



**Daisylab.fr**

**POURQUOI ...**



Daisylab.fr

**POURQUOI**  
**La porte du garage**  
**s'ouvre-t-elle ?**



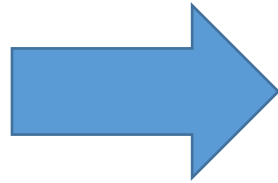
L'ouverture de la porte du garage est  
une illustration d'un principe  
de base de l'automatisme :



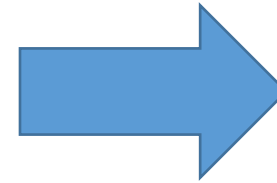


Daisylab.fr

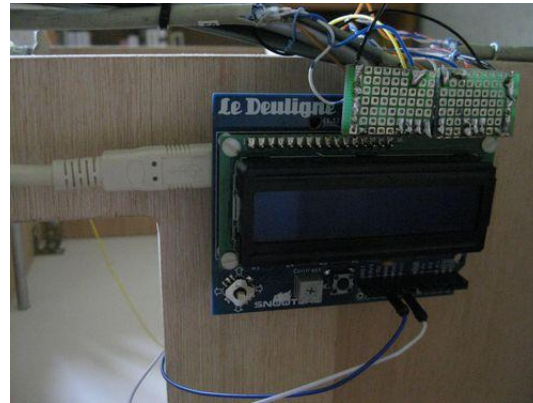
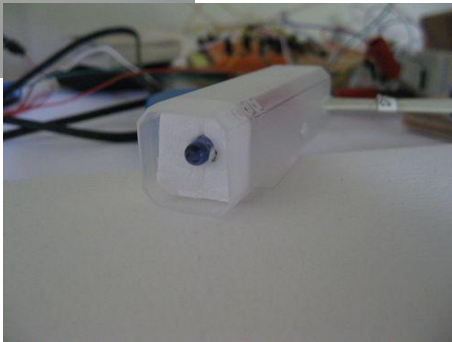
CAPTEUR :  
BARRIERE INFRA  
ROUGE



TRAITEMENT  
INFORMATION:  
ARDUINO



ACTION :  
MOTEUR PORTE  
GARAGE  
( Servo de  
modélisme)

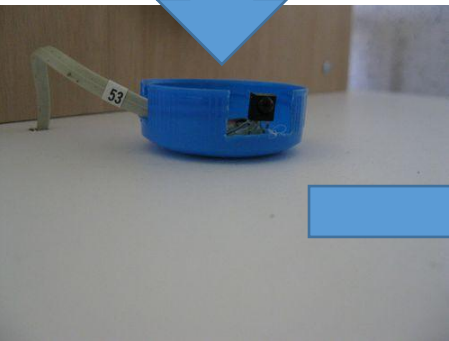
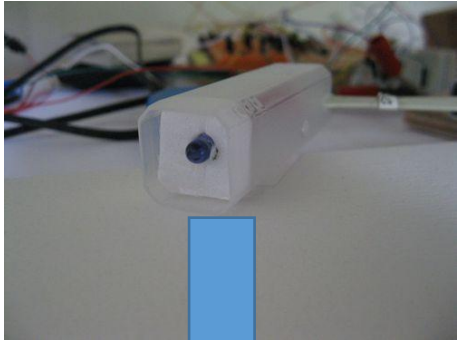




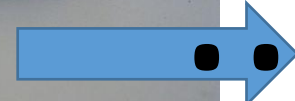
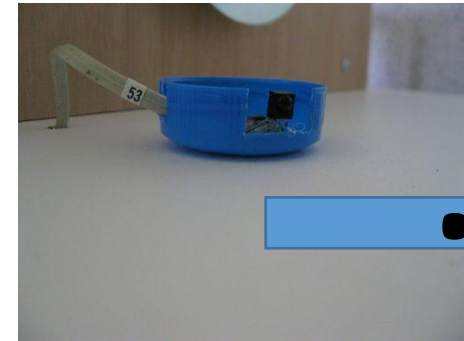
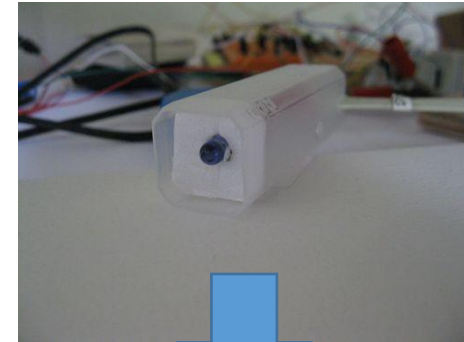
CAPTEUR :  
BARRIERE INFRA ROUGE

Une diode émettrice  
de lumière infra rouge

Un récepteur, qui donne une  
information = 0  
lorsqu'il reçoit la lumière  
et = 1 lorsqu'il ne la reçoit plus



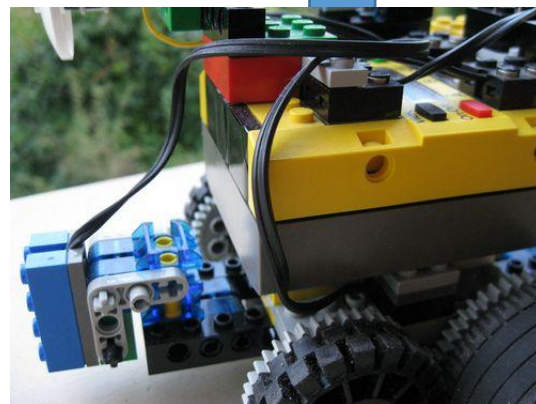
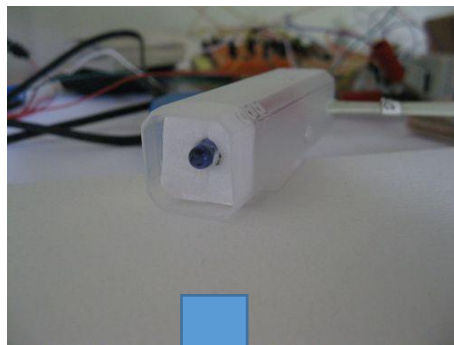
0..



1..

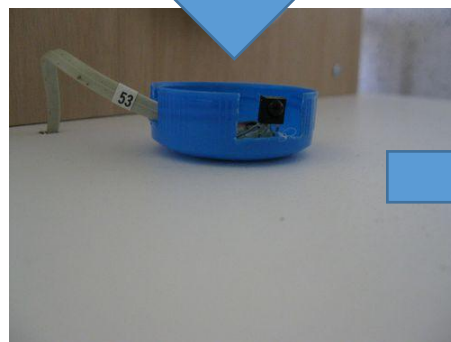


Daisylab.fr



Lorsqu'il passe devant, le robot coupe le faisceau, et le récepteur ne reçoit plus de lumière.

Le récepteur passe à 1.



..1..

CAPTEUR :  
BARRIERE INFRA ROUGE



Daisylab.fr

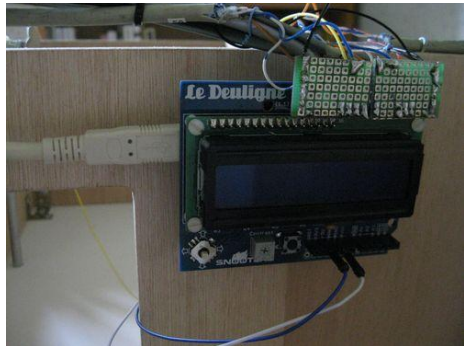
## TRAITEMENT INFORMATION: ARDUINO

Le récepteur infra rouge est connecté à un microcontrôleur open source l'Arduino : cela correspond à une **ENTREE** pour l'Arduino.

**Les 0 et les 1 sont matérialisés par des tensions, 0V pour 0 et +5V pour 1**

Si une ENTREE reçoit du +5V on dit qu'elle est à 1,  
Si elle reçoit du 0 V on dit qu'elle est à 0.

Un programme installé sur l'Arduino va scruter l'entrée correspondant au Récepteur infra rouge et déclencher une action en fonction de son état 0 ou 1



Voici un extrait du programme qui ouvre la porte !

```
Void loop() {
```

```
....
```

```
if (digitalRead(PIN_INFRAROUGE)==1)
```

```
{
```

```
    PositionGarage=RotationGarage(target_garageouvert);
```

```
} else
```

```
{
```

```
.... Suite du code....
```

```
}
```

```
.....
```

```
}
```

**.....Et en français qu'est ce que cela veut dire ??**





Daisylab.fr

**Void loop() {**

....

```
if (digitalRead(PIN_INFRAROUGE)==1)
{
  PositionGarage=RotationGarage(target_garageouvert);
} else
{
  .... Suite du code....
}
```

.....

```
}
```

Le programme commence par une instruction Void loop qui indique que l'Arduino doit exécuter ce qui suit en boucle, sans jamais s'arrêter !



```
Void loop() {  
....  
  
if (digitalRead(PIN_INFRAROUGE)==1)  
{  
    PositionGarage=RotationGarage(target_garageouvert);  
} else  
{  
    .... Suite du code....  
}  
  
.....  
}
```

La séquence **IF ... ELSE** veut dire :

Si (if) la condition suivant est vraie exécute ce qui suit entre les { }

Sinon (else) exécute ce qui suit maintenant entre les {}

Remarquez les { } qui encadrent chaque partie de code : cela fait partie de la syntaxe du langage de programmation de l'Arduino.



Là on demande à l' Arduino de lire la broche sur laquelle est raccordée le récepteur infra rouge :

⇒digitalRead(PIN\_INFRAROUGE)

Puis on compare le résultat à la valeur 1 :

```
Void loop() {  
.....  
  
if (digitalRead(PIN_INFRAROUGE)==1)  
  {  
    PositionGarage=RotationGarage(target_garageouvert);  
  } else  
  {  
    .... Suite du code....  
  }  
  
.....  
}
```

**Si la valeur lue est 1, la condition digitalRead(PIN\_INFRAROUGE)==1 est VRAIE**

**Et l'Arduino va exécuter l'instruction qui suit :**

PositionGarage=RotationGarage(target\_garageouvert);

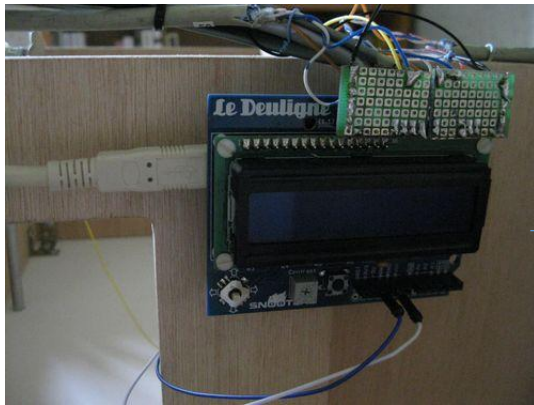
ACTION :  
MOTEUR PORTE GARAGE  
( Servo de modélisme)

```
Void loop() {  
....  
if (digitalRead(PIN_INFRAROUGE)==1)  
{
```

Pour ouvrir ou fermer la porte , l'Arduino va commander un servomoteur, tel que l'on en trouve dans les avions ou voitures télécommandés.

```
PositionGarage=RotationGarage(target_garageouvert);
```

```
} else  
{  
.... Suite du code....  
}  
  
.....  
}
```



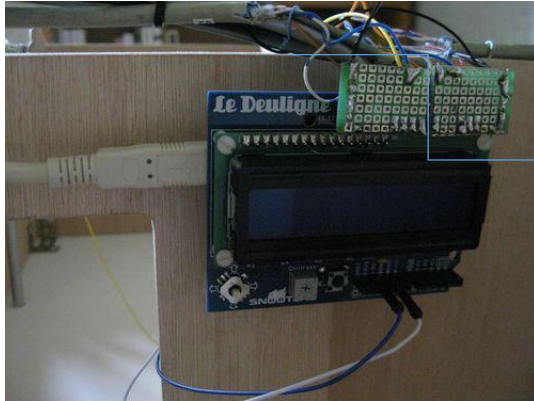
Le servomoteur est raccordé à une borne de **SORTIE** de l'Arduino ainsi qu'à une alimentation extérieure.



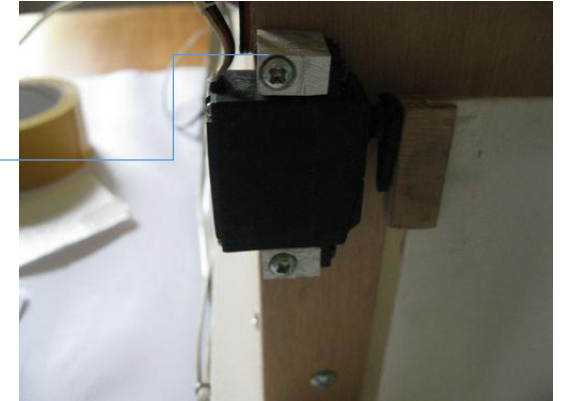
**Le servomoteur tournera lorsque la sortie sera mise à +5V par l'Arduino**



ACTION :  
MOTEUR PORTE GARAGE  
( Servo de modélisme )



C'est en mettant la SORTIE sur laquelle est connecté le servomoteur successivement à 0V et +5V (impulsions) que le programme va pouvoir positionner le servomoteur précisément.

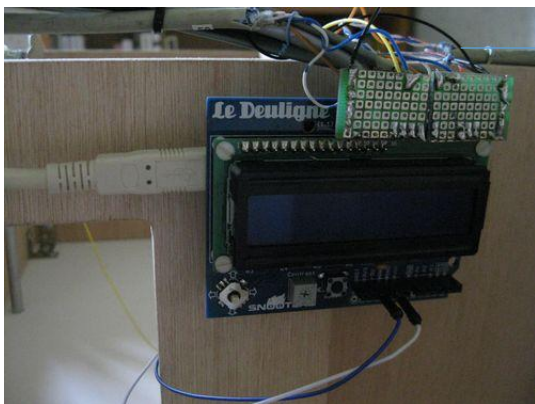


Pour simplifier, en partant de 0, envoyer 90 impulsions va mettre le servomoteur à 90° , en envoyer 180° le mettra à 180°



Daisylab.fr

**Il faut encore beaucoup d'instructions pour que la porte s'ouvre, en voici un extrait !**



**ACTION :  
MOTEUR PORTE GARAGE  
( Servo de modélisme)**

```
int RotationGarage(int cible)
{
  int Pact;
  int angle;
  if (PositionGarage < cible )
  {
    for (angle = PositionGarage; angle <= cible; angle +=1)
    {
      servoGarage.write(angle);
      delay(freine);
    }
  } else
  if (PositionGarage > cible)
  {
    for (angle PositionGarage; angle >= cible; angle -=1)
    {
      servoGarage.write(angle);
      delay(freine);
    }
  }
  Pact=cible;
  return Pact;
}
```





```
int RotationGarage(int cible)
{
int Pact;
int angle;
if (PositionGarage < cible )
{
for (angle = PositionGarage; angle <= cible; angle +=1)
{
servoGarage.write(angle);
delay(freine);
}
} else

if (PositionGarage > cible)
{
for (angle PositionGarage; angle >= cible; angle -=1)
{
servoGarage.write(angle);
delay(freine);
}
}
Pact=cible;
return Pact;
}
```

ACTION :  
MOTEUR PORTE GARAGE  
( Servo de modélisme)

**La cible correspond à ce que l'on veut obtenir.**

**Dans notre cas, 90 pour porte ouverte  
et 0 pour porte fermée.**

**Si la position de la porte est inférieure à la cible,  
on fait tourner le servo dans le sens d'ouverture.**

**Sinon, on le fait tourner dans le sens inverse.**

**Pour ouvrir, on va donc dire au servo:  
va à la position 1 puis 2 puis 3... jusqu'à 90!**

**Pour ne pas ouvrir la porte trop vite on attend un  
peu entre chaque ordre, c'est l'instruction delay()**



**Daisylab.fr**