ÉLECTRONIQUE en amateur

Des articles sur l'électronique, l'Arduino, le Raspberry Pi, la programmation de microcontrôleurs, la robotique, des expériences scientifiques, et bien d'autres choses encore.

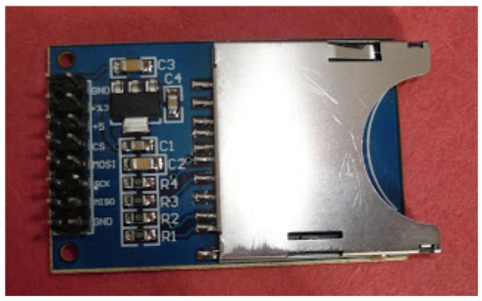
mercredi 6 février 2019

Carte SD et Blue Pill (STM32Duino)

Je continue mon exploration de la carte STM32F103 "Blue Pill" en l'associant, cette fois-ci, à un lecteur de carte SD, afin d'écrire et lire des fichiers sur une carte SD.

Comme d'habitude, j'utiliserai l'IDE Arduino pour programmer la carte (la marche à suivre a été expliquée ici).

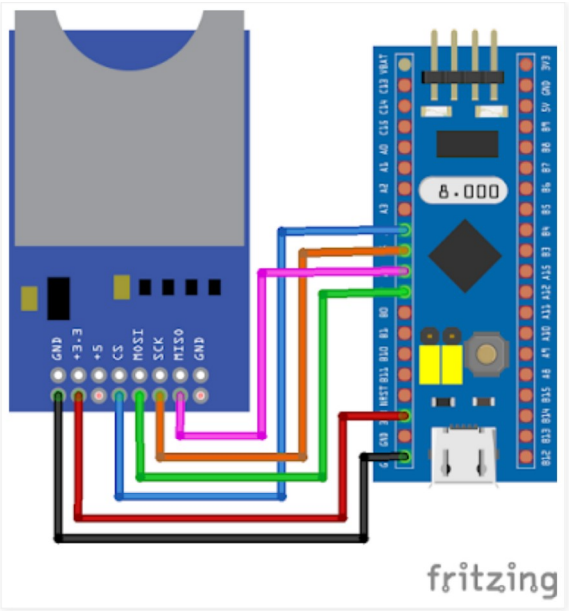
Mon lecteur de carte SD est un module qui communique au moyen du protocole SPI. C'est le modèle économique qu'on trouve facilement sur les sites de vente asiatiques. Il est conçu pour être utilisé à un niveau logique de 3,3 V, ce qui est parfait pour notre Blue Pill.



Connexions

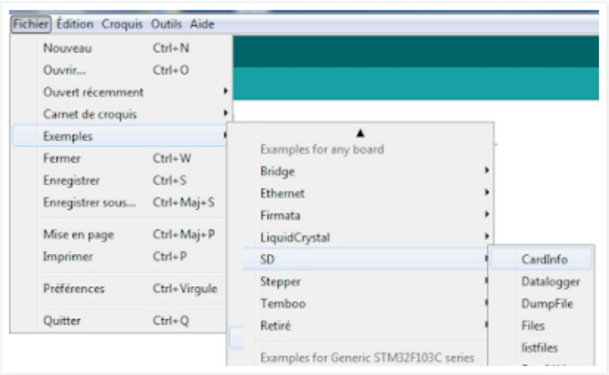
Le module lecteur de carte SD est branché à la Blue Pill de la façon suivante:

- GND du module carte SD --- GND de la Blue Pill
- +3.3 du module carte SD --- 3.3 de la Blue Pill
- +5 du module carte SD --- pas connecté
- CS du module carte SD --- A4 de la Blue Pill
- MOSI du module carte SD --- A7 de la Blue Pill
- SCK du module carte SD --- A5 de la Blue Pill
- MISO du module carte SD --- A6 de la Blue Pill



Vérification du bon fonctionnement

Une fois le lecteur de carte SD branché à la Blue Pill, l'exemple "*CardInfo*" de la bibliothèque "SD" est la première chose à essayer: il permet de vérifier que tout fonctionne correctement (notez qu'il n'est pas nécessaire d'installer la bibliothèque SD: il s'agit d'une bibliothèque déjà incluse par défaut avec l'IDE Arduino).



La seule modification à apporter au sketch "*Cardinfo*" est le numéro de la broche utilisée pour le "chip select", à la ligne 36: il faut remplacer "4" par "PA4":

```
const int chipSelect = PA4;
```

Si tout va bien, le moniteur série de l'IDE Arduino devrait montrer un message donnant diverses informations concernant la carte SD insérée dans le lecteur.

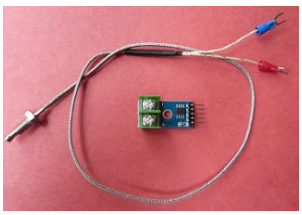
```
Initializing SD card...Wiring is correct and a card is present.

Card type:         SD2
Clusters:          61608
Blocks x Cluster:  32
Total Blocks:      1971456

Volume type is:    FAT16
Volume size (Kb):  985728
Volume size (Mb):  962
Volume size (Gb):  0.94

Files found on the card (name, date and size in bytes):
TRASH~1/     2019-01-08 16:47:54
  INFO/          2019-01-08 16:47:54
  FILES/         2019-01-08 16:47:54
  EXPUNGED/      2019-01-08 16:48:02
```

Si vous obtenez plutôt un message du genre "*initialization failed*", il y a un problème quelque part (et il faut le régler). Ces modules pour cartes SD ont la réputation d'être capricieux. Parfois, ça fonctionne avec une carte, et pas avec une autre!

Article récent

Utilisation d'un thermocouple avec MAX6675 et Arduino

Nous avons déjà eu de nombreuses occasions de mesurer une température avec une carte Arduino, en utilisant différents capteurs comme le DS1...



Articles les plus consultés

ESP32-CAM: première utilisation avec l'IDE Arduino

L'ESP32-CAM est carte offerte à prix très modique (moins de 10 euros) qui comporte un microcontrôleur ESP32 et une caméra OV2640. Une d...

Programmer l'ESP32 avec l'IDE Arduino (premiers pas)

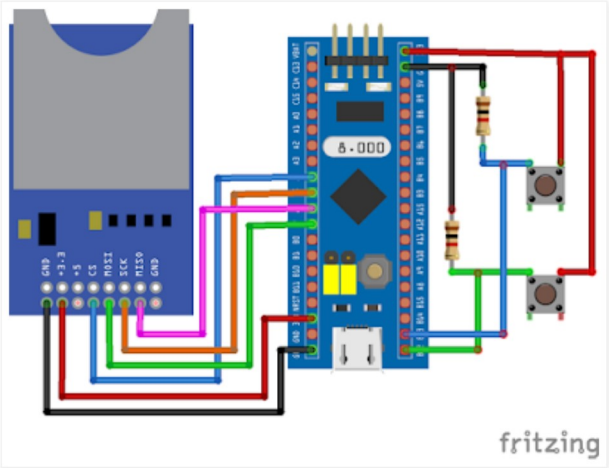
Puisque je me suis

```
Initializing SD card...initialization failed. Things to check:
* is a card inserted?
* is your wiring correct?
* did you change the chipSelect pin to match your shield or module?
```

Écriture et lecture d'un fichier

L'exemple "*ReadWrite*", situé dans le même répertoire que "*CardInfo*" montre assez clairement comment procéder pour écrire à l'intérieur d'un fichier texte sur la carte SD. Je vous propose ci-dessous une variante commentée en français et prête à être utilisée sans modifications sur votre Blue Pill.

Nous branchons deux boutons poussoir à la Blue Pill: un bouton qui nous permettra d'écrire quelque chose dans un fichier chaque fois qu'il sera enfoncé, et un deuxième bouton qui lira le contenu du fichier (et le transmettra au moniteur série) chaque fois qu'il sera enfoncé. J'ai branché ces boutons (associés à des résistances pull-down) aux broches B12 et B13 de la Blue Pill.



La bibliothèque SD permet d'interagir avec notre fichier sans trop de difficulté:

- **File monFichier**: création d'une variable de type "File" nommée "monFichier".
- **SD.begin(PA4)**: Initialise la communication avec la carte, PA4 étant la broche de la Blue Pill reliée à la broche "chip select" du lecteur de carte.
- **monFichier = SD.open("Archives.txt", FILE_WRITE)** : Ouverture du fichier texte intitulé "Archives.txt", dans le but de modifier son contenu. Si ce fichier n'existe pas, il sera créé, puis ouvert.
- **monFichier.print("Valeur du compteur: ")**: écriture des mots "Valeur du compteur: " à l'intérieur du fichier actuellement ouvert (qui est "Archives.txt").
- **monFichier.println(compteur)**: écriture de la valeur numérique de la variable "compteur" dans le fichier actuellement ouvert, suivi d'un saut de ligne.
- **monFichier.close()**: fermeture du fichier.
- **SD.open("Archives.txt")**: ouverture du fichier texte "Archives.txt" dans le but de le lire.
- **monFichier.available()**: retourne le nombre de bytes du fichier disponibles pour la lecture
- **monFichier.read()**: retourne le prochain caractère à lire dans le fichier (ou -1 s'il ne reste plus rien à lire).
- **monFichier.close()**: fermeture du fichier.

Il est important de toujours ouvrir le fichier avant de pouvoir le lire ou y écrire, et on n'oublie pas de le refermer ensuite.

Je n'ai pas utilisé toutes les classes disponibles dans la bibliothèque SD: il en existe d'autres pour, par exemple, faire la liste des fichiers d'un répertoire, effacer un fichier, etc. La liste et la description de toutes les classes peut être consultée sur le site officiel arduino.

```
1  /*************************************************************
2
```

beaucoup amusé avec l'ESP8266 au cours des derniers mois, il était inévitable que je fasse un jour ou l'autre la ...

Fabrication d'un anémomètre (Arduino)

Un anémomètre est un appareil qui permet de mesurer la vitesse du vent. C'est très facile de construire un anémomètre à coupelle, et d...

ESP32: Utilisation des entrées analogiques

La mesure d'une tension analogique a toujours été un point faible de l'ESP8266: le premier module (ESP-01) ne donnait accès à aucu...

Communication par nRF24L01 entre deux cartes Arduino

Aujourd'hui, nous établissons une communication sans fil entre deux cartes Arduino, au moyen d'une paire de récepteurs-émetteurs nR...

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 avions vu comment ...

Écran OLED SH1106 I2C et STM32

Le projet d'aujourd'hui consiste à afficher du texte et des images sur un petit écran OLED SH1106 i2c de 128 X 64 pixels. L'écra...

Utilisation d'un thermocouple avec MAX6675 et Arduino

Nous avons déjà eu de nombreuses occasions de mesurer une température avec une carte Arduino, en utilisant différents capteurs comme le DS1...

Capteur de couleurs TCS3200 (GY-31) et Arduino

Cette semaine, j'ai eu l'agréable surprise de trouver dans ma boîte aux lettres le capteur de couleurs TCS3200 que j'avais comma...

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 systè...

Abonnement flux RSS

Articles

Commentaires

```
  Exemple d'écriture et lecture de carte SD avec une Blue Pill
  Un bouton poussoir ajoute une ligne dans un fichier, l'autre
  bouton lit le fichier et communique son contenu dans le moniteur séri

  Plus d'infos:
  https://electroniqueamateur.blogspot.com/2019/02/carte-sd-et-blue-pil

*****************************************************************/

// blibliothèques nécessaires pour la carte SD
// (déjà installées dans l'IDE)
#include <SPI.h>
#include <SD.h>

const int boutonEcr = PB12; // broche du bouton "enregistrer"
const int boutonLire = PB13; // broche du bouton "lire"

// variables utiles pour la précédure anti-rebond
int etatBoutonEcr, etatBoutonLire;   //état actuel des boutons
int ancEtatBoutonEcr = 0, ancEtatBoutonLire = 0;   // état précédent de
unsigned long heureRebondEcr = 0, heureRebondLire = 0;  // à quel momen
unsigned long delaiRebond = 50;

File monFichier;
int compteur = 0;

void setup() {

  Serial.begin(9600); // initialisation de la communication série
  delay(100); // pour éviter que des caractères étranges apparaissent d

  // les deux boutons définis en entrée
  pinMode(boutonEcr, INPUT);
  pinMode(boutonLire, INPUT);

  Serial.print("Initialisation de la carte SD...");

  if (!SD.begin(PA4)) {  // "PA4" est la broche de la Blue Pill branché
    Serial.println("echec!");
    while (1);
  }
  Serial.println("reussie");
}

void loop() {

  int resultat;

  // bouton "enregisrer"

  resultat  = digitalRead(boutonEcr);

  if (resultat != ancEtatBoutonEcr) {
    heureRebondEcr = millis();
  }

  if ((millis() - heureRebondEcr) > delaiRebond) {
    if (resultat != etatBoutonEcr) {
      etatBoutonEcr = resultat;

      if (etatBoutonEcr == 1) {
        // on a appuyé sur le bouton enregistrer
        // ouverture (ou création) du fichier "Archives.txt")
```

Libellés

- Afficheur (46)
- App Inventor (5)
- Arduino (204)
- ATTiny (8)
- Blogs (12)
- Capteur (74)
- Circuit sans microcontrôleur (69)
- Communication (40)
- ESP32 (38)
- ESP8266 (33)
- Étude de CI (25)
- Instruments (18)
- Kit (6)
- Livre (33)
- Logiciels (27)
- Micro:Bit (6)
- MSP430 (30)
- PIC (19)
- Raspberry Pi (54)
- Réparation (6)
- Robotique (61)
- Science (43)
- Son et musique (61)
- STM32 (38)
- Trouver des composants (16)

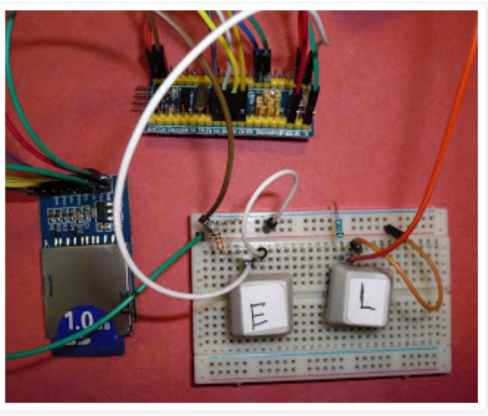
Nombre total de pages vues

3,110,863

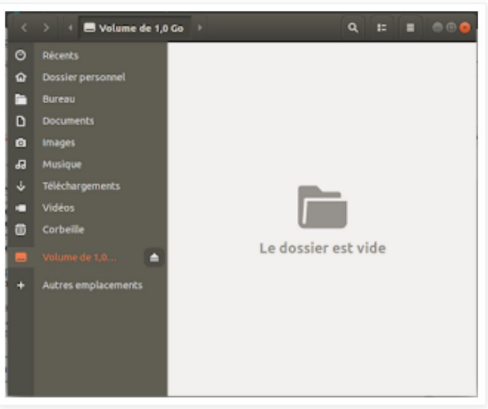
```
66          monFichier = SD.open("Archives.txt", FILE_WRITE);
67          // si l'ouverture a réussi, on écrit à l'intérieur
68          if (monFichier) {
69            Serial.print("Ecriture dans le fichier Archives.txt ... ");
70            monFichier.print("Valeur du compteur:  ");  // dans le fichie
71            monFichier.println(compteur);
72            compteur++;
73            // on referme le fichier
74            monFichier.close();
75            Serial.println("reussie.");
76          } else {
77            // si l'ouverture du fichier a échoué:
78            Serial.println("Erreur d'ouverture du fichier Archives.txt");
79          }
80        }
81      }
82    }
83
84    ancEtatBoutonEcr = resultat;
85
86
87    // bouton "lecture"
88
89    resultat  = digitalRead(boutonLire);
90
91    if (resultat != ancEtatBoutonLire) {
92      heureRebondLire = millis();
93    }
94
95    if ((millis() - heureRebondLire) > delaiRebond) {
96      if (resultat != etatBoutonLire) {
97        etatBoutonLire = resultat;
98
99        if (etatBoutonLire == 1) {
100         // on a appuyé sur le bouton lire, alors on ouvre le fichier
101         monFichier = SD.open("Archives.txt");
102         if (monFichier) {
103           Serial.println("Contenu du fichier Archives.txt:");
104           // on lit le contenu du fichier jusqu'à la fin
105           while (monFichier.available()) {
106             Serial.write(monFichier.read()); // transcription dans le m
107           }
108           // fermeture du fichier:
109           monFichier.close();
110         } else {
111           // en cas d'erreur lors de l'ouverture du fichier:
112           Serial.println("Erreur d'ouverture du fichier Archives.txt");
113         }
114       }
115     }
116   }
117
118   ancEtatBoutonLire = resultat;
119
120 }
121
```

BluePill_SD.ino hosted with ❤ by GitHub view raw

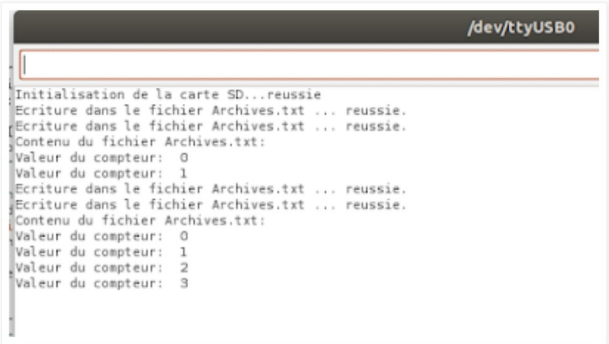
Mise à l'essai



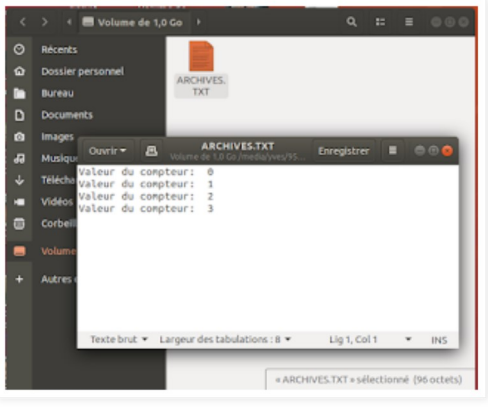
Au départ, la carte SD était vide.



J'ai appuyé deux fois sur le bouton "écrire", puis une fois sur le bouton "lire". Le moniteur série m'indique le contenu du fichier (2 lignes de texte). J'appuie encore deux autres fois sur le bouton "écrire", et une fois sur le bouton lire: le moniteur série me montre les 4 lignes de texte maintenant enregistrées dans le fichier.



La carte SD contient maintenant un fichier intitulé "ARCHIVES.TXT dans lequel 4 lignes de texte ont été enregistrées.



Yves Pelletier (Twitter, Facebook)

Publié par Yves Pelletier à 07:33

Libellés : STM32

Aucun commentaire:

Publier un commentaire

Saisissez votre commentaire…

Ajouter un commentaire en tant que : Compte Google

Publier **Aperçu**

Article plus récent Accueil Article plus ancien

Inscription à : Publier les commentaires (Atom)