



Rapport du projet :

Tutoriel : Créer un système de sécurité pour Les vidéoprojecteurs

Encadré par : Mr. BOUDHIR Anouar

Réalisé par : - HADDI Nawfal

Plan :

- 1- Introduction**
- 2- Prérequis**
- 3- Conception électronique**
- 4- Application web**
- 5- Programme du système**
- 6- Conclusion**

1-Introduction :

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui.

Ce que nous allons apprendre aujourd'hui est un mélange d'électronique et de programmation. On va en effet parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de l'électronique.

En outre nous allons précisément vous montrer comment construire un système de sécurité pour les vidéoprojecteurs en utilisant des matériels électroniques et code de programmation.

2-Prérequis :

Dans ce travail vous aurez besoin d'une part de connaissance basique en programmation et électronique vous pouvez lire [ce cours](#) afin d'avoir ces connaissances, et en PHP et MySQL, d'autre part vous aurez besoin des matériels suivant :

- Arduino Uno R3 (15-20\$)
- Capteur de vibration SW-420 (2-3 \$)
- Capteur à ultrasons HC-SR04 (3\$)
- Module WIFI ESP8266 (5\$)
- LED (0.1\$)
- Sirène (Buzzer 0.6\$)
- fils de connexion
- Une breadboard

3-Conception électronique :

1) Capteur de vibration SW-420 :

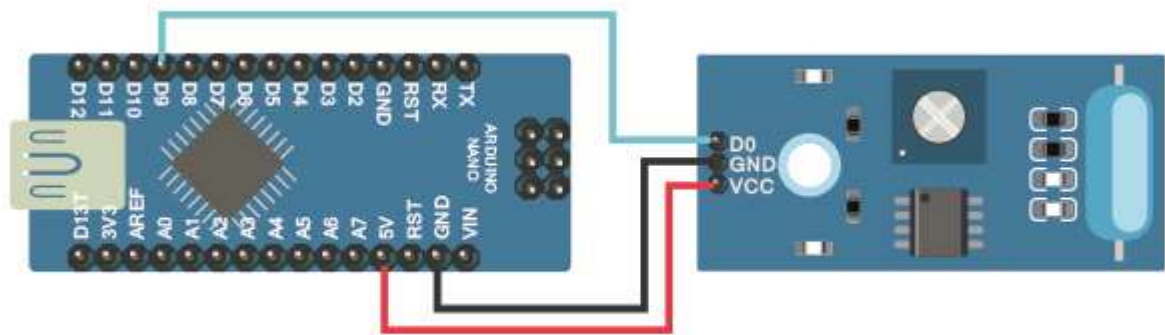
Ce capteur servira à récupérer une valeur lorsqu'il capte une vibration. Le fonctionnement de ce capteur est très simple, il reçoit l'énergie à partir de l'arduino est quand il est bougé il laisse passer le courant par la sortie D0 pendant la période de mouvement. en général le capteur a trois broches : DO, GND et VCC .

Son montage est comme le suivant :

Pin de l'Arduino -> D0 du module.

GND de l'Arduino -> GND du module.

VCC de l'Arduino -> VCC du module



Voilà un code exemple pour le tester :

```
int ledPin =13;
int EP =9;
void setup(){
  pinMode(ledPin, OUTPUT);
  pinMode(EP, INPUT); //set EP input for
  measurment
  Serial.begin(9600); //init serial 9600
}
```

1

```
void loop(){
  long measurement =TP_init();
  delay(50);
  Serial.print("measurment = ");
  Serial.println(measurement);
  if (measurement > 1000){ digitalWrite(ledPin,
  HIGH);}
  else{ digitalWrite(ledPin, LOW); }}
```

2

```
long TP_init(){
  delay(10);
  long measurement=pulseIn (EP, HIGH); //wait for the pin to get HIGH and returns measurement
  return measurement; }
```

3

2) Capteur à ultrasons HC-SR04:

Le capteur à ultrason simple permet, comme un micro, de capter des sons. Sauf qu'il ne capte que les ultrasons, c'est-à-dire les sons situés au-delà des 20KHz (précisément de 20 KHz à 10 MHz).

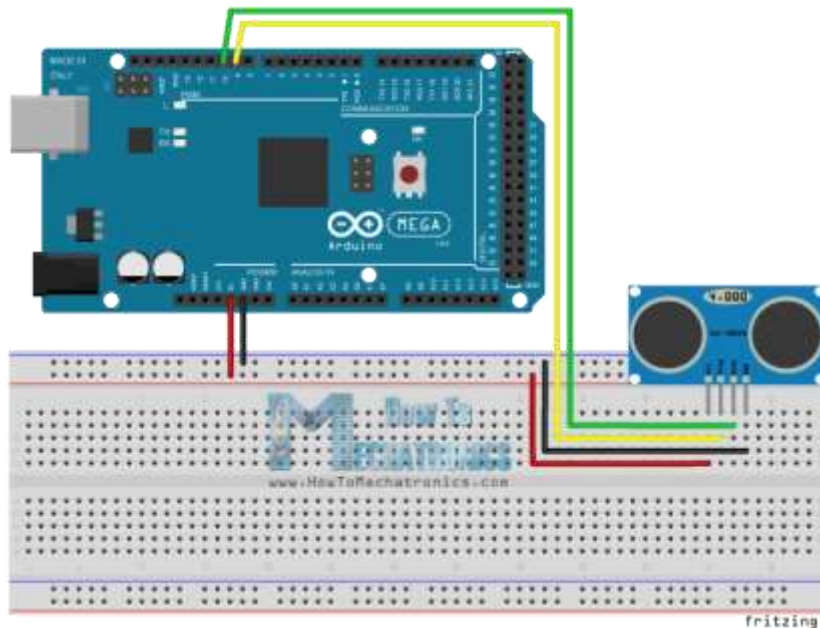
Le HC-SR04 contient à la fois un capteur et un émetteur à ultrasons.

Voici une petite photo de présentation :

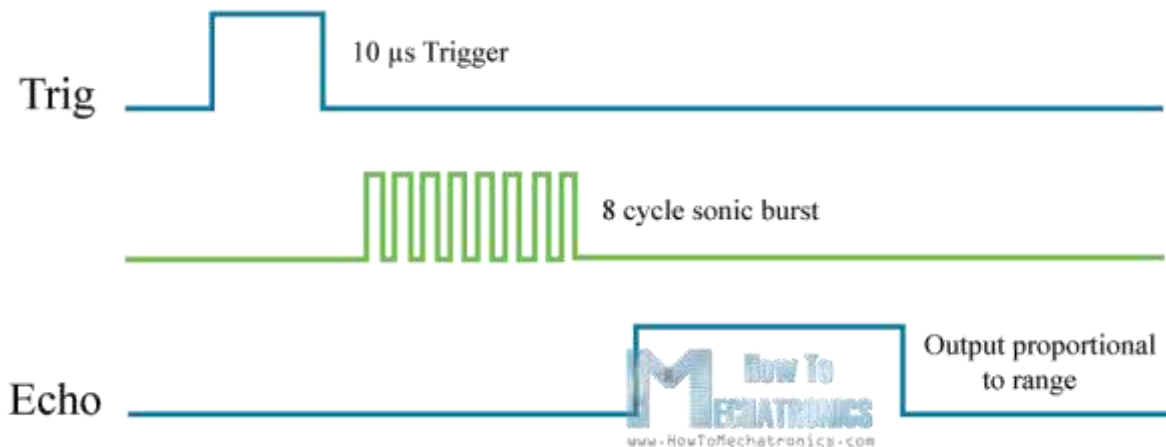


Le module ultrasonique HC-SR04 dispose de 4 broches, Terre, VCC, Trig et Echo. Les broches de masse et VCC du module doivent être connectées à la masse et au broche de 5 volts de la carte Arduino, ainsi les broches trig et echo doivent être branché aux broches d'Entrée / Sortie numérique de la carte Arduino.

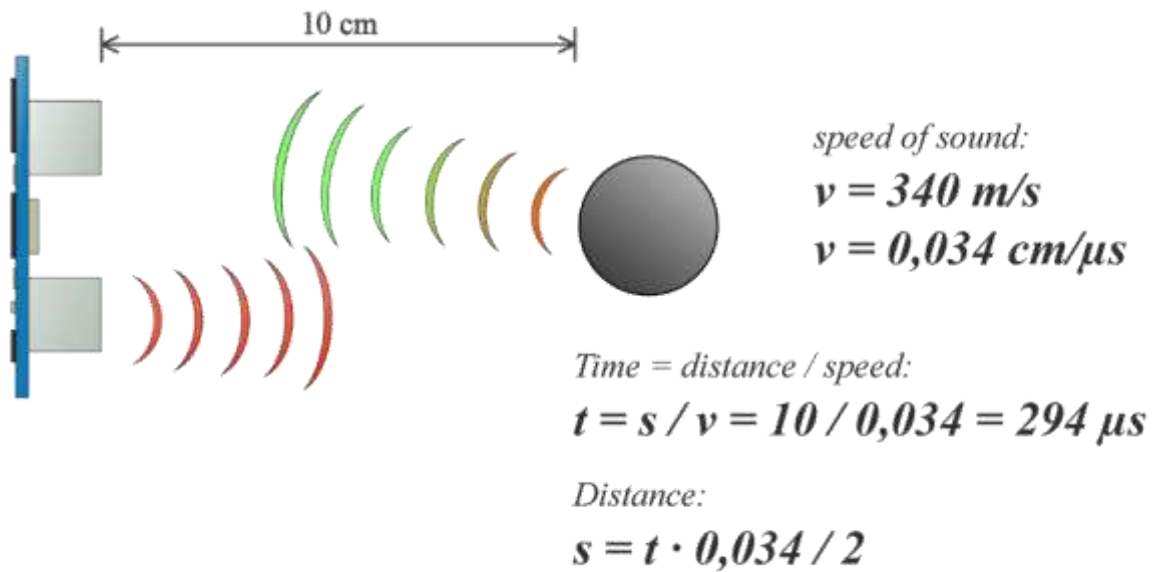
Voilà son montage :



Afin de générer une échographie, vous devez définir la Trig sur un état haut pour 10 μ s. Cela enverra une rafale sonore de 8 cycles qui voyagera à la vitesse du son et sera reçue dans la broche Echo. La broche Echo affichera le temps en microsecondes parcouru par l'onde sonore.



Comment calculer la distance ?



Code pour le tester :

```
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
// defines variables
long duration;
int distance;
```

```
void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as
                             // an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as
                             // an Input
  Serial.begin(9600); // Starts the serial
                       // communication }
}
```

```
void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW); delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH); delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance= duration*0.034/2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance: "); Serial.println(distance);
}
```

3) Module WIFI ESP8266:

L'ESP8266 est un circuit intégré avec un microcontrôleur permettant la connexion en WiFi. Les modules intégrant ce circuit sont très utilisés pour contrôler des périphériques par Internet. L'ESP8266 est livré avec un firmware préinstallé vous permettant d'en prendre le contrôle à l'aide de « commandes AT » standards pouvant provenir d'une carte Arduino avec qui il peut communiquer par liaison série. Mais toute la souplesse et la puissance de ce module résident dans le fait que vous pouvez aussi y développer et flasher votre propre code, rendant ainsi le module entièrement autonome.

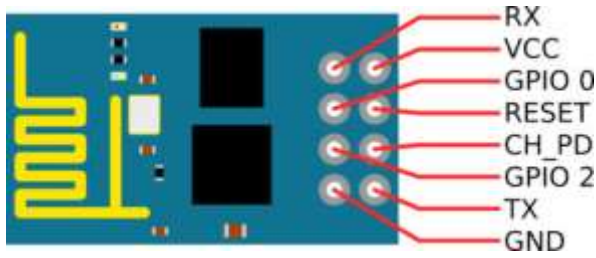
Pour l'Internet des Objets (IdO), ce petit module (25x14 mm) vous rendra de grands services pour quelques dirhams à déboursier seulement.

L'ESP8266 en bref :

- 1- Supporte les protocoles 802.11 b/g/n ;
- 2- Se connecte à votre routeur et fonctionne comme client ou comme point d'accès ou même les deux en même temps !
- 3- Peut fonctionner comme serveur web avec sa propre adresse IP ;
- 4- Le module ESP-01 comprend deux broches numériques qui peuvent être configurées en entrée ou en sortie (pour piloter des LED, des relais, etc.). Ces broches peuvent être utilisées pour générer un signal modulé en largeur d'impulsion (PWM). D'autres versions du module comportent davantage de broches d'entrées-sorties (l'ESP-12 par exemple), mais ils se programment tous de façon similaire ;
- 5- Possède une entrée analogique (ADC/TOUT), mais qui n'est pas câblée sur l'ESP-01 ;
- 6- Peut fonctionner en communiquant avec une carte Arduino ou être programmé pour fonctionner en toute autonomie ;
- 7- Dispose de nombreux outils et environnements de développement (EDI) pour le programmer ;
- 8- Et plein d'autres choses encore...

D'abord on commence par Flasher notre module à l'aide de l'arduino afin qu'on soit tous dans la même route :

Pour le montage de Flash veuillez suivre le montage suivant :

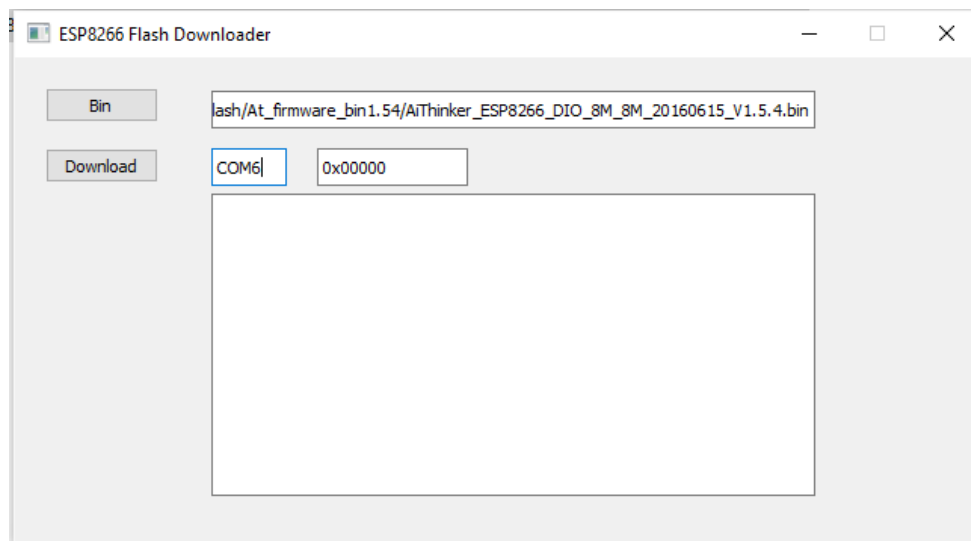


ESP RX -> ARDUINO RX
 ESP TX -> ARDUINO TX
 ESP VCC -> ARDUINO 3.3V
 ESP CH_PD -> ARDUINO 3.3V
 ESP GPIO 0 -> ARDUINO GND
 ESP GND -> ARDUINO GND

Vous trouverez dans [ce lien](#) le programme utilisé pour le flash et le firmware.

Pour le flasher, vous accédez à l'application esp8266_flasher.exe et vous mettez votre firmware dans le champ de bin et vous spécifiez le port COM utilisé par l'arduino comme il est montré dans l'image ci-dessous.

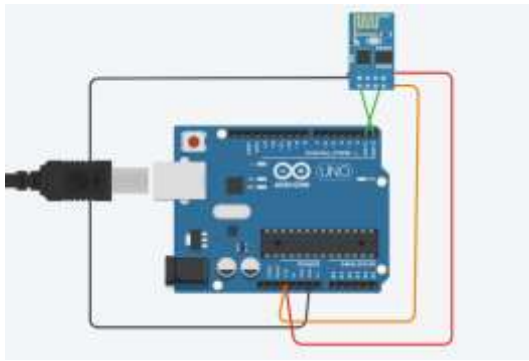
NB : (pour savoir quel port utilisé par l'arduino vous accédez dans votre EDI arduino à outils -> port -> COM..).



Quand le flash est terminé, nous serons prêts pour se connecter à notre module esp8266.

Maintenant on passe à la communication avec notre module à travers l'arduino.

D'abord on doit faire le montage suivant :

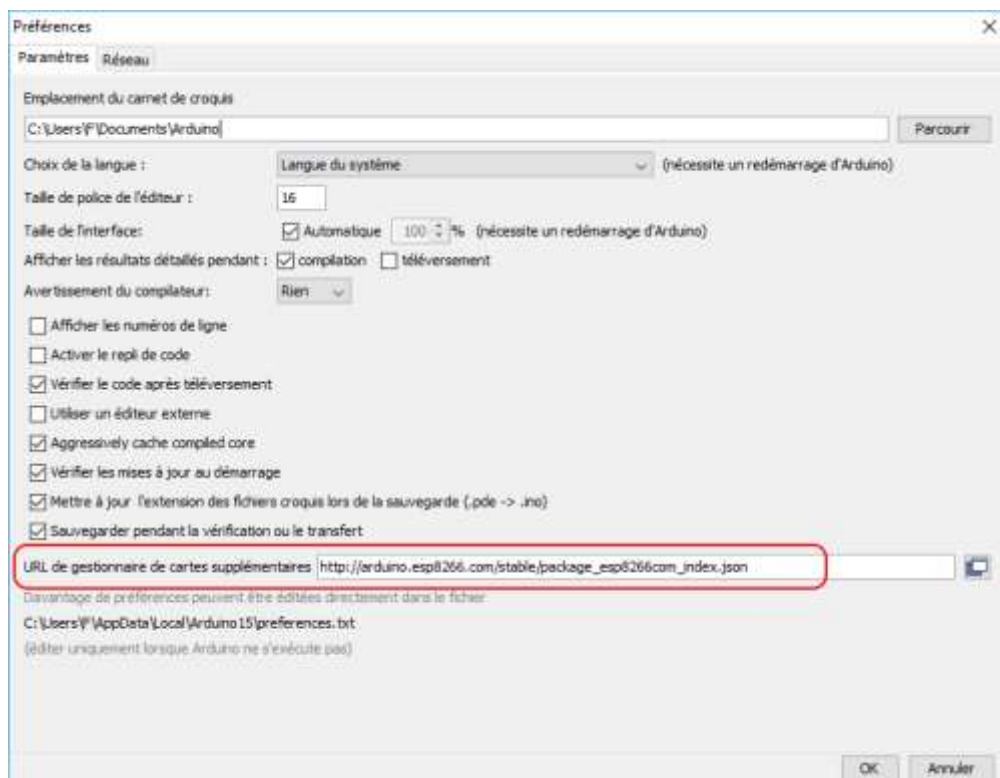


ESP RX -> ARDUINO TX
 ESP TX -> ARDUINO RX
 ESP VCC -> ARDUINO 3.3V
 ESP CH_PD -> ARDUINO 3.3V
 ESP GND -> ARDUINO GND

De nombreux outils existent pour programmer l'ESP8266, mais l'EDI Arduino (Environnement de Développement Intégré) est sans doute le moyen le plus facile d'y arriver.

Vous aurez besoin de [la dernière version de l'EDI Arduino](#) et d'y ajouter le support de l'ESP8266.

Dans le menu Préférences de l'EDI Arduino, il faut rentrer une URL pour gérer les cartes supplémentaires :

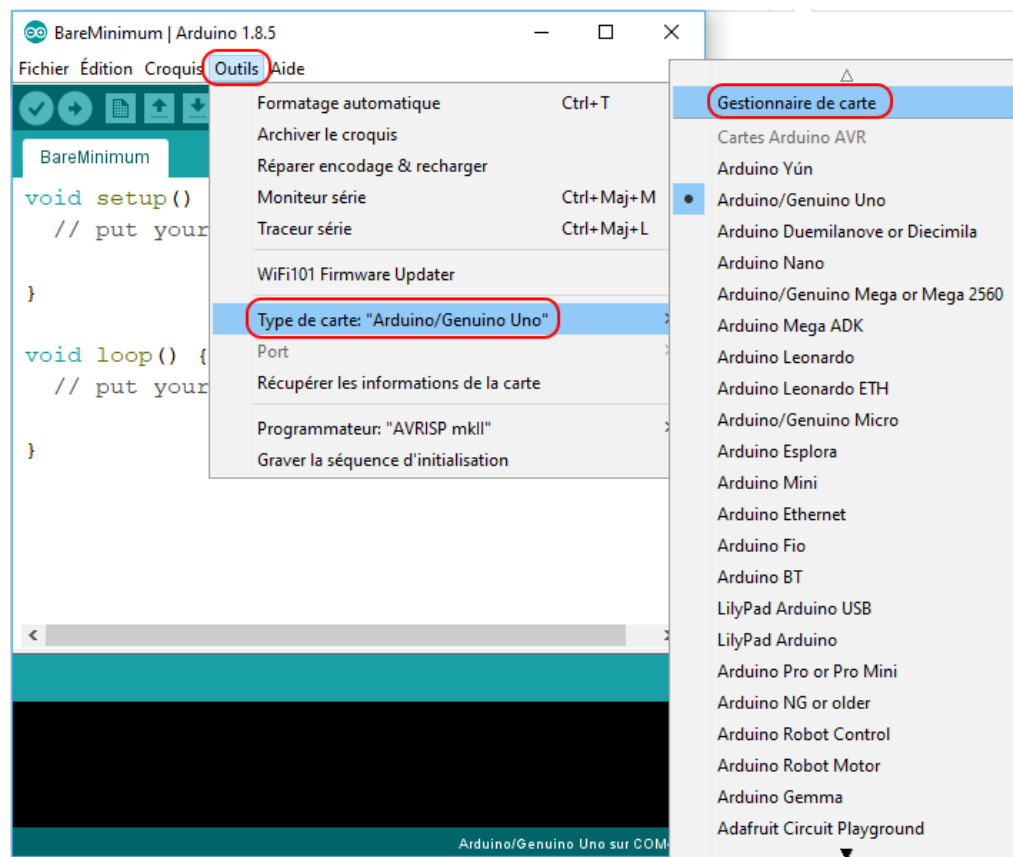


Ici, il faut rentrer l'URL:

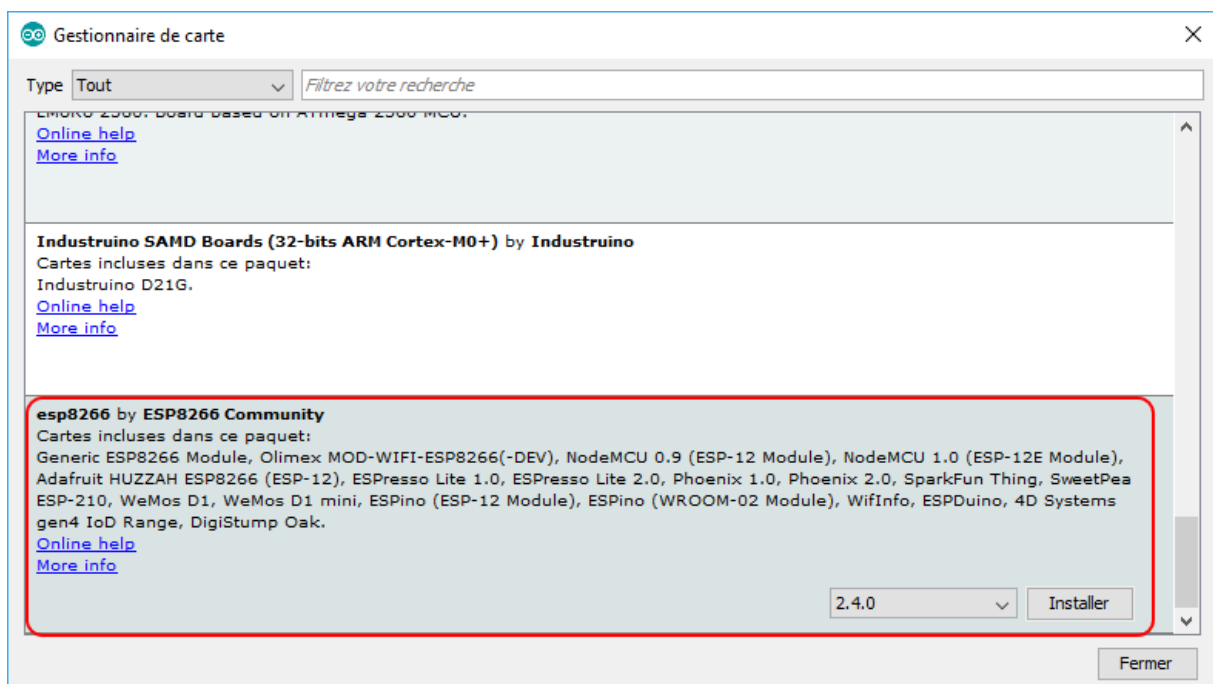
http://arduino.esp8266.com/stable/package_esp8266com_index.json

Plus de détails sur le site [ESP8266 Arduino Core](https://arduino-esp8266.readthedocs.io/en/2.4.0/index.html). <https://arduino-esp8266.readthedocs.io/en/2.4.0/index.html>

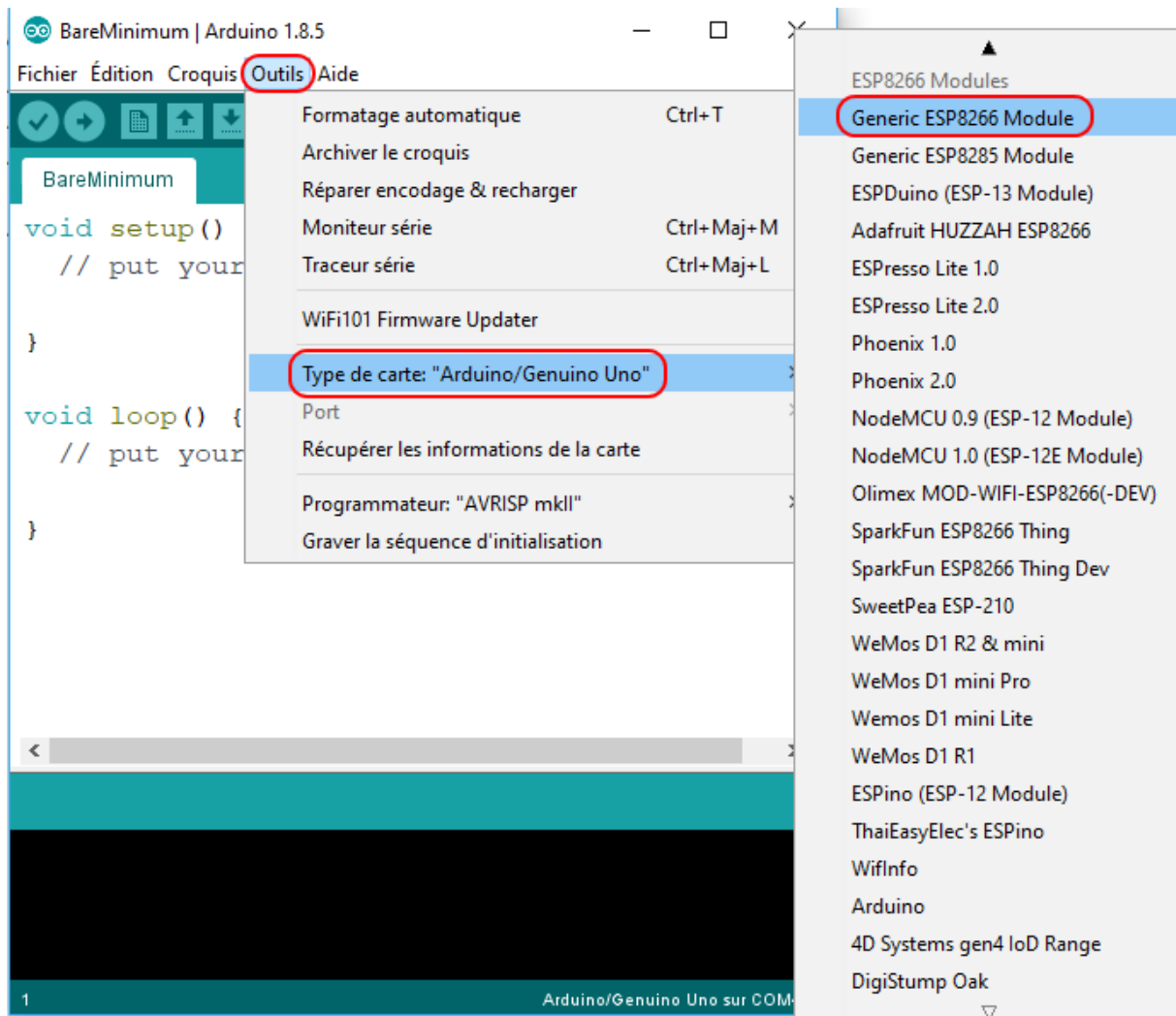
Sélectionnez ensuite le Gestionnaire de cartes (Boards Manager) :



Dans le gestionnaire, sélectionnez le paquet esp8266 :



Vous pouvez maintenant choisir la cible Generic ESP8266 Module :



Dans notre EDI, on doit téléverser un programme vide à notre Arduino afin d'éviter les erreurs indésirables, voilà un programme vide comment peut être écrit:

```
Void setup(){}  
Void loop(){}  

```

Après le téléversement :

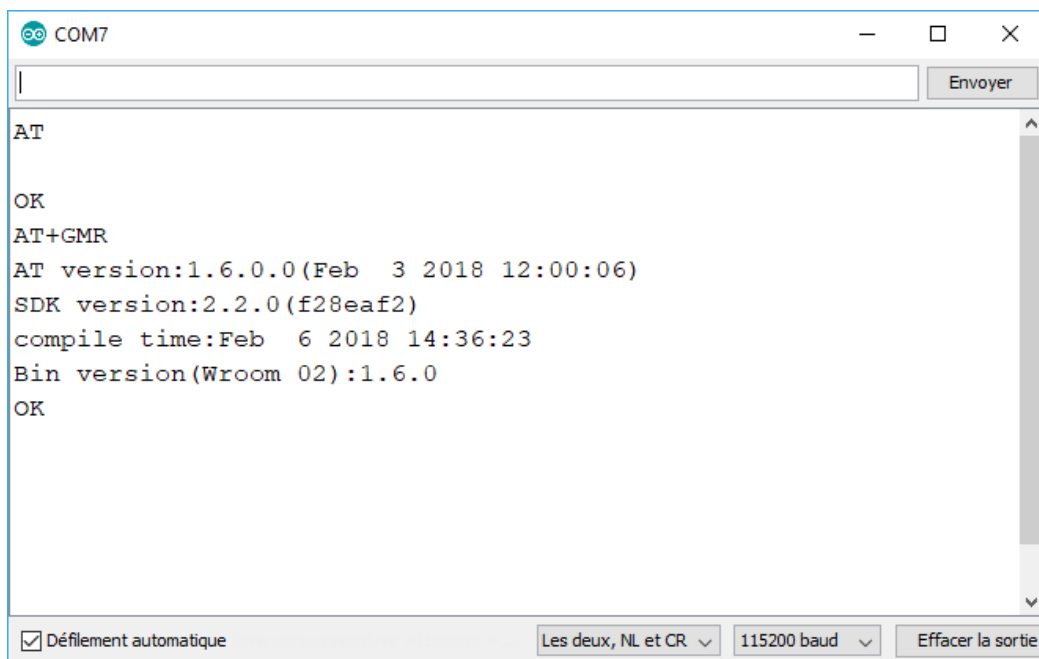
- 1- Ouvrez le Moniteur Série (Serial Monitor) ;
- 2- Sélectionnez la vitesse de transmission, normalement 115 200 bauds (baud rate), mais selon le firmware, elle peut être inférieure ;
- 3- Sélectionnez « Les deux, NL et CR » (Both NL & CR), pour envoyer un New Line et Carriage Return à chaque envoi (bouton Envoyer) ;
- 4- Tapez (sans les guillemets) la commande « AT », puis cliquez sur le bouton « Envoyer ». L'ESP devrait renvoyer « OK » dans le Moniteur Série. Si ce n'est

pas le cas, pressez le bouton RESET ou essayez à nouveau en redémarrant après avoir débranché le câble USB. Vérifiez aussi que vous avez sélectionné le bon port COM. Si l'ESP ne répond toujours pas, peut-être que le firmware « standard » n'est pas installé et qu'il ne comprend pas les commandes AT. Dans ce cas, cela n'est pas vraiment un problème puisque vous allez réécrire par-dessus le firmware avec votre propre code ;

5- Tapez « AT+GMR ». L'ESP retourne le numéro de version du firmware.

NB : on aura besoin que notre module communique sur le baud rate 9600 .

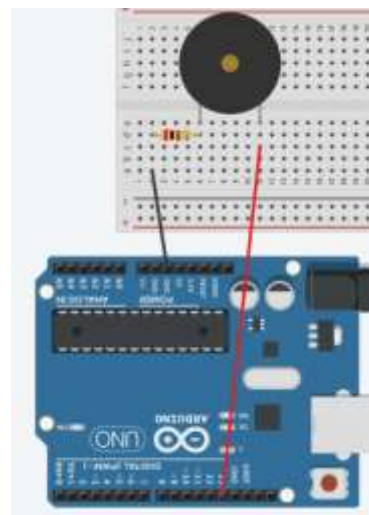
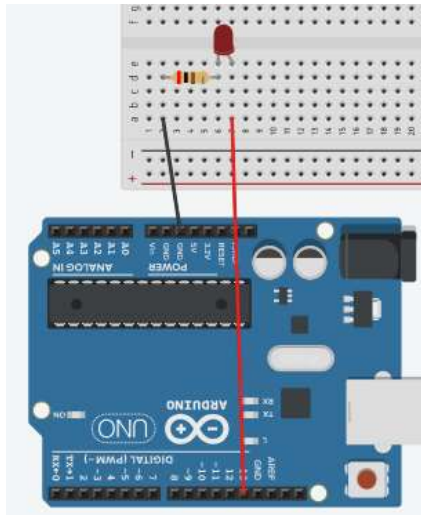
C'est pour cela Tapez « AT+UART=9600,8,1,0,0 » afin de changer la vitesse de transmission.



Vous trouverez dans [ce lien](#) la liste des commandes AT principaux afin de se connecter à un réseau wifi et effectuer d'autres réglages.

4) LED et Sirène :
























Pour LED et Sirène leurs montage est très simple :



4-Application web :

D'abord on doit créer un site web qui contient deux pages , une reçoit les données détectés et elle les enregistre dans la base de données « esp.php » , et une autre page qui affiche ces données à partir de la base de données « espdetails.php » .

Pour cela on commence par créer la base de données et dans cette dernière on crée une table qui s'appelle « informations » où on va enregistrer nos données.

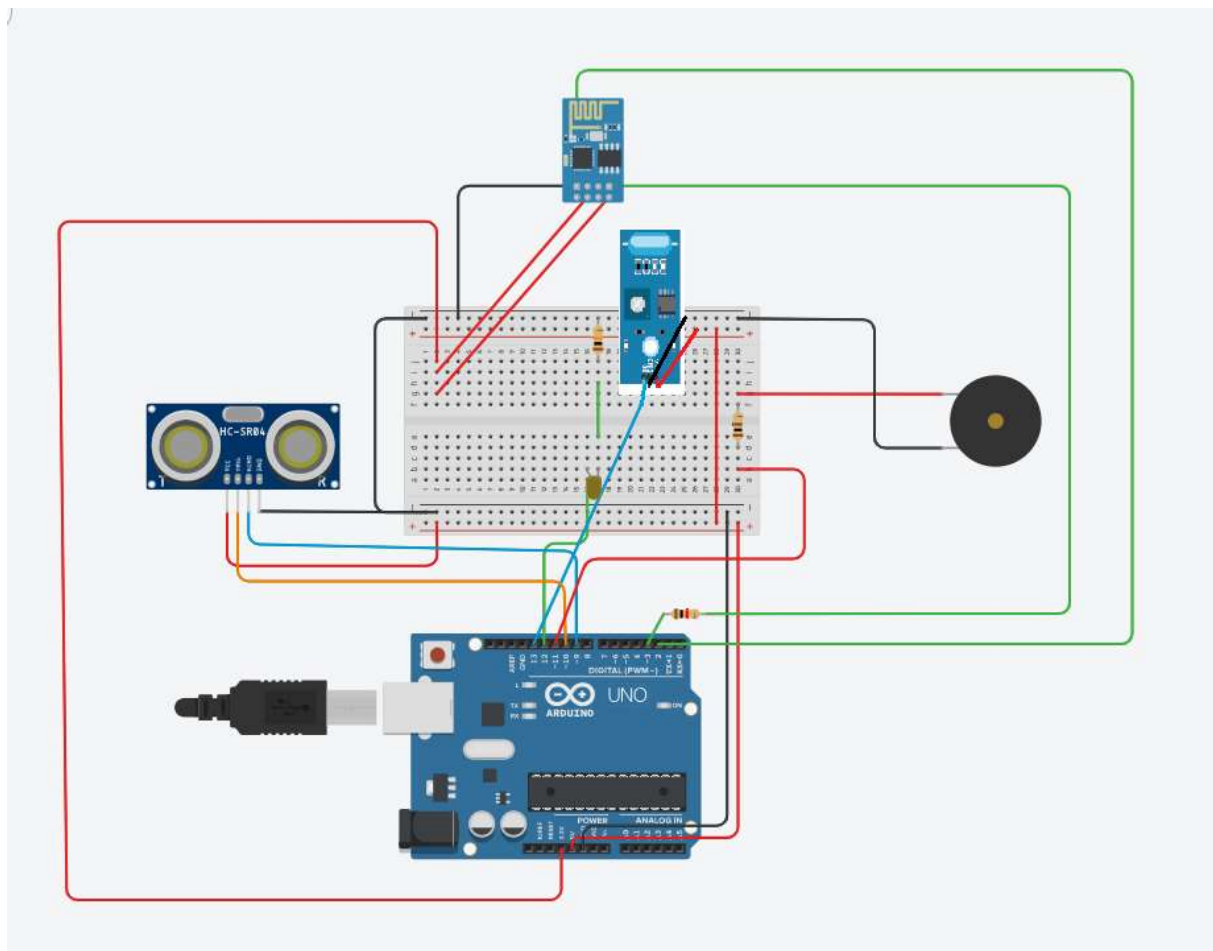
+ Options											
  				ID	DANGER	DISTANCE	VIBRATION	TEMPS			
		Éditer		Copier		Supprimer	2	1	24cm	1	2019-02-05 02:28:31
		Éditer		Copier		Supprimer	3	1	24cm	1	2019-02-05 02:28:43
		Éditer		Copier		Supprimer	4	1	24cm	1	2019-02-05 02:29:58
		Éditer		Copier		Supprimer	5	1	19cm	1	2019-02-05 04:13:22
		Éditer		Copier		Supprimer	6	0	24cm	0	2019-02-05 04:41:01
 <input type="checkbox"/> Tout cocher Avec la sélection :  Éditer  Copier  Supprimer  Exporter											

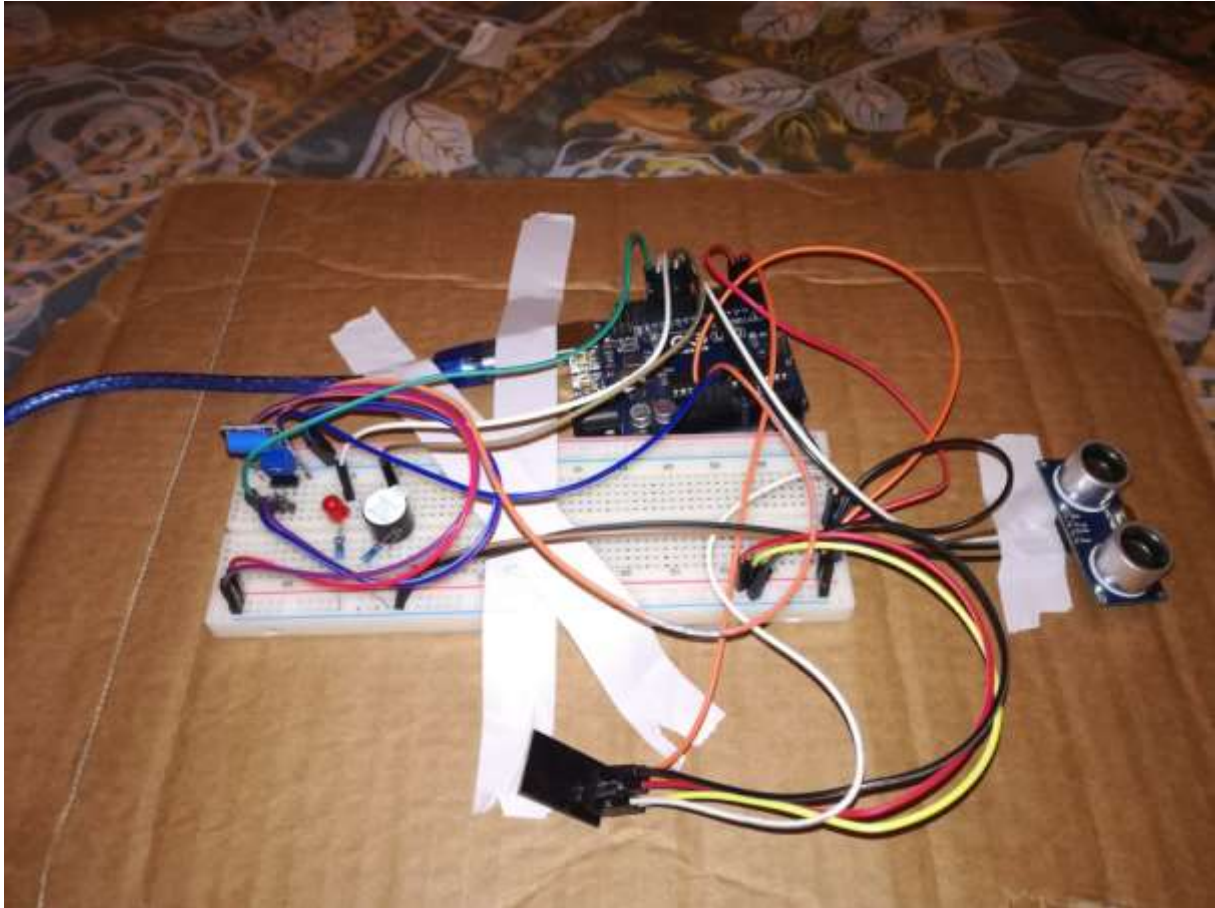
Vous trouvez dans [ce lien](#) les fichiers qu'on avait utilisé :
esp.php , espdetails.php et la base de données esp.sql .

5-Programme de système :

Avant de commencer on doit réaliser le circuit final, voilà comment il sera :

<p>ESP :</p> <p>ESP RX -> Resistance 6Kohm -> ARDUINO 3</p> <p>ESP TX -> ARDUINO 2</p> <p>ESP VCC -> ARDUINO 3.3V</p> <p>ESP CH_PD -> ARDUINO 3.3V</p> <p>ESP GND -> ARDUINO GND</p>	<p>Capteur de vibration :</p> <p>Pin 13 de l'Arduino -> D0 du module.</p> <p>GND de l'Arduino -> GND du module.</p> <p>5V de l'Arduino -> VCC du module</p>
<p>Capteur de distance :</p> <p>Vcc de capteur -> 5V de l'arduino</p> <p>Trig de capteur -> pin 10 de l'arduino</p> <p>Echo de capteur -> pin 9 de l'arduino</p> <p>GND de capteur -> GND de l'arduino</p>	<p>LED :</p> <p>pin 12 d'arduino -> led -> resistance 220 ohm -> gnd d'arduino</p> <p>Sirène :</p> <p>Pin 11 -> sirène -> resistance 220ohm-> gnd d'arduino</p>





Vous devez d'abord comprendre les bases d'un protocole HTTP. Le protocole HTTP (Hypertext Transfer Protocol) est conçu pour permettre la communication entre clients et serveurs.

HTTP fonctionne comme un protocole de requête-réponse entre un client et un serveur. Un navigateur Web peut être le client et une application sur un ordinateur hébergeant un site Web peut être le serveur. Exemple: un client (navigateur) soumet une requête HTTP au serveur; puis le serveur renvoie une réponse au client. La réponse contient des informations d'état sur la demande et peut également contenir le contenu demandé. Sur notre exemple, le esp8266 est le client et le serveur qui héberge notre site Web est le serveur.

Donc, effectuer une requête HTTP Post doit être sous une certaine forme :

POST /esppost.php HTTP/1.0

Host: serverconnect.site88.net

Accept: */*

Content-Length: "name1=value1&name2=value2".Length

Content-Type: application/x-www-form-urlencoded

name1=value1&name2=value2

Vous trouverez le fichier esp.ino dans [ce lien](#), vous le transférez vers votre arduino et normalement tout doit bien marcher.

6-Conclusion :

Ce projet est une bonne pratique pour s'initier au monde d'internet des objets, et apprendre comment mélanger deux domaines différents électronique et informatique.

J'espère que vous allez bien arriver à réaliser ce projet mais ce n'est pas la fin, il y a plein d'autres choses que vous pouvez faire, comme par exemple, créer une application Android qui affichera les données au lieu de les afficher sur un site Web un peu ennuyeux , ou ajouter une caméra qui prend la photo de voleur et l'envoie à son administrateur.