

PROJET DE TELECOMMANDES VIA LE RESEAU GSM PAR ENVOI DE SMS

Est présenté ici le principe de gestion d'un module GPRS par un Arduino Nano -ou- Uno- en vue de contrôler à distance des équipements: commander des actions, acquérir des informations (état, mesures). Ceci à faible coût.

LES MOYENS :

- **Smartphone** : envoi des SMS de télécommande et réception de l'état du système télécommandé
- **Arduino Nano** : recevoir les commandes et activer des sorties. Acquérir aussi des entrées
- **Module GPRS SIM800** : assurer la réception et l'envoi des SMS
- **Carte SIM** (abonnement opérateur)

Le SIM800 et l'Arduino communiquent par liaison série avec des commandes AT.

Exemple de prix

High-Tech > Téléphones portables et accessoires > Accessoires téléphones portables > Autocollants



Gprs Sim800l SIM GSM Quadribande Carte Embarquée Ttl Avec Antenne Pour Arduino
de MagiDeal

★★★★★ 2 commentaires client

Prix: EUR 12,78

Nouveau Prix: **EUR 7,99 LIVRAISON GRATUITE en France métropolitaine.**
Économisez: EUR 4,79 (37%)
Tous les prix incluent la TVA.

En stock.

Faites-vous livrer 11 et 21 déc. en choisissant la Livraison Rapide lors du passage de commande. En savoir plus.

Expédié et vendu par Passionné Capricorne.

1 neuf à partir de EUR 7,99

- SIM800L GPRS GSM Module avec câble d'antenne
 - Quadri-bande 850/900/1800 / 1900MHz
 - GPRS multi-fente Class12 connectivité: max. 85.6kbps (down-load / up-load)
 - GPRS station mobile de classe B
- » Voir plus de détails

Comparer avec des articles similaires

Nos prix incluent l'éco-participation sur tous les produits concernés. Vous voulez recycler votre appareil électrique ou électronique gratuitement ? En savoir plus ici.

SODIAL(R) Nano V3.0 AVR ATmega328 P-20AU Carte Module Noir Bleu w USB Cable

de SODIAL(R)

★★★★★ 103 commentaires client | 18 questions ayant reçu une réponse

Prix: **EUR 3,49 LIVRAISON GRATUITE en France métropolitaine.**
Tous les prix incluent la TVA.

En stock.

Faites-vous livrer 11 et 21 déc. en choisissant la Livraison Rapide lors du passage de commande. En savoir plus.

Expédié et vendu par Sonline-HK.

4 neufs à partir de EUR 3,49

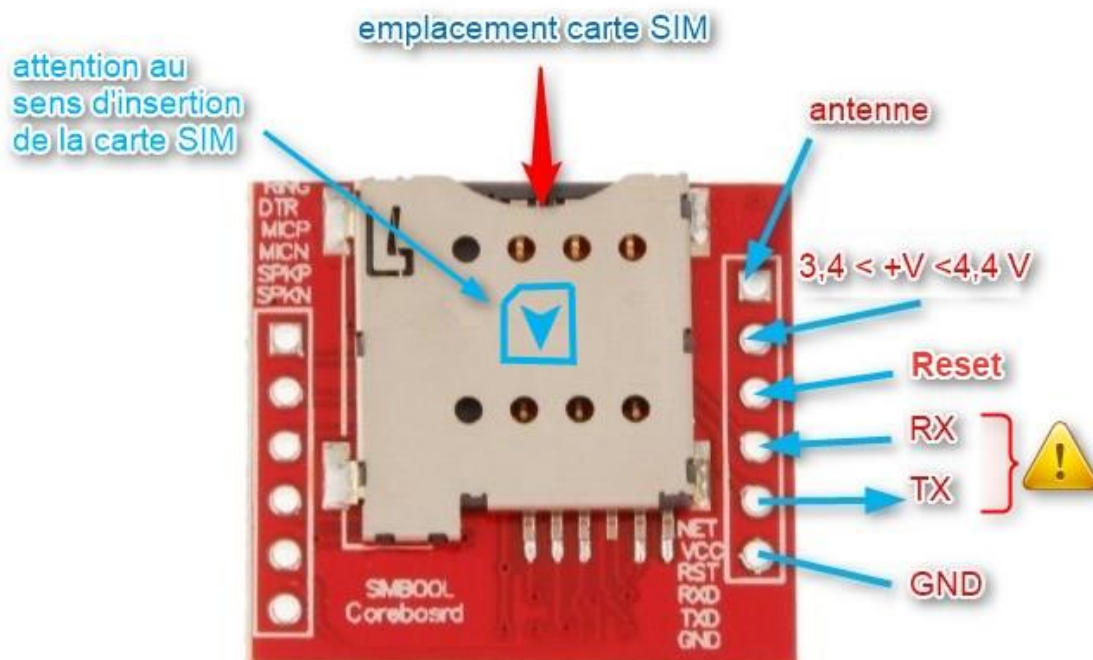
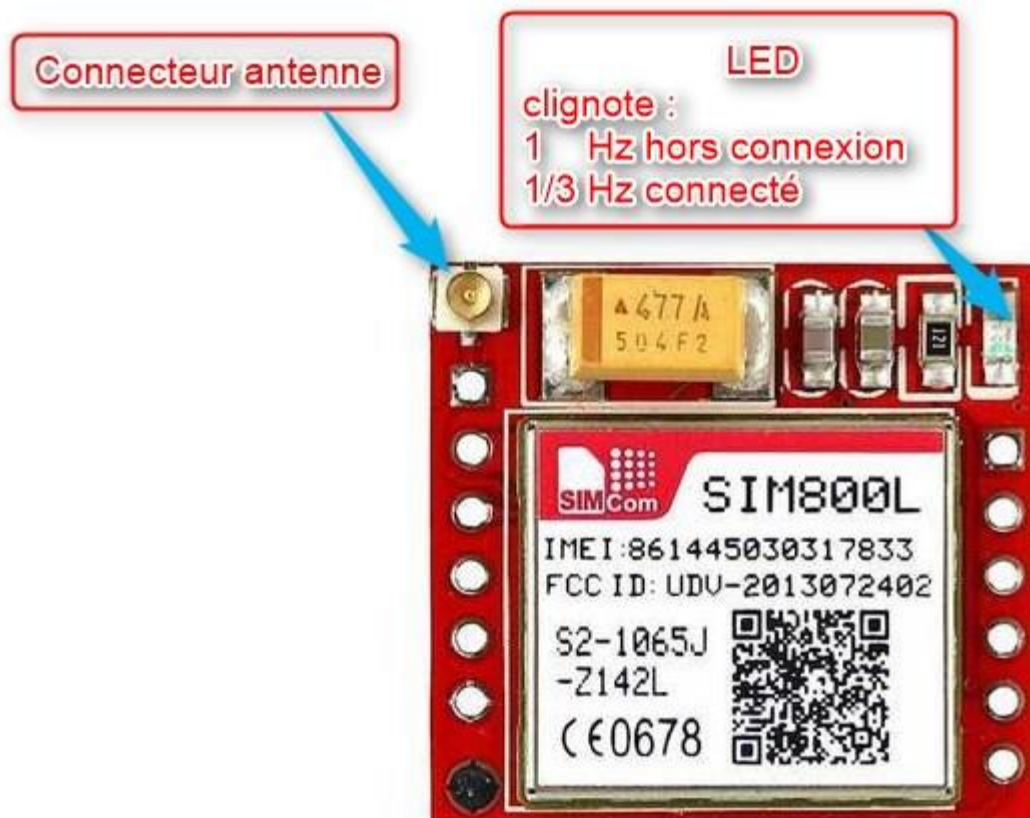
- Nom de l'article: Nano V3.0 AVR ATmega328 P-20AU Conseil Module w Cable USB pour Arduino; E / S numeriques Pins: 12 (D2 ~ D13); broches d'entree analog (A0 ~ A7)
 - Pins PWM de sortie: 6 (D3 / D5 / D6 / D9 / D10 / D11); Microcontrôleur: Atmel ATmega328P-UA; Alimentation: 5V ~ 12V DC & pile 9V
 - Periodicite: 16MHz; Taille: 5,8 x 3,4 x 1,7cm / 2,3" x 1,3" x 0,7" (L * W * T)
 - Materiel: Carte électronique; Carte Couleur: Bleu, Noir
 - Poids total: 23g; Contenu du package: 1 x Nano ATmega328 P-20AU Module Carte 1 x USB Cable(32cm)
- » Voir plus de détails

Comparer avec des articles similaires

Nos prix incluent l'éco-participation sur tous les produits concernés. Vous voulez recycler votre appareil électrique ou électronique gratuitement ? En savoir plus ici.



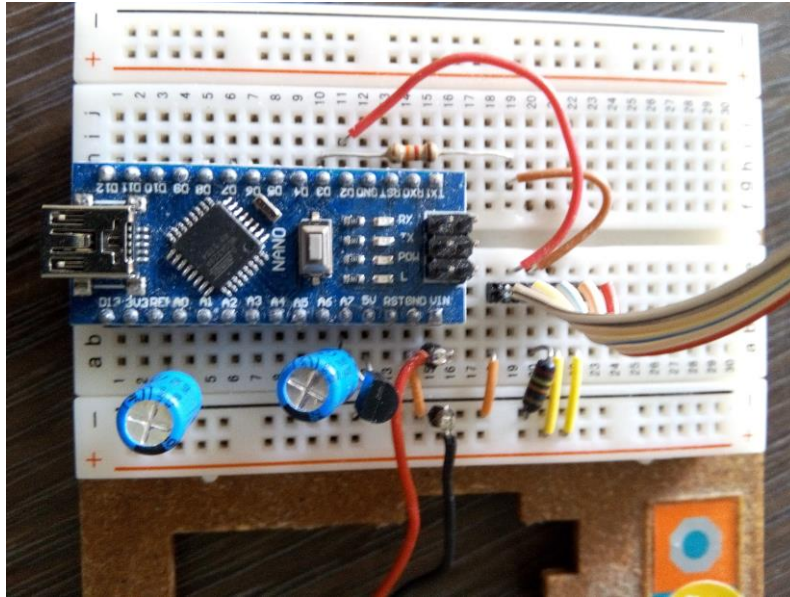
MODULE SIM800



Attention !
Sur certaines cartes (couleur verte) TX et RX sont inversés

MONTAGE D'ESSAI

Arduino Nano



D2 connectée à TX du SIM800

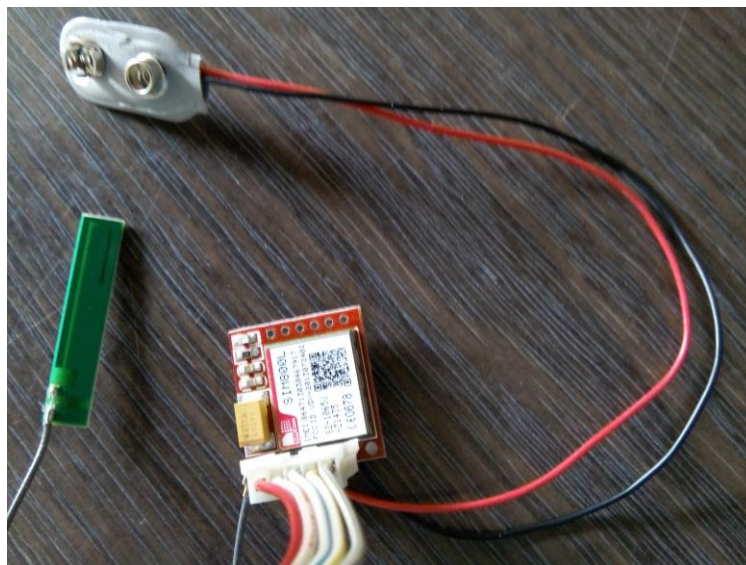
D3 connectée à RX du SIM800 via un pont diviseur pour ne pas dépasser une tension de 3,3 V.

D0 et D1 réservées au port série IDE de l'Arduino.

Disponibles pour les télécommandes :

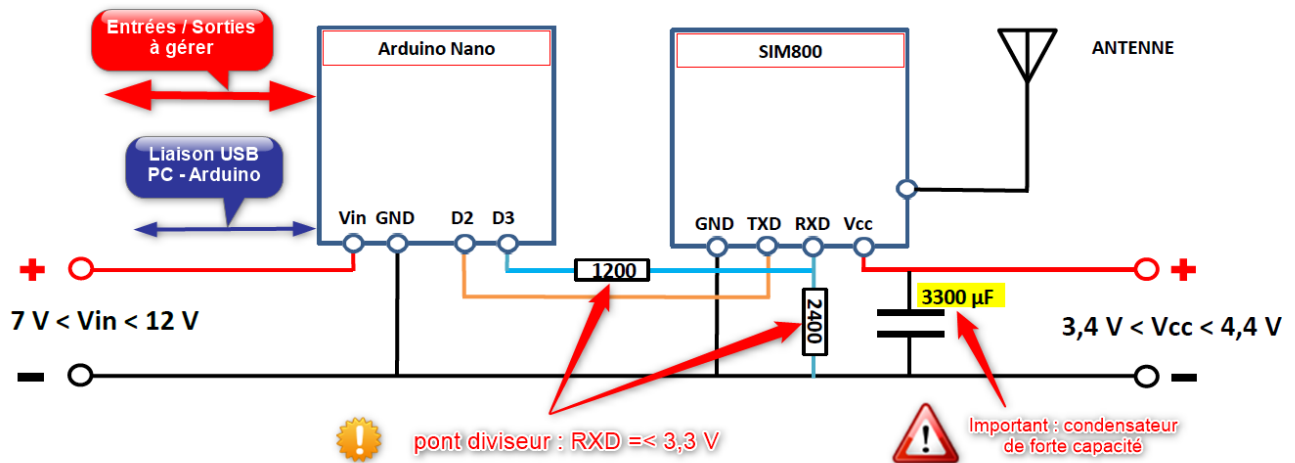
- D4 à D12 pour des E/S TOR
- A0 à A5 pour des E/S TOR ou entrées analogiques
- A6 et A7 pour des entrées analogiques (uniquement Arduino Nano)

SIM800



Alimentation entre 3,4 V et 4,4 V au plus direct du module SIM800 pouvant fournir au moins 2 A. Par exemple 3 batteries NI-MH en série de 2000 mAh (voir remarque ci-dessous).

SCHEMA DE PRINCIPE



Remarque : On apportera un soin particulier à l'alimentation du SIM800. Les pointes de courant peuvent créer de brèves chutes de tension qui, si elles sont inférieures à la tension minimale, entraînent le « reset » du module.

LES COMMANDES AT

Pour paramétrer ce module et **envoyer/recevoir des SMS**, on utilisera les commandes AT suivantes :

1- Vérifier la connexion avec le module :

Commande : **AT**
Réponse : **OK**

2- Envoyer le code PIN si nécessaire (carte SIM initialisée avec un code) :

Commande : **at+cpin="XXXX"** // XXXX code PIN de la carte SIM
Réponse : **OK**
+CPIN: READY

Call Ready

SMS Ready

3- Activer le mode texte (ASCII) :

Commande : **AT+CMGF=1**
Réponse : **OK**

Permet d'activer le mode texte (ASCII) pour l'envoi du SMS.

4- Envoi d'un SMS :

⇒ *Se fait en 3 étapes.*

4.1- Passer le n° de téléphone :

Commande : `AT+CMGS="+336xxxxxxxx"` // n° avec préfixe du pays
Réponse : `>` // C'est l'invite à saisir le message

4.2- Saisie du message :

Commande : `> ligne 1` // Chaque ligne se termine par <CR>
`> ligne 2` // touche « Entrée » avec un terminal
`> ligne 3`
`> ligne 4`
`>`

4.3- Envoi du message :

Commande : `<CTRL-Z>` // voir ci-dessous le programme de test
Réponse : `+CMGS: X` // X numéro du SMS envoyé
`OK`

5- Réception d'un SMS :

5.1- Avertissement de réception d'un SMS vers l'Arduino :

Réception : `+CMTI: "SM",X` // X numéro du SMS reçu

5.2- Récupération du SMS :

Commande : `AT+CMGR=X` // X numéro du SMS
Réponse :
`+CMGR: "REC UNREAD","+336xxxxxxxx","","17/09/23,16:08:01+08"`
`ligne 1`
`ligne 2`
`ligne 3`
`ligne 4`

`OK`

6- Autres commandes utiles :

- Effacer les SMS reçus de la mémoire du module SIM800

Commande : **AT+CMGD=1,4**

Réponse : **OK**

- Rapport de qualité du réseau

Commande : **AT+CSQ**

Réponse : **+CSQ:<rss>,<ber>** // rssi : niveau du signal
// ber : taux d'erreur (en %)

OK

- Liste des opérateurs réseau accessibles

Commande : **at+cops=?**

Réponse : **+COPS: (2,"SFR","SFR","20810"),(1,"BOUYGUES
TELECOM","BYTEL","20820"),(1,"Orange F","Orange","20801"),,(0-
4),(0-2)**

OK

- Nom de l'opérateur connecté

Commande : **at+cops?**

Réponse : **+COPS: 0,0,"Orange F"**

OK

PROGRAMME DE TEST (ARDUINO UNO OU NANO)

Permet de simuler un terminal :

Liaison série entre le moniteur de l'IDE Arduino et le module SIM800

Le code :

```
#include <SoftwareSerial.h>
```



disposer d'un 2^e port série

```
// TX du SIM800 connecté à D2 de l'Arduino  
#define SIM800_TX_PIN 2
```

```
// RX du SIM800 connecté à D3 de l'Arduino  
#define SIM800_RX_PIN 3
```

```
//Création d'une liaison série logicielle avec SIM800  
SoftwareSerial gsm(SIM800_TX_PIN, SIM800_RX_PIN);
```

```
void setup() {  
  
  Serial.begin(9600);  
  while(!Serial);  
  
  //initialisation liaison série Arduino <-> SIM800  
  gsm.begin(9600);  
  delay(1000);  
  
  Serial.println("Setup terminé !");  
  
}
```

```
void loop() {  
  
  //Lire l'envoi du SIM800 (si disponible) et transmettre au  
  Moniteur série  
  if(gsm.available()){  
    Serial.write(gsm.read());  
  }  
  
  //Lire l'envoi du Moniteur série (si disponible) et transmettre au  
  SIM800  
  if(Serial.available()){  
    char car = Serial.read();  
    if (car==126) car= char(26);    // signe ~ => simule Ctrl-z qui  
    n'est pas possible sur le moniteur série  
    gsm.write(car);  
    //gsm.write(Serial.read());  
  }  
  
}
```