

Domotique

Projet Final Domotique

Dans le cadre de la formation licence professionnelle automatique et informatique industrielle, nous avons à réaliser un projet domotique.

Nous avons choisi de réaliser un robot suiveur de ligne, détecteur de couleur et anti collision. Le but de ce projet est de programmer, comprendre les capteurs que nous allons utiliser et de pouvoir commander un robot à l'aide d'un site web connecté en Wi-Fi, le laisser réaliser un chemin en suivant une ligne blanche, ou qu'il suive une ligne colorée.

Au fur et à mesure le projet a évolué. Dans un premier temps, nous avons travaillé à l'aide d'une carte de programmation Arduino, de type Mega. Le but était de tester dans un premier temps les différents capteurs que nous souhaitons ensuite appliquer à un ESP8266 NodeMCU V1.0 qui embarque un microcontrôleur et un module Wifi. Nous expliquerons également comment installer les plug-in associé à cette carte Esp8266 NodeMCU V1.0.

Quant à l'ESP8266, nous avons testé la connexion en websocket avec une application Android.

Dans un premier temps nous allons voir le matériel mit à notre disposition.

Pour réaliser les tests des différents capteurs nous un Arduino Micro.

Voici la liste des capteurs utilisés :

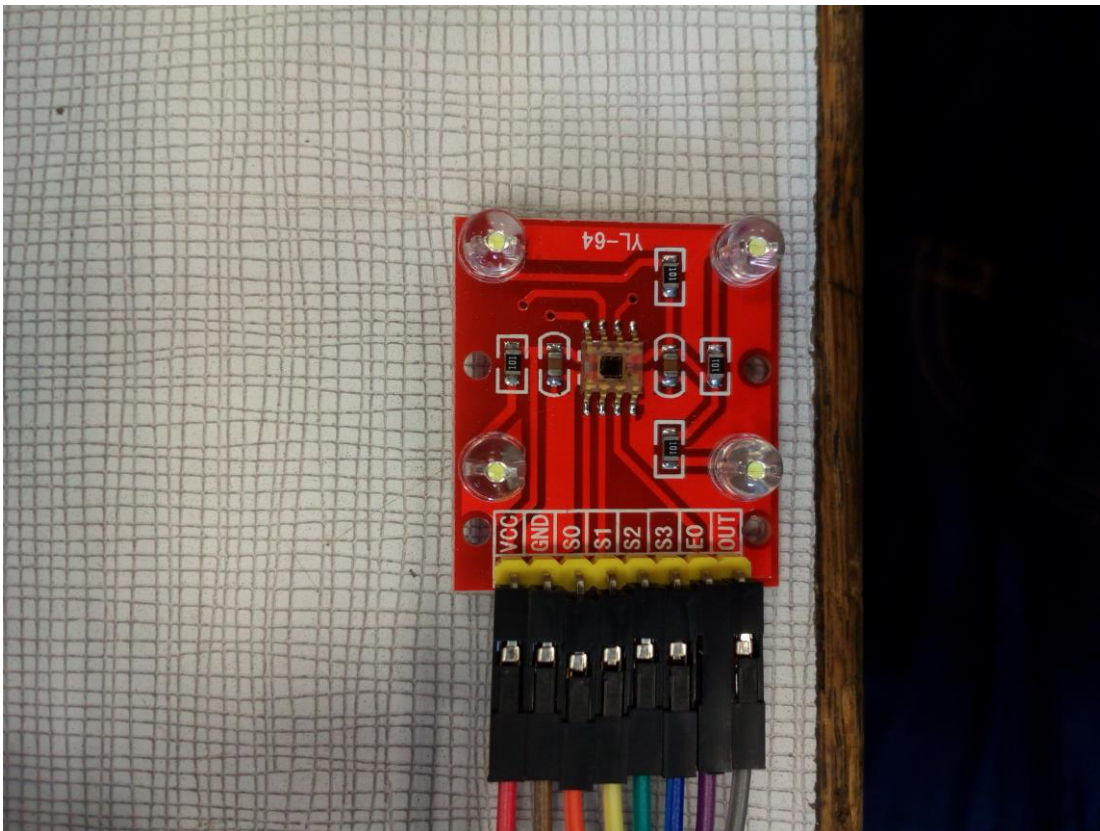
- Capteur de couleur
- Hacheur
- Capteur ultrason
- Boussole

Dans un premier temps nous allons analyser le capteur de couleurs ainsi que son code.

Voici le capteur de couleur

Comme vous pouvez le voir, on retrouve ici les différents pins qui sont à câbler sur notre Arduino.

On retrouve une alimentation Vcc, la masse GND, des sortie S0, S1, S2, S3, E0, et OUT



Capteur de couleur basé sur le circuit TCS3200 et 4 leds blanches. Le capteur délivre une fréquence en fonction de la couleur. Il est constitué de 4 filtres pour chaque couleur, la fréquence de sortie est proportionnelle à la couleur choisie.

Le module se raccorde sur 5 E/S digitales pour la sélection du filtre et la mesure de la fréquence de sortie d'une carte compatible Arduino. Il peut s'utiliser avec une plaque de montage sans soudures.

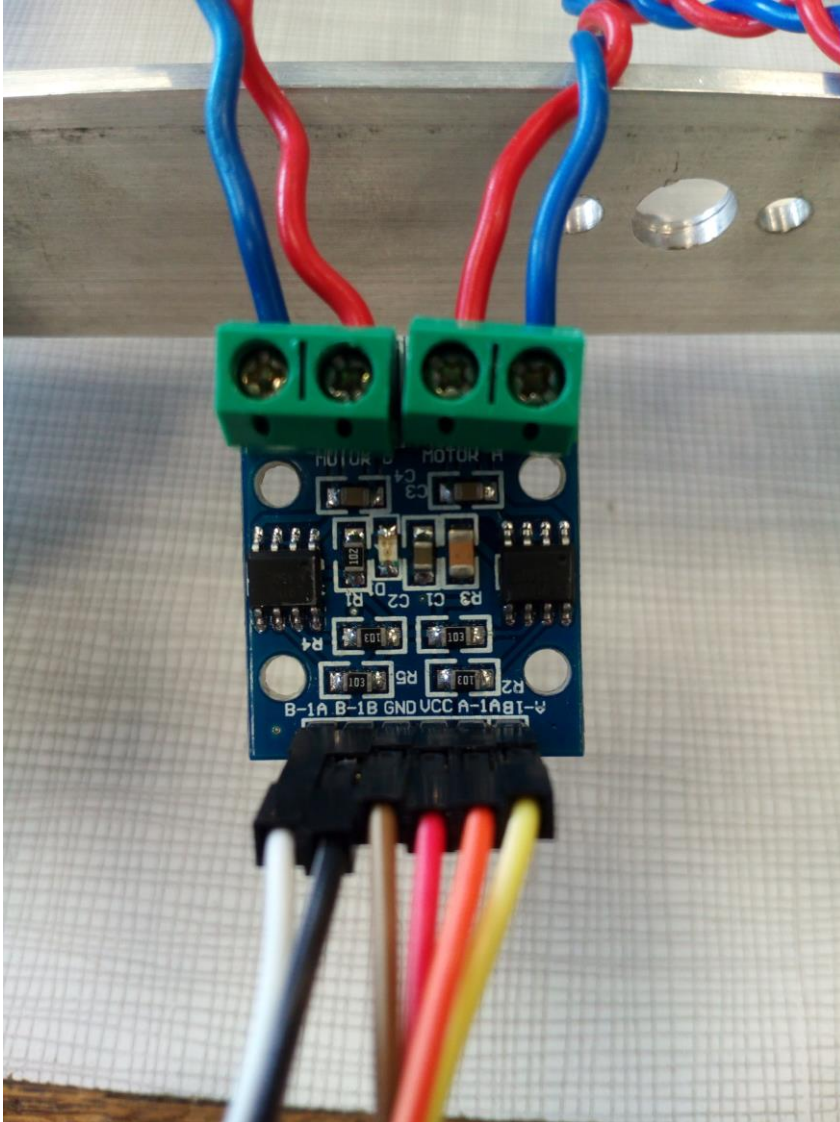
Alimentation : 3 à 5 Vcc

<http://dsp.stackexchange.com/questions/8949/how-do-i-calculate-the-color-temperature-of-the-light-source-illuminating-an-ima>

Hacheur

Le hacheur est un dispositif de l'électronique de puissance mettant en œuvre un ou plusieurs interrupteurs électroniques commandés ce qui permet de modifier la valeur de la tension d'une source de tension continue avec un rendement élevé.

Ici on utilise ce hacheur :



On envoie deux signaux pour 1 moteur la masse et Vcc.

On branche deux moteurs sur chaque module.

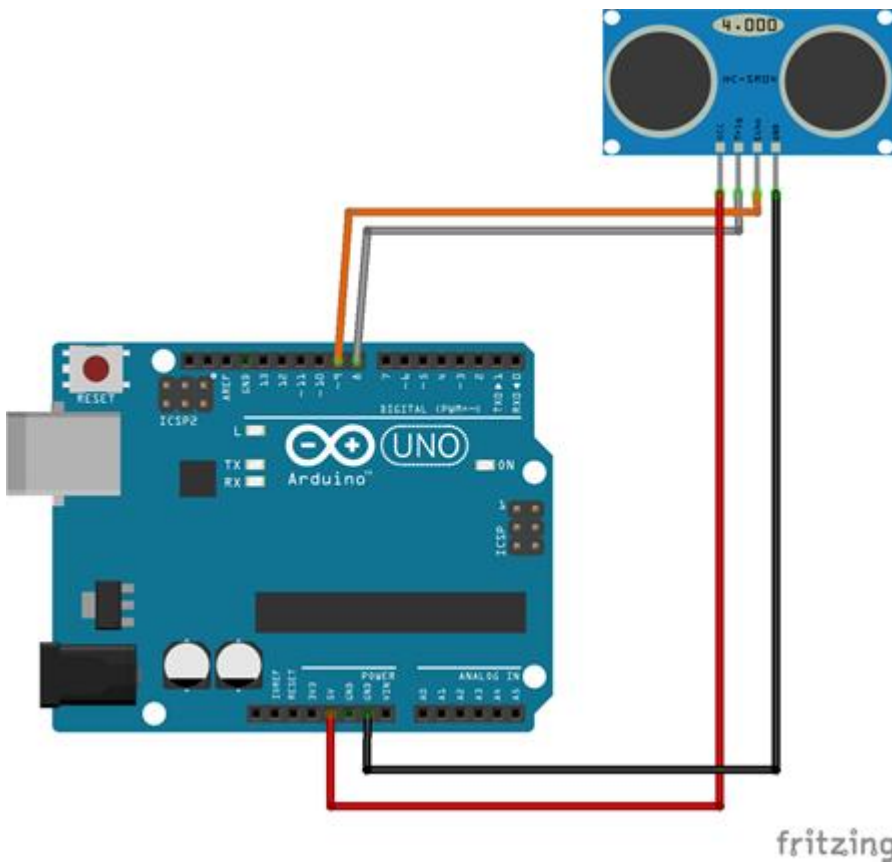
Capteur Ultrason

Nous allons réaliser un montage simple permettant de comprendre le fonctionnement du capteur ultrason HC-SR04.

Pour ce montage, les éléments sont :

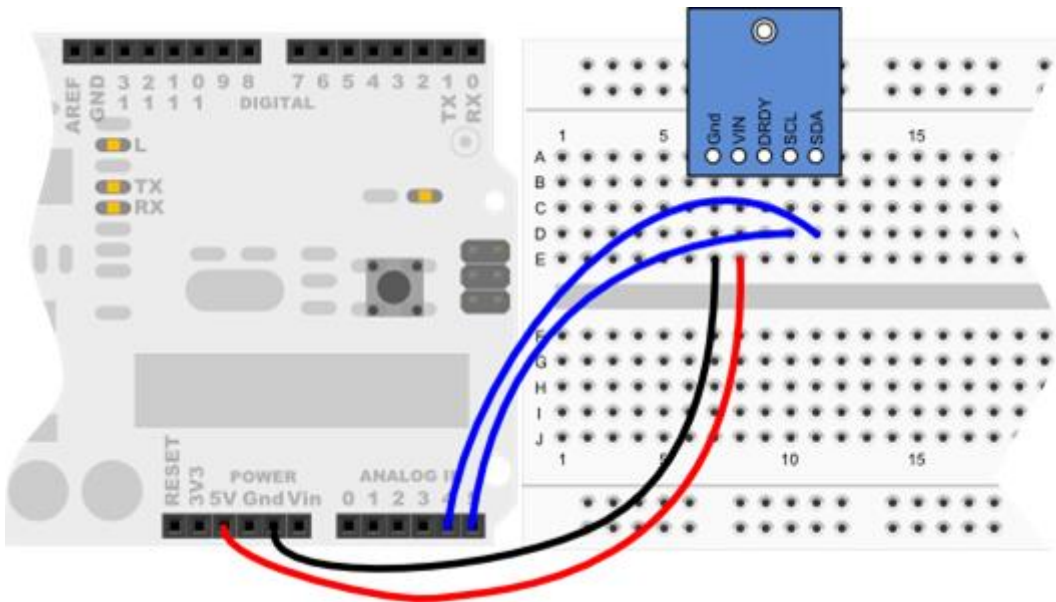
- Arduino UNO
- Capteur ultrason

Le montage électrique :



Boussole HmC5883L

Nous avons également testé une boussole HMC5883L. Elle possède 5pins :



Ce capteur permet d'avoir la position angulaire comme une véritable boussole. Il est possible d'avoir un affichage graphique à l'aide d'un autre logiciel de programmation qui est Processing. Plus d'information sur le site suivant :

<http://arduikyo.blogspot.fr/2013/08/hmc5883l-compass-arduino-processing.html>

Différents codes :

```
// Capteur de couleurs

#include <Metro.h> // library <return>
#include <math.h>
Metro ledlightup = Metro(60,true);
Metro processdata = Metro(50,true);
Metro TcsTrigger = Metro(10,true);

int s0=8,s1=9,s2=10,s3=11; // port definition of color sensor
int out=2;
int flag=0; // initialization
int counter=0;
int countR=0,countG=0,countB=0;
void setup()
{
  Serial.begin(115200);
  pinMode(s0,OUTPUT);
  pinMode(s1,OUTPUT);
  pinMode(s2,OUTPUT);
  pinMode(s3,OUTPUT);
}

void TCS()
{
  digitalWrite(s0,HIGH);
  digitalWrite(s1,HIGH);
```

```

attachInterrupt(0, ISR_INT0, CHANGE);

}

void ISR_INT0()
{
    counter++;
}

int Raverage = 0;
int Baverage = 0;
int Gaverage = 0;

void loop()          // call function
{
    TCS();
    if(TcsTrigger.check()){
        Tcstrigger();
    }
    if(processdata.check()){
        procedata();
    }
    if(ledlightup.check()){
        ledup();
    }
}

void Tcstrigger()    // data acquisition
{
    flag++;
    if(flag==1){
        digitalWrite(s2,LOW);
        digitalWrite(s3,LOW);
        countR=counter;
        digitalWrite(s2,HIGH);
        digitalWrite(s3,HIGH);
    }
    else if(flag==2){
        countG=counter;
        digitalWrite(s2,LOW);
        digitalWrite(s3,HIGH);
    }
    else if(flag==3){
        countB=counter;
        digitalWrite(s2,LOW);
        digitalWrite(s3,LOW);
        flag=0;
    }
    counter=0;
}

void procedata()    // data processing
{
    static int Rinput[5] ={
        0,0,0,0,0
    },Binput[5] ={
        0,0,0,0,0
    },Ginput[5] ={
        0,0,0,0,0
    };
}

```

```
for(int i = 4;i > 0;i--){  
    Rinput[i] = Rinput[i-1];  
    Binput[i] = Binput[i-1];  
    Ginput[i] = Ginput[i-1];  
}
```

```
if(countR < 2500)  
    Rinput[0] = countR;  
else  
    Rinput[0] = Rinput[1];
```

```
if(countB < 2500)  
    Binput[0] = countB;  
else  
    Binput[0] = Binput[1];
```

```
if(countG < 2500)  
    Ginput[0] = countG;  
else  
    Ginput[0] = Ginput[1];
```

```
Raverage = 0;  
Baverage = 0;  
Gaverage = 0;
```

```
for(int i = 0;i <= 4;i++){  
    Raverage += Rinput[i];  
    Baverage += Binput[i];  
    Gaverage += Ginput[i];  
}  
Raverage /= 5;  
Baverage /= 5;  
Gaverage /= 5;
```

```
}
```

```
void ledup()    // data output  
{  
    int ledvalueR = Raverage;  
    int ledvalueG = Gaverage;  
    int ledvalueB = Baverage;
```

```
    ledvalueR = constrain(ledvalueR,350,1700);  
    ledvalueB = constrain(ledvalueB,350,1500);  
    ledvalueG = constrain(ledvalueG,350,1650);  
    Serial.print("R:");  
    Serial.print(Raverage,DEC);  
    Serial.print(" ");  
    Serial.print(ledvalueR,DEC);  
    Serial.print("B:");  
    Serial.print(Baverage,DEC);  
    Serial.print(" ");  
    Serial.print(ledvalueB,DEC);  
    Serial.print("G:");  
    Serial.print(Gaverage,DEC);  
    Serial.print(" ");  
    Serial.println(ledvalueG,DEC);  
}
```



```
//Code Capteur ultrason
```

```
#define trigPin 8
```

```
#define echoPin 9
```

```
int dist = 0;
```

```
void setup ()
```

```
{
```

```
Serial.begin (9600);
```

```
pinMode(trigPin, OUTPUT);
```

```
pinMode(echoPin, INPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
long duration, distance;
```

```
digitalWrite(trigPin, LOW);
```

```
delayMicroseconds(2);
```

```
digitalWrite(trigPin, HIGH);
```

```
delayMicroseconds(10); //Trig déclenché 10ms sur HIGH
```

```
digitalWrite(trigPin, LOW);
```

```
// Calcul de l'écho
```

```
duration = pulseIn(echoPin, HIGH);
```

```
// Distance proportionnelle à la durée de sortie
```

```
distance = duration*340/(2*10000); //Vitesse du son théorique
```

```
dist = distance;
```

```
Serial.print("distance: ");
```

```
Serial.print(dist);
```

```
Serial.println(" cm");
```

```
delay(1000);
```

```
}
```

```

//Hacheur couple ultrason

const int AIA1 = 2;
const int AIB1 = 3;
const int BIA1 = 4;
const int BIB1 = 5;
const int AIA2 = 6;
const int AIB2 = 7;
const int BIA2 = 8;
const int BIB2 = 9;
int trig = 12;
int echo = 11;
long lecture_echo;
long cm;
byte speed1 = 255;

void setup() {
  Serial.begin(9600);
  //PWM Moteur
  pinMode(AIA1, OUTPUT);
  pinMode(AIB1, OUTPUT);
  pinMode(BIA1, OUTPUT);
  pinMode(BIB1, OUTPUT);
  pinMode(AIA2, OUTPUT);
  pinMode(AIB2, OUTPUT);
  pinMode(BIA2, OUTPUT);
  pinMode(BIB2, OUTPUT);
  //Ultrasons
  pinMode(trig, OUTPUT);
  digitalWrite(trig, LOW);
  pinMode(echo, INPUT);
}

void loop() {
  forward();
}

void forward()
{
  analogWrite(AIA1, speed1);
  analogWrite(AIB1, 0);
  analogWrite(BIA1, speed1);
  analogWrite(BIB1, 0);
  analogWrite(AIA2, speed1);
  analogWrite(AIB2, 0);
  analogWrite(BIA2, speed1);
  analogWrite(BIB2, 0);
}

void ultrason ()
{
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  lecture_echo = pulseIn(echo, HIGH);
  cm = lecture_echo / 58;
  Serial.print("Distancem : ");
  Serial.println(cm);
}

```

```
delay(1000);  
}
```

ESP8266 Installation Arduino

Le but de ce projet est de programmer un ESP8266 avec le code exécuté sur notre Arduino UNO afin d'avoir une communication Wifi comprise dans l'ESP8266.

Pour cela nous nous sommes donc intéressé à la bibliothèque ESP8266 compatible Arduino ainsi que d'autres méthodes que nous allons voir.

Dans un premier temps nous allons voir le matériel mit à notre disposition.

Nous avons donc un ESP8266 V1.0 que vous pouvez retrouver en vente sur ce site :

http://e.banana-pi.fr/fr/recherche?controller=search&orderby=position&orderway=desc&search_query=esp8266

Un ordinateur ayant Arduino 1.6.8 installé qui est la dernière version à jour d'Arduino :

<https://www.arduino.cc/en/Main/Software>

Pour ajouter les plug-in nécessaire à l'outils de programmation de l'ESP8266 sur Arduino, je vous invite à suivre les instructions suivantes:

1.

Ajouter le lien de référence au développement de l'ESP8266.

Pour cela

Fichier -> Préférences

Copier le lien suivant dans **URL de gestionnaire de cartes supplémentaires** :

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

2.

Pour vérifier si votre lien est correct, aller dans :

Outils -> Type de carte -> Gestionnaire de carte

Lancez la recherche '**ESP8266**'

Puis Installez

Pendant ce temps je vous invite à vous boire un petit café si vous êtes matinal, sinon un petit thé fera l'affaire !

Une fois installé vous allez avoir accès à toute les fonctionnalités sur Arduino pour programmer votre ESP8266.

Tous simplement je vous demande de vous diriger vers '**Outils -> Type de carte**'

Descendez jusqu'à la petite flèche et vous verrez que vous aurez à votre disposition bien plus de type de carte dont la ESP8266 NodeMCU 1.0, correspondant à celle du site que je vous ai partagé précédemment.

Sélectionnez la.

Une panoplie d'options :

- * La fréquence du CPU : 80MHz ou 16MHz
- * La taille de la mémoire Flash
- * La fréquence de téléversement et communication avec le PC

Je vous conseille de ne laisser les paramètres de bases si vous n'êtes pas expert, cela ne vous apportera rien de modifier les valeurs.

Seul le Bauderate peut être utile selon les capteurs que vous utiliserez sur votre ESP8266 NodeMCU 1.0

Vous voilà fin prêt à programmer votre ESP8266 avec Arduino.

Bonne programmation !!

Autre Option : ESP8266 Installation Eclipse Arduino IDE

Pour cette installation je vous invite à suivre le Tuto du créateur qui est très bien fait et assez simple à réaliser.

Il a également fait des vidéos explicatives pour installer son plug-in.

https://www.youtube.com/playlist?list=PLCSjf7QmP3TDPSjT_5UIp0DHXzKdhlQsF

<http://eclipse.baeyens.it/>

ESP8266 Websocket Application

Pour commencer à travailler avec l'ESP8266, nous avons récupéré un code de websocket et nous avons modifié ce programme afin de pouvoir communiquer avec, et envoyé un message à l'ESP8266 à l'aide d'un Smartphone Android.

Sur ce Smartphone Android, nous avons récupéré une application Android sur le Play Store :



Cette application permet de se connecter au Websocket de l'ESP8266 et d'envoyer un message.

