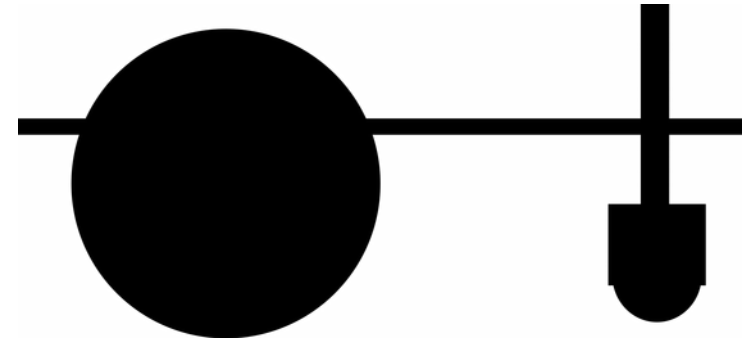
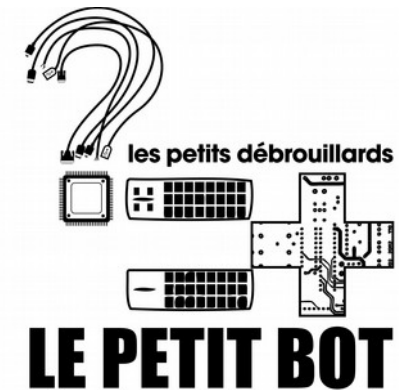
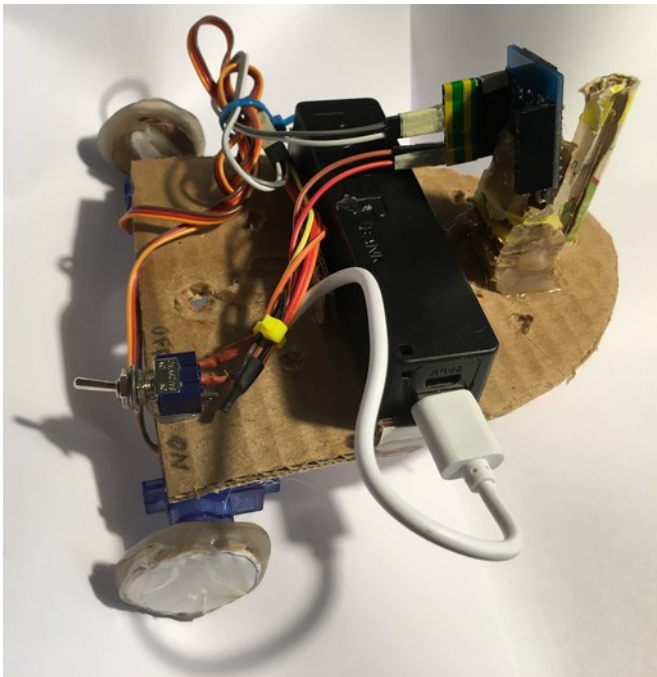


////////////////////
// Petit Bot //
////////////////////

// Un programme pédagogique des petits débrouillards ?=+ pour gérer le robot "Petit Bot"
// Voir sur http://wikidebrouillard.org/index.php?title=Petit_Bot_un_robot_controlable_en_Wifi
// Ce programme est inspire de : <http://www.esp8266.com/viewtopic.php?f=29&t=6419#sthash.gd1tJhwU.dpuf>
// Sous licence CC-BY-Sa
// Par des gens bien



////////////////////DECLARATIONS////////////////////////////////////

// on appelle la bibliothèque qui gère le Wemos D1 mini

#include <ESP8266WiFi.h>

// Gestion du Wifi

#include <ESP8266WebServer.h>

#include <DNSServer.h>

#include <ESP8266mDNS.h>

#include <WiFiClient.h>

// Definition du WiFi

const char *nomDuReseau = "petitbot"; **// Nom du réseau wifi du petit bot**

const char *motDePasse = "mot-de-passe"; **// Mot de passe du réseau wifi du petit bot**

// ATTENTION - le mot de passe doit faire soit 0 caractères soit 8 ou plus sinon ça marche pas

//création du serveur

ESP8266WebServer monServeur(80); **// Création de l'objet « monServeur »**

//Gestion des servomoteurs

#include <Servo.h> **// Appel de la bibliothèque qui gère les servomoteurs**

Servo servogauche; **// Déclaration du servo qui gère la première roue**

Servo servodroit; **// Seconde roue**

// Déclaration des Variables

//int --- en cours, partie du programme en cours de développement

int val = -1; **// La variable « val » nous sert à stocker la commande de l'utilisateur (stop, avance, ...).**

/////////////////INITIALISATION DU PROGRAMME/////////////////
(n'est fait qu'une seule fois)

```
void setup(){
  delay(1000);
  Serial.begin(9600); // Ouvre une connexion série pour monitorer le fonctionnement du code quand
on reste branché a l'ordinateur
  Serial.println();
  Serial.println();
  configDuWifi();
  configDuServeur() ;
  monServeur.begin() ;
  servodroit.detach(); // Place les servos hors tension
  servogauche.detach();
  pinMode(LED_BUILTIN, OUTPUT); // met la led du Wemos en sortie
  digitalWrite(LED_BUILTIN, LOW); // met la led du Wemos sur le niveau bas ce qui l'allume.
  pinMode(LED, OUTPUT);
  digitalWrite(LED, HIGH);
}
```

/////////////////BOUCLE PRINCIPALE/////////////////
(se répète indéfiniment)

```
void loop(){
  val = -1;
  monServeur.handleClient(); // Le serveur « monServeur » attend la connexion des navigateurs
}
```

//////////GESTION DES INSTRUCTIONS//////////

```
void configDuServeur() {
```

```
  monServeur.on("/avance", HTTP_GET, []() { // Si clic sur l'URL « http://192.168.4.1/avance »  
    val = 1; // alors val = 1, c'est pour les conditions de la fonction « redactionPageWeb »  
    Serial.println("avance"); // Ecrit sur le moniteur série « avance »  
    redactionPageWeb(); // Appelle la fonction « redactionPageWeb »  
  });
```

```
  monServeur.on("/recule", HTTP_GET, []() {  
    val = 2;  
    Serial.println("recule");  
    redactionPageWeb();  
  });
```

```
  monServeur.on("/droite", HTTP_GET, []() {  
    val = 4;  
    Serial.println("droite");  
    redactionPageWeb();  
  });
```

```
  monServeur.on("/gauche", HTTP_GET, []() {  
    val = 3;  
    Serial.println("gauche");  
    redactionPageWeb();  
  });
```

```
monServeur.on("/stop", HTTP_GET, []() {  
  val = 0;  
  Serial.println("stop");  
  redactionPageWeb();  
});
```

```
monServeur.on("/", HTTP_GET, []() {  
  val = -1;  
  redactionPageWeb();  
});  
}
```

//////////LA PAGE WEB DE CONROLE DU PETIT BOT//////////

```
void redactionPageWeb(){
```

```
    // Prépare la page web de réponse (le code HTML sera écrit dans la chaine de caractère  
    "pageWeb").
```

```
    String pageWeb = "<!DOCTYPE HTML>\r\n";
```

```
    pageWeb += "<html>\r\n";
```

```
    pageWeb += "<center>";    // On ouvre la balise qui va centrer les boutons
```

```
    pageWeb += "<h1 style=\"font-size:300%;\"> Le petit bot ";
```

```
    pageWeb += "<style type=\"text/css\">  body { color: #212121; background-color: #CC0C59 } </style>";
```

```
    // On finalise l'écriture de la page Web et on donne les instructions aux servos
```

```
    pageWeb += instruction(val); // pour cela on appelle la fonction "instruction" en lui envoyant « val »
```

```
    // On termine l'écriture de la page Web
```

```
    pageWeb += "</h1>";
```

```
    pageWeb += "<br>"; //aller à la ligne
```

```
    pageWeb += "<br>"; //aller à la ligne
```

```
    pageWeb += "<a href=\"/stop\"><button style=\"font-size:200%; width: 18%; background-color:#0CCC16;  
border-radius: 12px\">Stop </button></a>\r\n";    // créer un bouton "Stop", qui envoie sur l'URL /stop
```

```
    pageWeb += "<a href=\"/avance\"><button style=\"font-size:200%; width: 18%; background-  
color:#0CCC16; border-radius: 12px\">Avance </button></a>\r\n"; // créer un bouton "Avance"...
```

```
    pageWeb += "<a href=\"/recule\"><button style=\"font-size:200%; width: 18%; background-  
color:#0CCC16; border-radius: 12px\">Recule </button></a>\r\n";
```

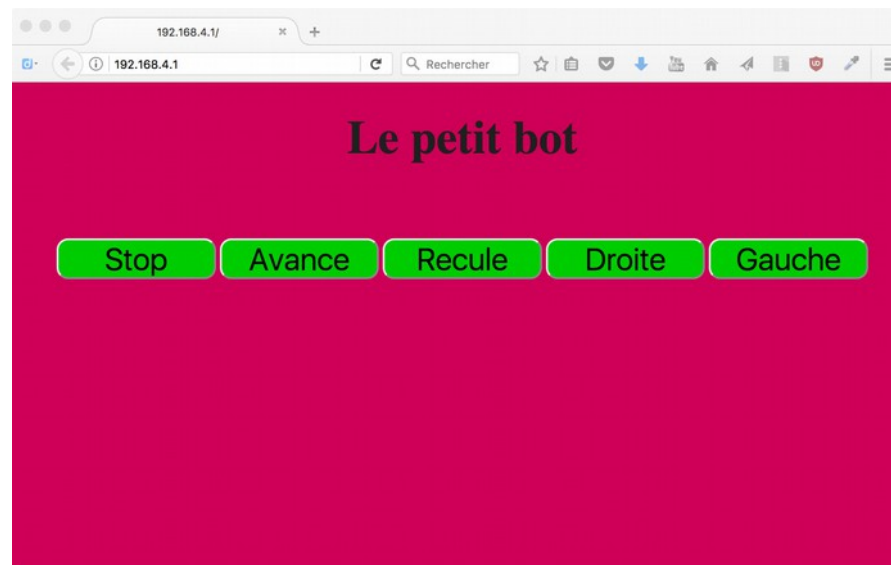
```
    pageWeb += "<a href=\"/droite\"><button style=\"font-size:200%; width: 18%; background-  
color:#0CCC16; border-radius: 12px\">Droite </button></a>\r\n";
```

```
    pageWeb += "<a href=\"/gauche\"><button style=\"font-size:200%; width: 18%; background-
```

```
color:#0CCC16; border-radius: 12px\">Gauche </button></a><br />\r\n";  
pageWeb += "</center>"; // tout est centré sur la page  
pageWeb += "</html>\n"; // Fin de la page Web
```

// On envoie la page web

```
server.send(200, "text/html", pageWeb);  
delay(1);  
}
```



/////////////////INSTRUCTIONS/////////////////

```
String instruction(int valeur){ // Cette fonction traite les instructions envoyées par la donnée « val »  
// placée dans la variable local « valeur » et renvoie une chaine de caractère « completePage ».  
    int gauche;                // Variable dont la valeur fait tourner le servo gauche  
    int droite;                // Idem pour le servo droit  
    String completePage;       // Déclaration de la chaine de caractère qui sera renvoyée par cette fonction  
pour compléter la page web  
    switch(valeur){             // En fonction de la variable valeur on va donner un ordre aux servos  
        case 0 :                // et un texte à la chaine de caractère "completePage"  
            completePage = " est a l'arrêt";  
            droite = 90; // immobile  
            gauche = 90;  
            break;  
        case 1 :  
            completePage = " avance ";  
            droite = 180; // Tourne dans le sens des aiguille d'une montre  
            gauche = 0; // dans l'autre sens  
            break;  
        case 2 :  
            completePage = " recule ";  
            droite = 0;  
            gauche = 180;  
            break;  
        case 3 :  
            completePage = " tourne a gauche ";  
            droite = 180;
```



```
gauche = 180;
break;
case 4 :
completePage = " tourne a droite ";
droite = 0;
gauche = 0;
break;
}
servogauche.attach(D1);    // Broche D1
servodroit.attach(D2);    // Broche D2
servogauche.write(gauche);
servodroit.write(droite);
return completePage;      // on renvoie la chaine de caractère pour compléter la page web
}
```

/////////////////CONFIGURATION WIFI/////////////////

```
void configDuWifi(){ // Fonction de configuration du Wifi
  WiFi.mode(WIFI_AP); // le wemos est en mode "Point d'Accès" (il déploie un réseau wifi)
  WiFi.softAP(nomDuReseau, motDePasse, 2); // on démarre le "Point d'Accès".
  MDNS.begin(nomDuReseau); // gérer les DNS ce qui rendra votre petit bot accessible
  MDNS.addService("http", "tcp", 80); // via http://nomDuReseau.local
  IPAddress monIP = WiFi.softAPIP(); // on récupère l'adresse IP du petit Bot
  Serial.print("Adresse IP de ce Point d'Accès : ");
  Serial.println(monIP); // on l'écrit sur le moniteur série
  return; // on retourne à l'endroit où la fonction a été appelée.
}
```

