

Ce site utilise des cookies provenant de Google afin de fournir ses services, personnaliser les annonces et analyser le trafic. Les informations relatives à votre utilisation du site sont partagées avec Google. En acceptant ce site, vous acceptez l'utilisation des cookies.

EN SAVOIR PLUS OK !



Électronique, Arduino, Raspberry Pi, robotique, expériences scientifiques, MIDI, amplificateurs, effets pour guitare, microcontrôleurs, DIY...pour le plaisir!

Accueil Arduino Raspberry Pi Autres µcontrôleurs Musique Robots + moteurs Divers Contact



lundi 5 mai 2014

Communication par USB entre Raspberry Pi et Arduino

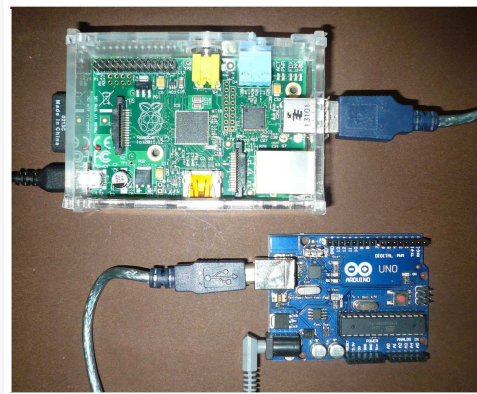
Pourquoi voudrait-on établir une communication entre un Raspberry Pi et un Arduino? Maintenant que [nous avons appris comment utiliser les pins GPIO du Raspberry Pi](#), ce dernier n'est-il pas capable d'agir de façon autonome?

Voici quelques raisons possibles:

- Le Raspberry Pi n'est pas idéal pour accomplir des actions avec un timing précis, car Linux pourrait décider de se consacrer temporairement à une autre tâche. Par exemple, si vous contrôlez un moteur pas à pas avec le Raspberry Pi, le moteur pourrait s'arrêter momentanément, de temps à autres. L'Arduino est plus fiable dans ce genre de situation.
- Vos capteurs peuvent se trouver sur un shield conçu pour être inséré sur un Arduino. Évidemment, la disposition des pins GPIO du Raspberry Pi n'est pas la même.
- Si vous êtes déjà un utilisateur expérimenté de l'Arduino mais un néophyte en Python/Linux/Raspberry Pi, il y a certaines choses que vous savez déjà faire très facilement avec l'Arduino, mais que vous ne savez pas encore comment faire avec le Raspberry Pi.
- Vous désirez établir un réseau de plusieurs capteurs satellites qui se rapportent à une seule station centrale, et plusieurs Arduinos coûtent moins cher que plusieurs Raspberry Pi.

La bonne nouvelle, c'est que c'est tout à fait possible, et même relativement facile, d'établir une communication entre un Raspberry Pi et un Arduino. Et ce ne sont pas les options qui manquent: communication série (UART), I2C, Ethernet, communication sans fil...

Dans cet article, nous abordons l'option qui s'avère probablement la plus simple: la communication série par l'entremise d'un câble USB: vous allez simplement brancher l'Arduino à un des deux ports USB du Raspberry Pi, de la même façon que vous le faites lorsque vous désirez programmer l'Arduino avec un ordinateur conventionnel. Mais attention: avant de brancher l'Arduino au Raspberry Pi, assurez-vous que l'Arduino dispose de sa propre alimentation; le Raspberry Pi pourrait avoir des ratés si sa propre alimentation devait, en plus, alimenter l'Arduino par le port USB.



Archives du blog

- 2016 (20)
- 2015 (58)
- ▼ 2014 (57)
 - décembre (4)
 - novembre (6)
 - octobre (6)
 - septembre (6)
 - août (3)
 - juillet (5)
 - juin (2)
 - ▼ mai (6)
 - Film en accéléré (time lapse) avec un vieil appare...
 - Fabrication d'un métronome
 - Identifieur d'accords MIDI (Arduino)
 - Afficheur LCD 16 X 2 et Raspberry Pi
 - Communication par USB entre Raspberry Pi et Arduin...
 - Construction d'un petit robot chercheur de lumière...
- avril (4)
- mars (7)
- février (2)
- janvier (6)

- 2013 (56)
- 2012 (52)
- 2011 (26)
- 2010 (6)

Ce site utilise des cookies provenant de Google afin de fournir ses services, personnaliser les annonces et analyser le trafic. Les informations relatives à votre utilisation du site sont partagées avec Google. En acceptant ce site, vous acceptez l'utilisation des cookies.

EN SAVOIR PLUS OK !



Arduino et son alimentation



Installation de la bibliothèque pyserial

Installons d'abord la bibliothèque *pyserial* dans le Raspberry Pi: elle nous facilitera grandement la tâche lors de la conception de nos scripts en Python. Pour ce faire, vous utilisez cette commande pendant que le Raspberry Pi est branché à l'internet:

```
sudo apt-get install python-serial
```

Envoi d'un message de l'Arduino vers le Raspberry Pi

Commençons par une communication unidirectionnelle de l'Arduino vers le Raspberry Pi. Du côté Arduino, l'envoi de messages se fait par la commande "Serial.println". Par exemple, le sketch ci-dessous envoie un message toutes les 3 secondes. J'ai ajouté un compteur pour que le message ne soit pas rigoureusement identique d'une fois à l'autre.

Nous pourrions donc vérifier que tous les messages envoyés par l'Arduino sont bel et bien captés par le Raspberry Pi.

```
1  /*****
2  Toutes les 3 secondes, l'Arduino envoie un message qui
3  sera éventuellement reçu par un Raspberry pi.
4  *****/
5
6  int compteur = 0;
7
8  void setup(){
9      Serial.begin(9600);
10 }
11
12 void loop(){
13     Serial.print("Message numero ");
14     Serial.println(compteur);
15     Serial.println("Bonjour, la Framboise, ici l'Arduino!");
16     compteur++;
17     delay(3000);
18 }
19 }
```

ArduinoVersPiUSB.ino hosted with ♥ by GitHub

[view raw](#)

Comme vous pouvez le constater, c'est exactement comme si vous vouliez afficher un message sur le moniteur série de l'environnement de programmation Arduino; vous pouvez d'ailleurs utiliser le moniteur série pour vérifier dès maintenant que le sketch fonctionne de la façon prévue.



Yves Pelletier

Pages

- [Accueil](#)
- [Divers](#)
- [Arduino](#)
- [Raspberry Pi](#)
- [Autres µcontrôleurs](#)
- [Robots + moteurs](#)
- [Musique](#)
- [Contact](#)

Messages les plus consultés



Une horloge pour votre Arduino (Real Time Clock)

Il est parfois utile que votre Arduino connaisse la date et l'heure. Par exemple, l'Arduino pourrait être à la base d'un syst...



Communication par USB entre Raspberry Pi et Arduino

Pourquoi voudrait-on établir une communication entre un Raspberry Pi et un Arduino? Maintenant que nous avons appris comment utiliser les ...



Modules RF 433 MHz, VirtualWire et Arduino

Aujourd'hui, je vous présente quelques expérimentations réalisées avec un émetteur et un récepteur radiofréquence à 433 MHz obtenus sur...



Contrôler un moteur DC en python avec le Raspberry Pi

Voilà un certain temps que je n'ai pas joué avec mon Raspberry Pi...pourquoi ne pas l'utiliser pour contrôler un (ou deux) moteur(...



Module WiFi ESP8266: test et configuration avec Arduino Uno

L'ESP8266 est un circuit intégré qui vous permet, à très faible coût, de sauter à pieds joints dans le domaine de l'internet des obj...



Transformation d'un vieux écran d'ordinateur portable en moniteur HDMI-VGA-DVI

D'une part, j'avais encore ce vieux ordinateur portable (un Powerbook G3 Wallstreet) qui avait cessé de

Ce site utilise des cookies provenant de Google afin de fournir ses services, personnaliser les annonces et analyser le trafic. Les informations relatives à votre utilisation du site sont partagées avec Google. En acceptant ce site, vous acceptez l'utilisation des cookies.

EN SAVOIR PLUS OK !

ls /dev/tty*

Vous devriez avoir un résultat légèrement différent selon que l'Arduino est connecté ou non au Raspberry Pi. Le nom correct devrait être celui qui apparaît dans la liste pendant que l'Arduino est branché, mais qui n'y apparaît pas quand l'Arduino est débranché du port USB.

Voici donc un sketch en Python qui permettra au Raspberry Pi d'écouter ce que lui dit l'Arduino, et d'afficher le résultat. (Personnellement, j'utilise le logiciel WinSCP pour installer le fichier dans le Raspberry Pi).

```
1 #!/usr/bin/env python
2 # -*- coding: latin-1 -*-
3
4 import serial
5 ser = serial.Serial('/dev/ttyACM0', 9600)
6 while 1 :
7     print(ser.readline())
```

ArduinoVersPiUSP.py hosted with ♥ by GitHub

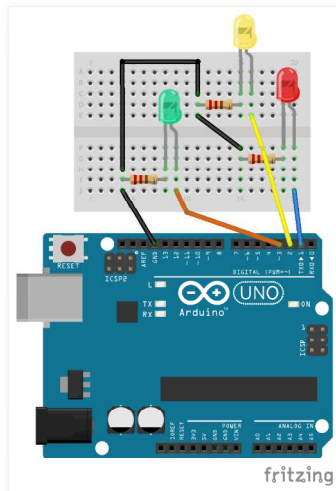
[view raw](#)

Pour interrompre le script, vous pouvez utiliser la combinaison Ctrl-C au clavier.

Envoi d'un message du Raspberry Pi vers l'Arduino

Essayons maintenant l'autre direction: branchons quelques LEDs à l'Arduino, et c'est le Raspberry Pi qui indiquera à l'Arduino quelles LEDs il doit éteindre ou allumer.

Donc, une LED (accompagnée de sa résistance protectrice) est branchée à chacune des pins suivantes de l'Arduino: 2, 3 et 4 (*attention: le schéma ci-dessous montre par erreur des connexions aux pins 1, 2, et 3!*).



Commençons par le sketch Arduino. Grâce à la fonction Serial.read, le sketch ci-dessous allume ou éteint les LEDs en fonction du message reçu: il allume la LED 1 (pin 2) s'il reçoit le message "1", il éteint la LED 2 (pin 3) s'il reçoit le message "5", etc.

```
1 /*****
2 3 LEDs sont reliées à l'Arduino (aux pins 2, 3 et 4). Les LEDs s'allument
3 en fonction du message reçu.
4 *****/
5
6 #define LED1 2
7 #define LED2 3
8 #define LED3 4
9
10 int message = 0;
```

propre
télécommande de téléviseur au
moyen d'un Arduino? Bonne idée!
Vous pourriez, par exemple, réunir
sur une même...



[Transmettre les données de l'Arduino vers un tableur \(Excel ou Libre Office Calc\)](#)

Ce n'est pas la première fois que je vous parle de data logging au moyen de l'Arduino: il y a quelques mois, nous avons vu comment ...



[Communication RF 433 MHz entre Raspberry Pi et Arduino](#)

Nous avons déjà abordé la communication entre le Raspberry Pi et l'Arduino au moyen d'un câble USB, ce qui imposait une certaine pr...



[Mesurer une température avec un Arduino](#)

Le projet est simple: construire un thermomètre au moyen d'un Arduino, d'un capteur de température et d'un afficheur à cristaux...

S'abonner à

[Articles](#)

[Commentaires](#)

Soyez avisés des nouveaux posts par email

Email address...

Submit

Libellés

433 MHz 74HC125 74HC595
Accéléromètre ADC Afficheur Afficheur
LCD Alimentation cc Amplificateur Audio
Amplificateur Opérationnel
Arduino ATTiny Audio Bascule de
Schmitt Bluetooth Capteur Carte SD
CD40106 CD4040 CD4051 Circuit numérique
Circuits imprimés Clavier Clavier
numérique; Comparateur Condensateurs
Cours en ligne Data logger Diode
domotique DS1307 DS18B20 Effets pour
guitare Électronique
conventionnelle Émetteur FM
Energia Engrenages ESP8266 Ethernet
Shield Étude de CI Expérience
Expérience scientifique Firmata HC-SR04
Horloge I2C Infrarouge Interrupteur Laser
LED LED infrarouge LEDs Livre LM317
Logiciels Magnétisme Matériel Matrice
Microchip Microcontrôleurs Microphone
MIDI Minuterie 555 Moteur cc Moteur
pas à pas mp3 MPLAB MSP430
Launchpad Musique NRF24L01
Nucleo Optocoupleur Orange Pi Oscillateur
Oscilloscope Outils Web Photographie
Photorésistance Phototransistor

Ce site utilise des cookies provenant de Google afin de fournir ses services, personnaliser les annonces et analyser le trafic. Les informations relatives à votre utilisation du site sont partagées avec Google. En acceptant ce site, vous acceptez l'utilisation des cookies.

EN SAVOIR PLUS OK !

```

16  pinMode(LED2, OUTPUT);
17  pinMode(LED3, OUTPUT);
18
19  Serial.begin(9600);
20  }
21
22  void loop()
23  {
24      if (Serial.available()) {
25          message = Serial.read()-'0'; // on soustrait le caractère 0, qui va
26
27          switch (message) {
28              case 1:
29                  digitalWrite(LED1, HIGH);
30                  break;
31              case 2:
32                  digitalWrite(LED2, HIGH);
33                  break;
34              case 3:
35                  digitalWrite(LED3, HIGH);
36                  break;
37              case 4:
38                  digitalWrite(LED1, LOW);
39                  break;
40              case 5:
41                  digitalWrite(LED2, LOW);
42                  break;
43              case 6:
44                  digitalWrite(LED3, LOW);
45                  break;
46          }
47      }
48  }

```

PiVersArduinoUSB.ino hosted with ♥ by GitHub

[view raw](#)

Pourquoi soustraire le caractère '0'? C'est parce qu'au départ, lorsque l'Arduino reçoit le message "1", il s'agit d'un caractère dont le code ASCII est 49. En soustrayant le caractère '0' (dont le code ASCII est 48), le caractère '1' qui a été reçu devient le nombre entier 1.

Si vous le désirez, vous pouvez vérifier que ce sketch fonctionne correctement à l'intérieur de l'environnement de programmation Arduino, avant même de brancher l'Arduino au Raspberry: dans le moniteur Série, tapez des nombres entre 1 et 6 et voyez que les LEDs s'allument ou s'éteignent en fonction des nombres que vous envoyez.

Et voici le script en Python qui sera exécuté par le Raspberry Pi: c'est maintenant lui qui envoie des informations à l'Arduino:

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  # Le Raspberry Pi envoie des messages à l'Arduino
5
6  import serial # bibliothèque permettant la communication série
7  import time   # pour le délai d'attente entre les messages
8
9  ser = serial.Serial('/dev/ttyACM0', 9600)
10  compteur = 0
11  while True:    # boucle répétée jusqu'à l'interruption du programme
12      if compteur < 6:
13          compteur = compteur + 1
14      else:
15          compteur = 0

```

[Stellans](#)
[Lauchpad](#)
[STM32](#)
[Nucleo](#)
[Stroboscope](#)
[Synthétiseur](#)
[Télécommande](#)
[Télémètre](#)
[Température](#)
[Transistor](#)
[Trouver des composants](#)
[TSOP4836](#)
[UART](#)
[Ultrasons](#)
[USB](#)
[VCO](#)
[VS1003](#)
[VS1053](#)
[XPress](#)

Facebook

Électronique en Amateur



Twitter

Tweets by @ElectroAmateur

Électro Amateur
 @ElectroAmateur

Fabrication d'un anémomètre avec #Arduino
electroniqueamateur.blogspot.ca/2016/06/fabric...



23 Jun

Électro Amateur
 @ElectroAmateur

Fabrication d'un récepteur radio

[Embed](#)

[View on Twitter](#)

Nombre total de pages vues

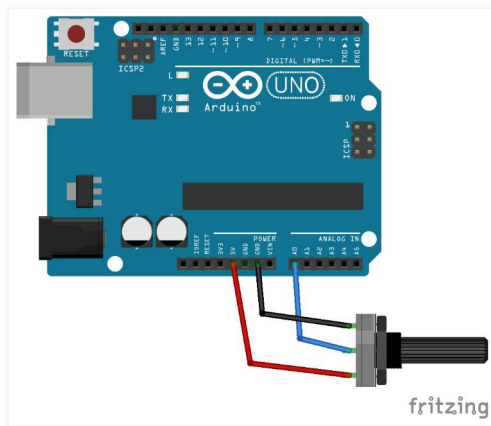
1270257

Ce site utilise des cookies provenant de Google afin de fournir ses services, personnaliser les annonces et analyser le trafic. Les informations relatives à votre utilisation du site sont partagées avec Google. En acceptant ce site, vous acceptez l'utilisation des cookies.

[EN SAVOIR PLUS](#) [OK !](#)

Communication bidirectionnelle entre le Raspberry Pi et l'Arduino

Finalement, voici un exemple où le Raspberry Pi demande une information à l'Arduino (l'état d'un potentiomètre dont le curseur est branché à A0), et affiche la réponse que lui a retournée l'Arduino.



Le sketch Arduino:

```

1  /*****
2   * L'Arduino attend la requête du Raspery Pi. Lorsqu'il reçoit cette r
3   * retour le signal mesuré à la pin A0.
4   *****/
5
6  int message = 0;
7
8
9  void setup()
10 {
11   Serial.begin(9600);
12 }
13
14 void loop()
15 {
16   if (Serial.available()) {
17     message = Serial.read()-'0'; // on soustrait le caractère 0, qui va
18
19     if (message == 5){
20       Serial.println(analogRead(A0),DEC);
21     }
22   }
23 }
24

```

BidirectionnelUSB.ino hosted with ❤ by GitHub

[view raw](#)

Le script en python du Raspberry Pi:

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  # Le Raspberry Pi demande une information à l'Arduino,
5  # puis il affiche la réponse à l'écran
6
7  import serial # bibliothèque permettant la communication série
8  import time   # pour le délai d'attente entre les messages
9

```

Ce site utilise des cookies provenant de Google afin de fournir ses services, personnaliser les annonces et analyser le trafic. Les informations relatives à votre utilisation du site sont partagées avec Google. En acceptant ce site, vous acceptez l'utilisation des cookies.

EN SAVOIR PLUS OK !

```
15 print(ser.readline()) #on affiche la reponse
16 time.sleep(1) # on attend pendant 1 seconde
```

BidirectionnelUSB.py hosted with ❤ by GitHub

[view raw](#)

Sources:

Mon point de départ a été la lecture de [cet article dans le blog d'Oscar Liang](#).

Yves Pelletier (Twitter: [@ElectroAmateur](#))

Publié par Yves Pelletier à 20:35



Recommander ce contenu sur Google

Libellés : [Arduino](#), [Python](#), [Raspberry Pi](#)

14 commentaires:

Anonyme 29 octobre 2014 à 12:09

Un grand merci pour ce tutorial !

[Répondre](#)

Anonyme 22 décembre 2014 à 07:40

merci pour l'ensemble de votre oeuvre que je viens de découvrir.

petit signalement: dans la partie TX_Pi Rx_Uno: le schéma montre un branchement sur les pins 1, 2 et 3 au lieu de 2,3 et 4.

bonne continuation!

[Répondre](#)

[Réponses](#)



Yves Pelletier

18 janvier 2015 à 13:27

Zut...

[Répondre](#)

Anonyme 13 mars 2015 à 08:35

Super tutorial mais, question bête, vous êtes sur quel OS ? Raspbian ?

[Répondre](#)

[Réponses](#)



Yves Pelletier

13 mars 2015 à 11:42

Oui, c'est Raspbian.

[Répondre](#)



Unknown 28 septembre 2015 à 04:56

Bonjour,

Le code dans ArduinoVersPiUSP.py montre une boucle infinie passant son temps à lire que quelque chose arrive sur le R Pi. Ca ne grève pas trop sa réactivité (si c'est aussi un serveur web par exemple) ? L'équivalent d'une interruption est-il possible ?

[Répondre](#)

[Réponses](#)



Unknown 20 octobre 2015 à 22:23

Sauf que readLine est un appel bloquant qui attend, tu peut le prouver toi même en affichant un compteur de boucle par exemple.

Ce site utilise des cookies provenant de Google afin de fournir ses services, personnaliser les annonces et analyser le trafic. Les informations relatives à votre utilisation du site sont partagées avec Google. En acceptant ce site, vous acceptez l'utilisation des cookies.

EN SAVOIR PLUS OK !

Ce commentaire a été supprimé par l'auteur.

Répondre

Automation Sense 18 décembre 2015 à 05:30

Intéressant le tuto

Répondre

Anonyme 1 janvier 2016 à 10:09

Bonjour,

J'ai un raspberry pi 2 sur lequel j'ai installé ubuntu mate 15.10.

J'ai aussi une arduino uno Rev 3.

J'ai des capteurs connectés à l'arduino. J'ai mis tout ce petit monde ensemble et ô joie le moniteur série de l'IDE Arduino m'indique bien les données des capteurs.

Les choses se compliquent ici : Je souhaite désormais installer opendaqcalc qui est une macro calc (le fameux tableur) pour pouvoir importer les données des capteurs arduino sur un tableur en temps réel. Le lien : <http://p.loussouarn.free.fr/arduino/arduino.html> (vers la fin de la page). Problème : lorsque je clique sur "connecter" : rien ne se passe ! J'ai quasiment tout essayé pour résoudre ce problème mais sans résultat. Selon vous, qu'est ce qui fait que le raspberry fait des siennes alors qu'avec mon vieux netbook windows xp j'obtiens les données sans problèmes sur calc? Merci par avance !!!

Répondre

Réponses



Yves Pelletier 1 janvier 2016 à 11:30

Il y a cet article dans lequel j'avais détaillé tous les réglages à faire pour Opendaqcalc (2e partie de l'article); je n'étais pas sur Raspberry Pi, mais l'OS était quand même Linux, ça vous aidera peut-être:

<http://electroniqueamateur.blogspot.ca/2014/10/transmettre-les-donnees-darduino-vers.html#more>

Anonyme 1 janvier 2016 à 13:40

Merci pour votre réponse. J'ai fait comme il fallait (sur RASPBIAN et sur UBUNTU MATE) J'ai essayé sur windows. Tout est ok. Je suis de plus en plus convaincu qu'il s'agit d'un problème de port lié spécifiquement au raspberry. Le plus surprenant est que le moniteur série d l'arduino fonctionne impeccablement bien et minicom (moniteur du pi) aussi. Le problème est donc soit dans le logiciel opendaqcalc qui n'est pas "conçu" pour le raspberry soit dans le raspberry lui-même.... j'ai envoyé un mail au concepteur de la macro mais celui-ci ne me répond plus...

Répondre



Unknown 29 avril 2016 à 11:25

Bonjour, ce tuto m'a été d'une très grande utilité.

Vous auriez une idée de comment envoyer les valeurs afficher par le fichier python vers un site web (déjà héberger sur mon rapsberry) ?

Merci d'avance :)

Répondre



Unknown 12 mai 2016 à 03:44

Bonjour je suis en terminale et pour mon projet dan lequel je dois faire fonctionner un tapis de course à l'aide d'un raspberry je me suis basé ton tuto, et voici mon problème : lorsque l'arduino envoie les données au rasp , le rasp les lis , tout va bien , mais lorsque le rasp envoie les données à l'arduino, un messaeg d'erreur apparait me disant qu'il y a un probleme avec le packages (bibliothèque du rasp) a la ligne 471 :dans le message d'erreur, à la fin: " n=os.write(self.fd,d)

TypeError:'str' does not support the buffer interface"

et je ne sais pas comment le résoudre pourriez vous m'aider svp ?

Répondre

Ce site utilise des cookies provenant de Google afin de fournir ses services, personnaliser les annonces et analyser le trafic. Les informations relatives à votre utilisation du site sont partagées avec Google. En acceptant ce site, vous acceptez l'utilisation des cookies.

EN SAVOIR PLUS OK !

Commentaire :

Bernard Mayer Fig

Déconnexion

Publier

Aperçu

☐ M'informer

Liens vers ce message

[Créer un lien](#)

[Article plus récent](#)[Accueil](#)[Article plus ancien](#)

Inscription à : [Publier les commentaires \(Atom\)](#)