Le module Bluetooth HC-05

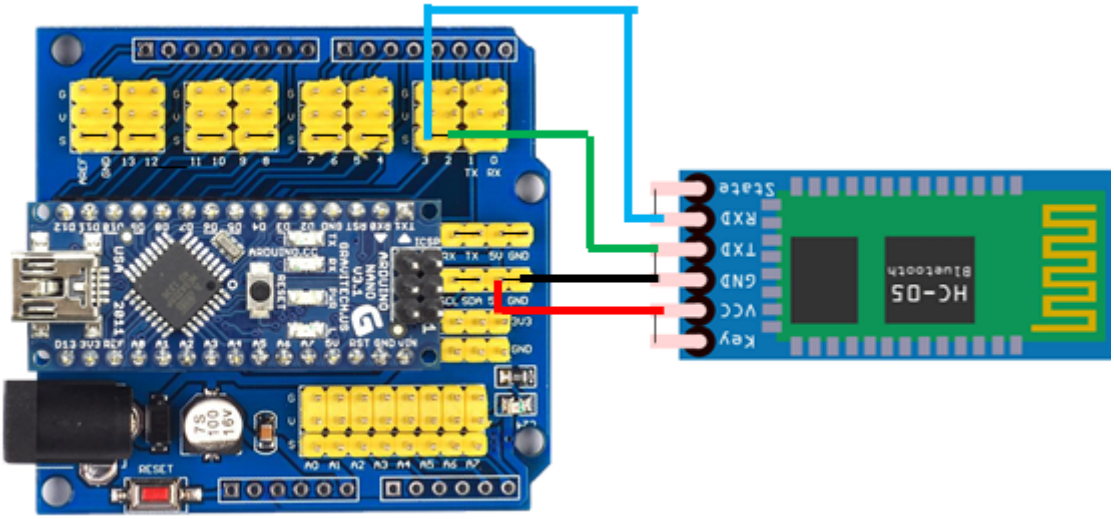
Test et configuration du module

- Suivre les instructions pour le module HC-05, qui peut être configuré en module maître ou esclave. Le module HC-06 lui ne peut être que esclave, mais il est plus simple à configurer.

https://www.aranacorp.com/fr/votre-arduino-communique-avec-le-module-hc-05/

- On peut communiquer avec le *module bluetooth HC-05* via la liaison série de l'*Arduino Nano*
- Se renseigner sur ce qu'est une liaison série, par ex. ici :

  https://arduino.blaisepascal.fr/bibliotheque-serial/
- Réaliser le câblage entre l'Arduino Nano et le module bluetooth HC-05



- Bien vérifier que le Pin qui **R**eçoit les données sur l'Arduino (**R**xDpin) soit connecté au Pin du HC-05 qui **T**ansmet (TxDpin). Et vice-versa, donc chaque câble transmet les données dans une direction opposée.
- Si on prend les Pins RX (Digital 0) et TX (Digital 1) de l'arduino pour connecter RXD et TXD du HC-05, cela rentre en conflit avec la connexion USB entre l'arduino et le PC.
- Donc on prend les Pins Digital 2 et 3 et on utilise la bibliothèque `SoftwareSerial`

```
//Test et Configuration du module Bluetooth HC-05

//Le port série matériel de l'Arduino Nano (Pins 0/RX et 1/TX) est déjà utilisé pour la liaison Arduino-USB avec l'ordinateur
//On utilise donc une liaison série logicielle pour la liaison Arduino-HC05 avec le module Bluetooth
#include <SoftwareSerial.h>  //Software Serial Port
#define RxDpin 2    //Pin Digital 2 pour arduino Rx (pin0=serial)
#define TxDpin 3    //Pin Digital 3 pour arduino Tx (pin1)
//Pour la liaison SoftwareSerial Arduino-HC05, quand on veut configurer le HC-05
//on maintient le bouton à côté de la PIN EN/KEY enfoncé au démarrage de l'Arduino
#define baudrate 38400 //Vitesse pour la liaion Arduino-HC05 en mode configuration
//Pour la liaison SoftwareSerial Arduino-HC05, quand on veut communiquer à travers le bluetooth depuis un smartphone,
//on appaire le HC-05 depuis l'appli avec le mot-de-passe par défaut : 1234
// #define baudrate 9600 //Vitesse pour la liaion Arduino-HC05 en mode utilisation smartphone
SoftwareSerial BTSerie(RxDpin,TxDpin);

char caractereTexte;
String phraseTexte;

void setup()

{
Serial.begin(9600); //Vitesse (baudRate) pour la liaison Arduino-USB
delay(500);
Serial.println("En mode communication USB - Pret pour les commandes AT");
Serial.println("Le HC-05 doit clignoter lentement (2 secondes)");

//Configuration de la liaison SoftwareSerial avec le HC-05
pinMode(RxDpin, INPUT);   //Configuration du Pin RxD (Receive) en mode entrée
pinMode(TxDpin, OUTPUT);  //Configuration du Pin TxD (Transmit) en mode sortie
BTSerie.begin(baudrate);  //Vitesse pour la liaion Arduino-HC05
if (baudrate==38400) {
  Serial.println("En mode communication USB - Pret pour les commandes AT");
  Serial.println("Le HC-05 doit clignoter lentement (2 secondes)");
}
else if (baudrate==9600){
  Serial.println("En mode smartphone - Pret pour être appairé");
```

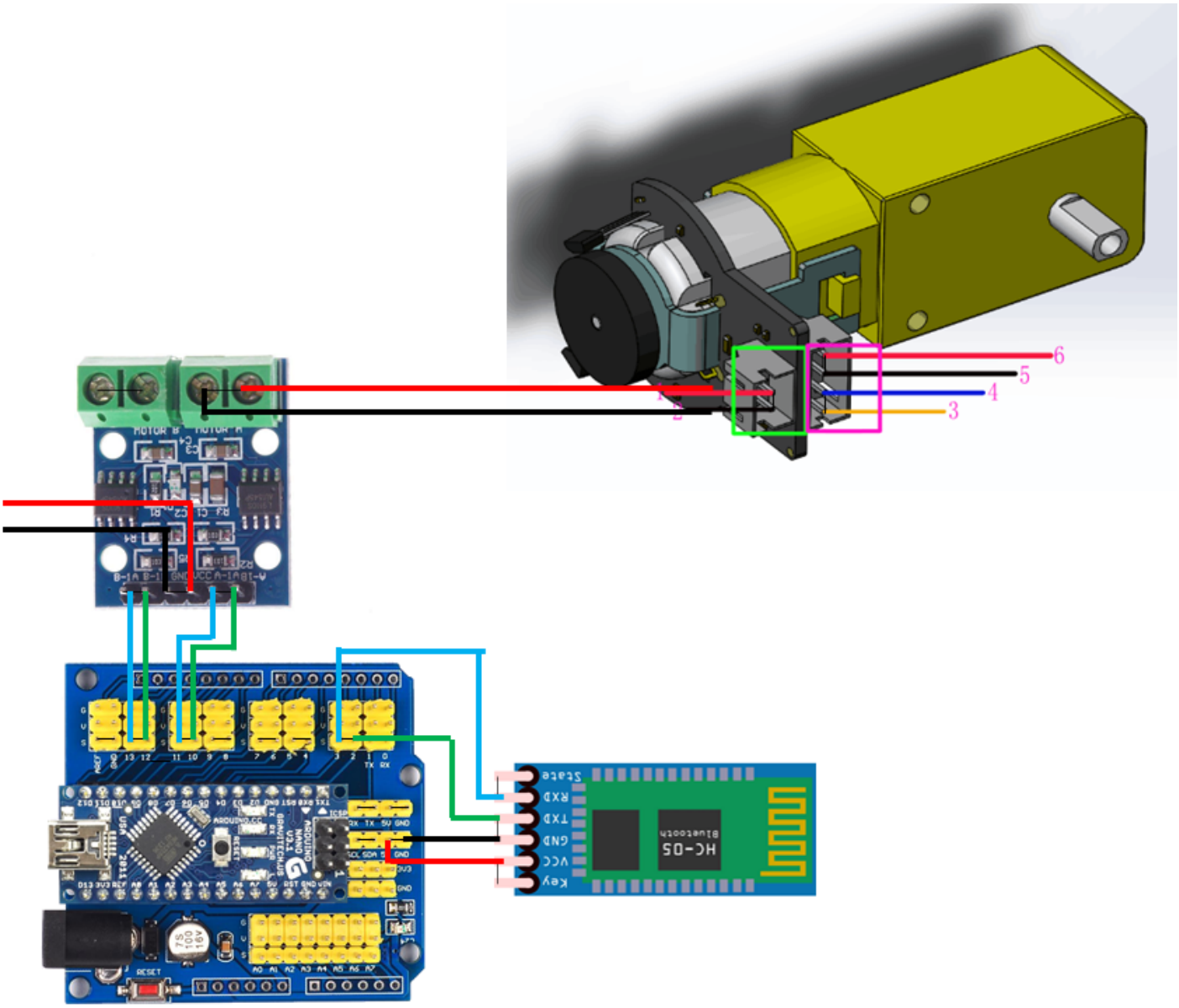
```
  Serial.println("Le HC-05 doit clignoter rapidement avant d'être appairé");
}
else{
  Serial.println("La vitesse de communication (baudrate) a été personnalisée");
}

delay(500);
// Commandes AT pour le HC-05
// BTSerie.print("AT+NAME?");  //Demande le nom du module. Noter le ?
// BTSerie.print("AT+NAME=RCcar-HC-05-1");  //Définir le nom du module.
BTSerie.print("AT+VERSION?");  //Demande le N° de version. Noter le ?
// BTSerie.print("AT+UART?");  //Demande la vitesse série (baudrate). Noter le ?
// BTSerie.print("AT+UART=57600,0,0");  //Définir la vitesse série (baudrate).
// BTSerie.print("AT+ROLE?");  //Demande le mode du module, maitre ou esclave. Noter le ?
// BTSerie.print("AT+PSWD?");  //Demande le mot-de-passe du module. Noter le ?
// La console série de l'ordinateur d'où l'on envoie les commandes AT doit être réglée de
telle sorte que
// les fins de ligne soient « les deux, NL et CR », ce qui revient à envoyer \r\n à la fin de
chaque commande.
BTSerie.print("\r\n");     // sur HC-05, toutes les commandes doivent se terminer par \r\n
// afficher ce que le module bluetooth répond
  Serial.print( BTSerie.read() );     // afficher sur la console ce qui est lu sur BT
// pour AT+VERSION?, c'est le n° de version puis OK qui s'affiche

void loop(){
  //On lit ce qui est envoyé à l'Arduino depuis la console via la liaison Serial
  readSerialPort();
  //Et on l'Arduino l'envoie au HC-05 via la liaison SoftwareSerial
  if(phraseTexte!="") BTSerie.println(phraseTexte);
  //L'Arduino lit ce que le HC-05 envoie via la liaison SoftwareSerial et l'envoie vers la
console
  if (BTSerie.available()>0){
    Serial.write(BTSerie.read());
  }
}
void readSerialPort(){
  phraseTexte="";
  while (Serial.available()) {
    delay(10);
```

```
    if (Serial.available() >0) {
      caractereTexte = Serial.read(); //le port série envoie des caractères de texte octet par
octet (byte from serial buffer)
      phraseTexte += caractereTexte; //construction d'une phrase en concaténant les caractères
reçus
    }
 }
}
```

Pilotage d'un moteur CC via bluetooth



```
//Voiture modélisée radiocommandée (RC car) avec deux moteurs CC à l'arrière et un servomoteur de direction

#include <Servo.h>
Servo myservo;  // create servo object to control a servo

//Le port série matériel de l'Arduino Nano (Pins 0/RX et 1/TX) est déjà utilisé pour la liaison Arduino-USB avec l'ordinateur
//On utilise donc une liaison série logicielle pour la liaison Arduino-HC05 avec le module Bluetooth
#include <SoftwareSerial.h>  //Software Serial Port
#define RxDpin 2    //Pin Digital 2 pour arduino Rx (pin0=serial)
#define TxDpin 3    //Pin Digital 3 pour arduino Tx (pin1)
//Pour la liaison SoftwareSerial Arduino-HC05, quand on veut configurer le HC-05
//on maintient le bouton à côté de la PIN EN/KEY enfoncé au démarrage de l'Arduino
// #define baudrate 38400 //Vitesse pour la liaion Arduino-HC05 en mode configuration
//Pour la liaison SoftwareSerial Arduino-HC05, quand on veut communiquer à travers le bluetooth depuis un smartphone,
//on appaire le HC-05 depuis l'appli avec le mot-de-passe par défaut : 1234
#define baudrate 9600 //Vitesse pour la liaion Arduino-HC05 en mode utilisation smartphone
SoftwareSerial BTSerie(RxDpin,TxDpin);

char caractereTexte;
String phraseTexte;

void setup()

{
Serial.begin(9600); //Vitesse (baudRate) pour la liaison Arduino-USB
delay(500);

//Configuration de la liaison SoftwareSerial avec le HC-05
pinMode(RxDpin, INPUT);   //Configuration du Pin RxD (Receive) en mode entrée
pinMode(TxDpin, OUTPUT);  //Configuration du Pin TxD (Transmit) en mode sortie
BTSerie.begin(baudrate);  //Vitesse pour la liaion Arduino-HC05
if (baudrate==38400) {
  Serial.println("En mode communication USB - Pret pour les commandes AT");
  Serial.println("Le HC-05 doit clignoter lentement (2 secondes)");
}
else if (baudrate==9600){
```

```
  Serial.println("En mode smartphone - Pret pour être appairé");
  Serial.println("Le HC-05 doit clignoter rapidement avant d'être appairé");
}
else{
  Serial.println("La vitesse de communication (baudrate) a été personnalisée");
}

delay(500);

pinMode(13,OUTPUT);   //left motors forward
pinMode(12,OUTPUT);   //left motors reverse
pinMode(11,OUTPUT);   //right motors forward
pinMode(10,OUTPUT);   //right motors reverse
pinMode(9,OUTPUT);   //Led
pinMode(5,OUTPUT);   //SG90 steering motor
myservo.attach(5);  // attaches the servo on pin 5 to the servo object
}

void loop() {
  // Serial.write(blueToothSerial.read());

  //On lit caractere par caractere sur la liaion Arduino-USB et on affiche sur la liaison
Arduino-HC05
  if (Serial.available()) {
    caractereTexte = Serial.read();
    BTSerie.write(caractereTexte);
    // Serial.println("Caractere envoye vers bluetooth : ");
    // Serial.println(caractereRecu);
  }
  //On lit caractere par caractere sur la liaion Arduino-HC05 et on affiche sur la liaison
Arduino-USB
  if (BTSerie.available()) {
    caractereTexte = BTSerie.read();
    Serial.print(caractereTexte);
  // }

if(caractereTexte == 'F'){            //move forward(all motors rotate in forward direction)
  digitalWrite(10,LOW);
  digitalWrite(12,LOW);
  digitalWrite(13,HIGH);
```

```
  digitalWrite(11,HIGH);
  // myservo.write(90);                  // sets the servo position according to the scaled
value
}

else if(caractereTexte == 'B'){      //move reverse (all motors rotate in reverse direction)
  digitalWrite(13,LOW);
  digitalWrite(11,LOW);
  digitalWrite(12,HIGH);
  digitalWrite(10,HIGH);
  // myservo.write(90);                  // sets the servo position according to the scaled
value
}

else if(caractereTexte == 'L'){      //turn right (left side motors rotate in forward
direction, right side motors doesn'caractereRecu rotate)
  digitalWrite(10,LOW);
  digitalWrite(12,LOW);
  digitalWrite(13,LOW);
  digitalWrite(11,HIGH);
  // myservo.write(60);                  // sets the servo position according to the scaled
value
}

else if(caractereTexte == 'R'){      //turn left (right side motors rotate in forward
direction, left side motors doesn'caractereRecu rotate)
  digitalWrite(10,LOW);
  digitalWrite(12,LOW);
  digitalWrite(11,LOW);
  digitalWrite(13,HIGH);
  // myservo.write(120);                  // sets the servo position according to the scaled
value
}


else if(caractereTexte == 'S'){      //STOP (all motors stop)
  digitalWrite(13,LOW);
  digitalWrite(12,LOW);
  digitalWrite(11,LOW);
  digitalWrite(10,LOW);
```

```
  // myservo.write(90);
}
delay(100);
  }
}
```

Depuis le clavier de PC

voir : https://innovation.iha.unistra.fr/books/robotique-educative/page/rc-car-voiture-modelisme-radiocommandee#bkmrk-depuis-le-moniteur-s

Depuis une application bluetooth de smartphone Android

- Ajouter le HC-05 depuis les paramètres Bluetooth Android
- Rentrer le code PIN `1234`
- Démarrer l'Application Bluetooth
- Appuyer sur les flèches / le joystick
- Si les flèches ne fonctionnent pas c'est qu'elles n'envoient pas les bons caractères
- Vous pouvez envoyer directement les caractères via un terminal série d'une Application Bluetooth

Voici les caractères Android :

- F pour Forward : Avancer
- B pour Backward : Reculer
- R pour Right : Aller à droite
- L pour Left : Aller à gauche
- S pour Stop : Arrêter

Voici le fonctionnement du joystick de l'application E&E: Arduino Automation par Engineers & Electronics

- Quand on appuie sur la flèche du haut, le caractère `F` (objet `char`) est envoyé par le téléphone au HC-05 puis à l'Arduino via la liaison série Software Arduino-HC05 qui est configurée par défaut en baudrate 9600 (vérifiable en envoyant la commande `BTSerie.print("AT+UART?");` au HC-05)
- Quand on relâche le bouton, le caractère `S` est envoyé

L'application E&E: Arduino Automation par Engineers & Electronics est testée ne semble pas fonctionner avec les Android récents ni iOS :

- https://play.google.com/store/apps/details?id=com.himanshu.ArduinoAutomation
- Chercher une application alternative en tapant les mots-clé bluetooth arduino

- Autres applications possibles : Carino, bluetooth2, BLE RobotCar, Arduino Bluetooth Controller (giumig)

Sources

https://knowledge.parcours-performance.com/arduino-bluetooth-hc-05-hc-06/

Revision #1
Created 16 October 2026 05:58:09 by Gaëtan Carron
Updated 16 October 2026 05:58:09 by Gaëtan Carron