية في نسختها الخامسة (دورة 2018) والتي تهدف بالأساس تمكين المتعلمات والمتعلمين من تعرف مجال الروبوتيات وتفتح أمامهم فرصة الاطلاع على التكنولوجيات الحديثة . وعليه ، تناولت الدراسة ثلاث مقاربات (وظيفية وعلمية وتقنية) وذلك في إطار تنفيذ المقرر الدراسي لمادة التكنولوجيا الصناعية .



مسبوكة الروبوت



صورة الروبوت

1. مبدأ الاشتغال :

ينطلق الروبوت ليقطع الحلبة المستطيلة الشكل باستقلالية تامة شريطة عدم التماس بجوانبها (التي تمثل العائق) والتوقف عند كل شمعة مشتعلة (يمينا أو شمالا) بعد التقاطها والتوجه نحوها لإطفائها . وتتحقق هذه المهمة من خلال الوظائف الأساسية التالية :

أ. وظيفة التقاط الشمعة المشتعلة :

يبعث لهب شعلة الشمعة أنواعا مختلفة من الأشعة منها فوق البنفسجية (ultra-violet) والأشعة تحت الحمراء (infrarouge) التي يمتد طول موجتها بين 760nm و 2500nm (تحت الحمراء الأقرب إلى الأشعة المرئية) . سنعتمد التقاط الأشعة تحت الحمراء باستعمال ملتقطين للأشعة تحت الحمراء أحدهما على اليمين والآخر على اليسار .

ب. وظيفة إطفاء الشمعة المشتعلة :

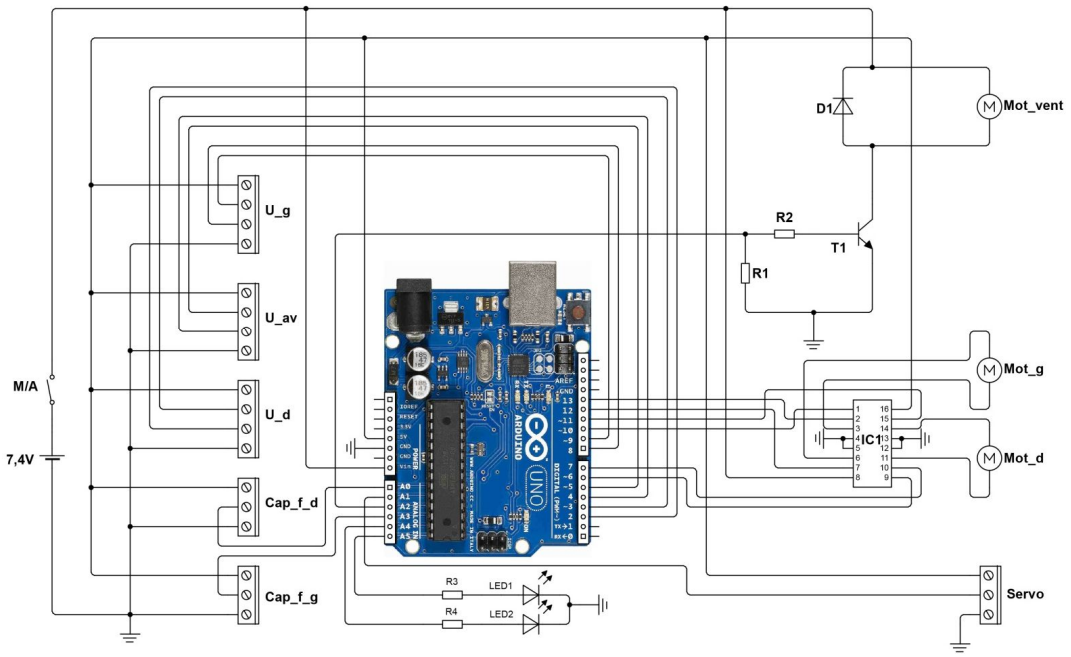
تم استعمال مروحة يتحكم في دورانها محرك كهربائي بحيث تولد المروحة تيارا هوائيا بقوة كافية لإطفاء لهب الشمعة سواء كانت الشعلة من جهة اليمين أو جهة اليسار تبعا لجهة الالتقاط . يتم توجيه المروحة بمحرك مؤازر بزاوية مناسبة نحو الشمعة المشتعلة .

ج- وظيفة التشوير:

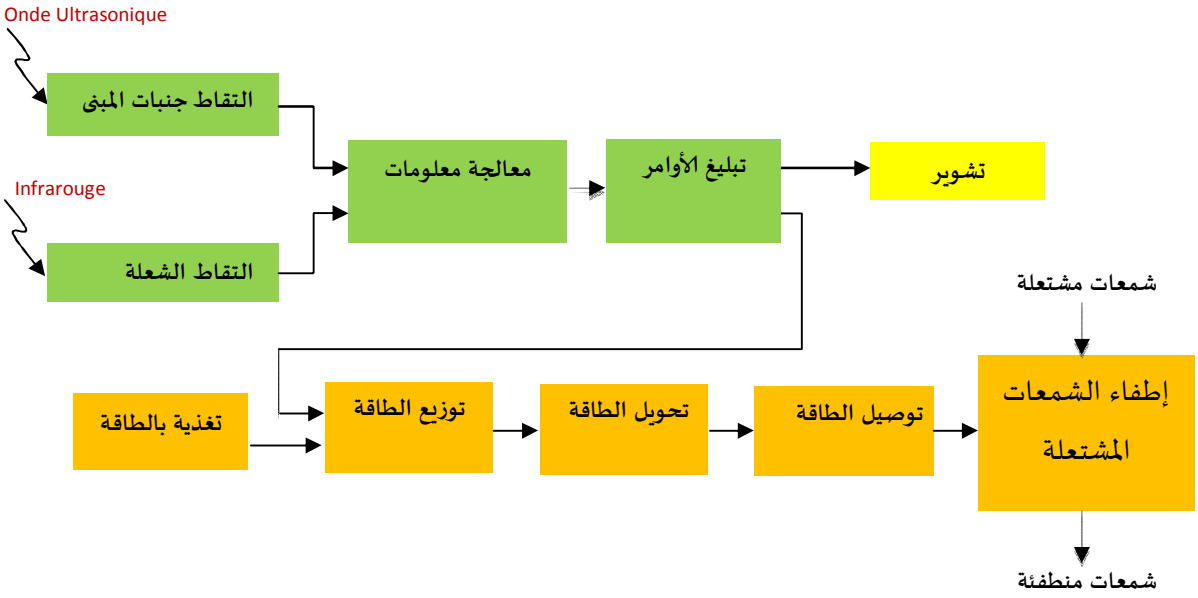
يبعث الروبوت إشارات ضوئية عند :

- التقاطه لهب شمعة مشتعلة ؛
- توقفه لإطفاء الشمعة .

2. الرشم البنيوي لتركيبة التحكم في الروبوت :

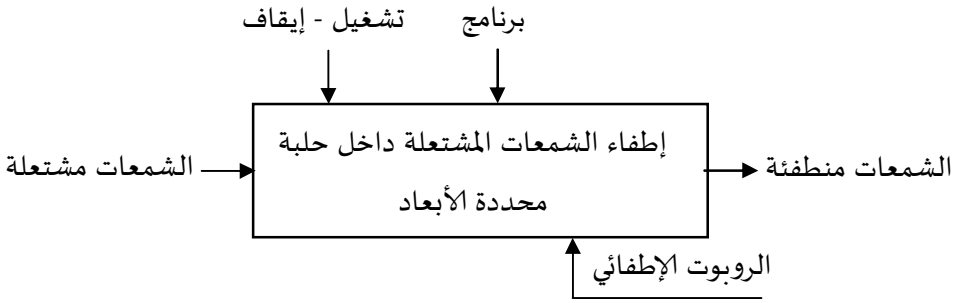


3.التمثيل الوظيفي باعتماد نموذج سلسلة المعلومات وسلسلة الطاقة :

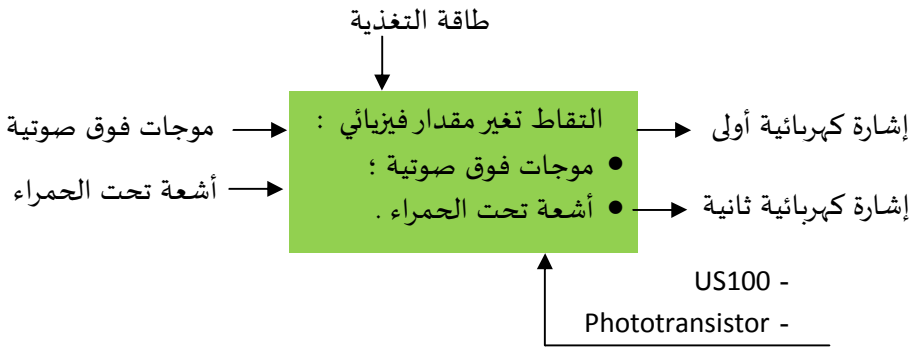


4.دراسة الوظائف التقنية للروبوت الإطفائي:

❖ التمثيل الوظيفي :



❖ وظيفة الالتقاط :



1. التقاط جوانب الحلبة :

تم اعتماد ملتقط الموجات فوق الصوتية **US100** لعدم تأثره خصوصا بمؤثرات المحيط الضوئية مقارنة بملتقطات الأشعة تحت الحمراء . يتميز الملتقط **US100** :

- بمسافة الالتقاط : من 2cm إلى 450cm ؛
- بزاوية الالتقاط : 15 درجة .

بالنسبة لتحريك الروبوت ، اعتمدنا محركا كهربائيا مرتبطا بمخفض للسرعة لكل عجلة ويتم تزويدهما بالطاقة الكهربائية من خلال الموزع L293D.

خصائص المحرك والعجلة	خصائص الموزع L293D	تعليل الاختيار
توتر الاشتغال : من3V إلى 6V التيار : 120mA سرعة الدوران : 240tr/min قطر العجلة : 54mm السرعة الخطية V=n*2*π*r/60=0,68m/s	التحكم في أربعة محركات في منحى واحد أو محركين في منحيين. تيار قصوي 600mA توتر اشتغال الدارة المدمجة : 7Vmax توتر اشتغال المحرك : 36Vmax	إمكانية التحكم في المحرك باعتبار التوتر (6V<36V) والتيار الكهربائي (120mA<600mA) زحمة صغيرة مقارنة باستعمال ترانزيستورات(8) سهولة الاستعمال والتركيب. إمكانية التحكم في سرعة المحركات بواسطة إشارة PMW

2. التقاط الشمعة المشتعلة :

الملتقط المعتمد هو KY-026 ، وهو ملتقط للأشعة تحت الحمراء التي تبعثها شعلة النار . ومن أهم خصائصه :

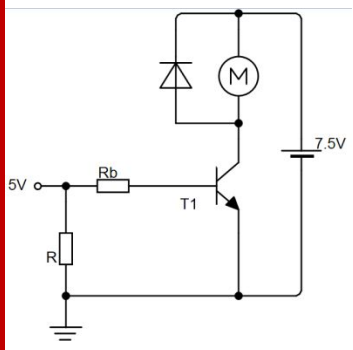
خصائص ملتقط الشعلة	تعليل الاختيار
- المجال : من 760nm إلى 1100nm . - توتر الاشتغال : من 3.3V إلى 5.5V . - إمكانية تغيير حساسيته بواسطة مقاومته المتغيرة - زاوية الاتقاط : 60°.	- ملائم لالتقاط شعاع الشمعة . - إمكانية التحكم في حساسية الالتقاط بواسطة مقاومته المتغيرة وباستعمال البرنامج - يتيح اختيار استغلال إشارة رقميةD0 أو تناظرية A0 - وحدة مدمجة ذات زحمة صغيرة. - سهل الاستغلال بواسطة بطاقة الأردوينو.

لقد استعملنا المربط التناظري (A0) للملتقط وقمنا بتجريب حساسيته لمعرفة القراءات التي يعطيها وبالتالي تحديد قيمة العتبة الملائمة. بالنسبة لبرنامجنا العتبة هي: (threshold=20).

❖ وظيفتي التوزيع والتحويل

الخاصيات الكهربائية للمحرك (توتر التغذية خلال التجربة 7.4V):

1. بدون مروحة: (I = 0.5Aà vide) ؛
2. عند تركيب المروحة: I=1.75A (en charge)



علما أن التيار الذي يعطيه مخرج الأردوينو تحت 5V هو 20mA وهو تيار أقل بكثير مما يطلبه المحرك (1.75A) . لذلك استعملنا الترانزيستورT1 بين المحرك وبطاقة الأردوينو لتوزيع التيار المطلوب مباشرة من منبع التغذية باستغلال وظيفة التبديل خصائص الترانزيستور **KTD1413**

الخاصيات الكهربائية **KTD1413** Transistor Darlington

$\beta min = 500$ ، VBEsat=1.6V ، VCEsat=0.9V ، Ic max=5A

لحساب قيمة المقاومة Rb ، نعتمد العلاقة جانبه : $Rb \leq \beta min \cdot r$

علما أن r تمثل المقاومة الداخلية للمحرك التي نقوم بقياسها مباشرة بواسطة جهاز القياس الأوم ميتر .

لقد استعملنا مروحة ذات أربع شفرات قطرها Ø=60mm تم تفكيكها من جهاز للتهوية يستعمل في مطابخ المنازل بصفة عامة . ولتتلاءم المروحة مع المحرك المؤازر لتوجيهها ، عملنا على بعض التعديلات عليها وإحداث ترابط تامة قابل للفك بينهما .

الخصائص الكهربائية للمحرك المؤازر **MG90S**:

- توتر الاشتغال بين :4.8V و 6V ؛
- سرعة الحركة بين :0.10s/60° و 0.11s/60° ؛
- زاوية الدوران: 180° ؛
- العزم : بين 2.2Kg.cm و 2.5Kg.cm .

ملاحظات :

1. عند استغلال مكتبة المحرك المؤازر ، يتم الإلغاء التلقائي لإمكانية تشغيل الخاصية PWM بالنسبة للمربطين (9 و10) فقط بالبطاقة Arduino UNO التي تم استعمالها .
2. المقاومة المركبة بين المربط A2 والأرضي تضمن حصول المستوى الدنيوي في المربط (Pull-down).

5.لائحة المركبات الكهروبية:

الترتيب	التعيين	العدد	ملاحظات
15	بطاقة الأردوينو UNO	1	REV3
14	ملتقط الموجات فوق الصوتية U_g,U_d,U_av	3	US100
13	ملتقط النار Cap_f_d,Cap_f_g	2	KY-026
12	ثنبيل متألق كهربائيا D2,D3	2	أحمر بقطر 5mm
11	ثنبيل ذو وصل D1	1	1N4148
10	مقاومة R3,R4	2	180Ω-1/4W
9	مقاومة R2	1	470Ω-1/4W
8	مقاومة R1	1	10kΩ-1/4W
7	ترنزيستور T1NPN	1	KTD1413
6	دارة مدمجة IC1	1	L293D
5	محرك مؤازر servo	1	MG90S
4	محرك كهربائي بالتيار المستمر Mot_vent	1	0.5A / 7.4V
3	محرك كهربائي بالتيار المستمر مع مخفض السرعة Mot_g,Mot_d	2	من-3V إلى 6V / 120mA
2	قاطع تيار M/A	1	ذو عتلة(à levier)
1	بطارية li-ion 18650	2	3.7V-2200mAh

ملحق : Code

```
/* progetconcournationnal(ROBOT POMPIER)-edition 2018.
 - programme pour controler le comportement du robot pompier:
 - realisation par les élèves :IMAD EL AYYACHI.
                              HAMZA EL MAKHTOUM.
 - programmé par le prof de technologie : BOUMEDIENE JABRI.
 - collegeabd el khalektoress -oujda-  .
 */
#include<Servo.h>
//ULTRASONS----------------
#defineECHOavant 5  // ultason avant.
#defineTRIGavant 4
intDISTavant, Da;
#defineECHOdroite 3 // ultrson droit.
#defineTRIGdroite 2
intDISTdroite, Dd;
#defineECHOgauche 9  // ultrason gauche.
#defineTRIGgauche 8
intDISTdgauche, Dg;
//MOTEURS-------------------
int pause =100;   //attente
int obstacle =20; //distance limite(en cm) devant un obstacle .
int Md =7 ;intENd =6 ;intspeedd =120; intMdr=10;
/* sens avant Moteur droite (Md=pin7) ;
vitesse Moteur droite (speedd=pin6)PWM;
sens arrière moteur droit (Mdr=pin10)*/
int Mg =12 ;intENg =11 ;intspeedg =100; int Mgr=13;
/* sens avant Moteur gauche (Mg=pin12) ;
vitesse Moteur gauche (speedg=pin11)PWM ;
sens arrière moteur gauche (Mgr=pin13);
(attention utilisation du servodesactive la fonction (PWM) de pin 9et10)*/
//capteurs de flamme----------
intcapteurfd=A0;
intvaleurfd=0;
intcapteurfg=A3;
intvaleurfg=0;
intthreshold=20;// seuil de detection (pour les 2 capteurs de flamme)=
valaeur prise selon les tests.
//signalisation-------------------
int led1=A4;  // led rouge (droit) qui s'allume à l'arret du robot.
int led2=A5;  // led rouge (gauche) qui s'allume à l'arret du robot.
//ventilation-----------------------
int ventilateur=A2; //commande du moteur ventilateur.(A2=pin analogique
prise comme numerique).
// SERVOMOTEUR-----------------------
Servomyservo;
//==============================="void
setup"================================
voidsetup() {
  //ULTRASONS-------------------
pinMode(ECHOavant,INPUT);
```

```
pinMode(TRIGavant,OUTPUT);
pinMode(ECHOdroite,INPUT);
pinMode(TRIGdroite,OUTPUT);
pinMode(ECHOgauche,INPUT);
pinMode(TRIGgauche,OUTPUT);
 //MOTEURS----------------------
pinMode(Md,OUTPUT); pinMode(Mdr,OUTPUT); pinMode(ENd,OUTPUT);
analogWrite(ENd,speedd);
pinMode(Mg,OUTPUT); pinMode(Mgr,OUTPUT); pinMode(ENg,OUTPUT);
analogWrite(ENg,speedg);
 //capteurs de flamme------------
pinMode(capteurfd,INPUT);
pinMode(capteurfg,INPUT);
 //ventilation-------------------
pinMode(ventilateur,OUTPUT);
 //signalisation--------------------
pinMode(led1,OUTPUT);
pinMode(led2,OUTPUT);
 // SERVOMOTEUR----------------------
myservo.attach(A1);
 /* l'utilisation de la bibliothèque"servo" désactive la fonctionnalité
 (PWM) de la commande analogWrite () sur les broches 9 et 10, qu'il y ait
ou non un Servo sur ces broches.*/
}
//LES FONCTIONS----------------------
voidstop1(){
analogWrite(ENd,speedd);
analogWrite(ENg,speedg);
digitalWrite(Md,0);
digitalWrite(Mg,0);
digitalWrite(Mdr,0);
digitalWrite(Mgr,0);
digitalWrite(led1,HIGH);
digitalWrite(led2,HIGH);
      }
voidavance(){
analogWrite(ENd,speedd);
analogWrite(ENg,speedg);
digitalWrite(Md,1);
digitalWrite(Mg,1);
digitalWrite(Mdr,0);
digitalWrite(Mgr,0);
digitalWrite(led1,LOW);
digitalWrite(led2,LOW);
   }
voiddroite(){  //servir pour equilibrer le dedeplacement au milieu de la piste
analogWrite(ENd,speedd);
analogWrite(ENg,speedg);
digitalWrite(Md,0);
digitalWrite(Mg,1);
digitalWrite(Mdr,0);
digitalWrite(Mgr,0);
```

```
    }
voidgauche(){  //servir pour equilibrer le dedeplacement au milieu de la piste
analogWrite(ENd,speedd);
analogWrite(ENg,speedg);
digitalWrite(Md,1);
digitalWrite(Mg,0);
digitalWrite(Mdr,0);
digitalWrite(Mgr,0);
  }
voidtourne_a_droite(){  //servir pour changer le sens du deplacement vers la
droite.
analogWrite(ENd,speedd);
analogWrite(ENg,speedg);
digitalWrite(Md,0);
digitalWrite(Mdr,1);// moteur droit (recule)
digitalWrite(Mg,1); //moteur gauche (avance)
digitalWrite(Mgr,0);
   }
voidtourne_a_gauche(){  //servir pour changer le sens du deplacement vers
la gauche.
analogWrite(ENd,speedd);
analogWrite(ENg,speedg);
digitalWrite(Md,1); //moteur droit (avance).
digitalWrite(Mdr,0);
digitalWrite(Mg,0);
digitalWrite(Mgr,1);//moteur gauche (recule).
   }
voidventilation(){
digitalWrite(ventilateur,HIGH);
delay(1000);
digitalWrite(ventilateur,LOW);
   }
//============================"voidloop"=======================
==================
voidloop() {
myservo.write(90);
avance();
digitalWrite(TRIGavant,LOW);
delayMicroseconds(2);
digitalWrite(TRIGavant,HIGH);
delayMicroseconds(10);
digitalWrite(TRIGavant,LOW);
floatDISTavant = pulseIn(ECHOavant,HIGH); //calcul de distance vers
l'obstacle de devant
      Da=DISTavant/58;
digitalWrite(TRIGdroite,LOW);
delayMicroseconds(2);
digitalWrite(TRIGdroite,HIGH);
delayMicroseconds(10);
digitalWrite(TRIGdroite,LOW);
floatDISTdroite = pulseIn(ECHOdroite,HIGH); //calcul de distance vers  l'obstacle du coté
droit
```

```
        Dd=DISTdroite/58;
digitalWrite(TRIGgauche,LOW);
delayMicroseconds(2);
digitalWrite(TRIGgauche,HIGH);
delayMicroseconds(10);
digitalWrite(TRIGgauche,LOW);
floatDISTgauche = pulseIn(ECHOgauche,HIGH); //calcul de distance vers
l'obstacle du coté gauche
        Dg=DISTgauche/58;
//DEPLACEMENT DU ROBOT:
  if ((Da<=obstacle)&& (Dd<Dg))
{stop1();tourne_a_gauche();delay(100);avance();}
else if ((Da<=obstacle)&& (Dg<Dd))
{stop1();tourne_a_droite();delay(100);avance();}
else if ((Da>obstacle) && (Dd<Dg)) {gauche();delay(25);}
else if ((Da>obstacle) && (Dg<Dd))  {droite();delay(25);}
  /*25ms : valeur experimentale pour eviter le contact
contre les cotés de la sortie du circuit*/
//DETECTION DE LA FLAMME:
valeurfd=analogRead(capteurfd);
valeurfg=analogRead(capteurfg);
if(valeurfd<=threshold)
 {stop1();myservo.write(15);delay(valeurfd);}
while(valeurfd<=threshold)
{stop1();delay(pause);ventilation();stop1();delay(3000);break;}
/* le stop avec un delay de 3000 pour permettre au robotdereagir
une 2eme fois si la bougie se réallumeapres la premiere ventilation..*/
if (valeurfg<=threshold)
 {stop1();myservo.write(160);delay(valeurfg);}
while(valeurfg<=threshold)
 {stop1();delay(pause);ventilation();stop1();delay(3000);break;}

 }
```