

LATEST NEWS

Controlador de passo com M983 (<http://www.dobitaobyte.com.br/controlador-de-passo-com-m983/>)

Programar para Android com Qt para-android-com-qt/

👤 | 🔍



(<http://www.dobitaobyte.com.br/>)

Embarcados, Linux, programação e IoT

DO BIT AO BYTE ([HTTP://WWW.DOBITAObYTE.COM.BR/](http://www.dobitaobyte.com.br/))

ARDUINO ([HTTP://WWW.DOBITAObYTE.COM.BR/GERAL/ARDUINO/](http://www.dobitaobyte.com.br/geral/arduino/))

GERAL ([HTTP://WWW.DOBITAObYTE.COM.BR/GERAL/GERAL/](http://www.dobitaobyte.com.br/geral/geral/))

Ler e escrever no SD card com Arduino

Djames Suhanko (<http://www.dobitaobyte.com.br/author/djames-suhanko/>) - 31 de dezembro de 2016 (<http://www.dobitaobyte.com.br/ler-e-escrever-no-sd-card-com-arduino/>)



Você provavelmente já pensou em expandir de alguma forma a memória do Arduino, seja para fazer um data logger ou para armazenar outros tipos de dados como, por exemplo, imagens para exibir no display. O propósito do artigo de hoje será lhe habilitar a utilizar o cartão SD (ou micro SD com adaptador) para leitura e escrita, através de um módulo para Arduino, utilizando o barramento SPI para fazer a comunicação.

Tópicos [Ocultar]

- 1 Lista de materiais
- 2 Barramento SPI
- 3 Biblioteca
- 4 Troubleshooting
 - 4.0.1 Alimentação
 - 4.0.2 Sobrecarga
 - 4.0.3 Defeito no SD card
 - 4.0.4 Wiring
 - 4.0.5 Nivel lógico
- 5 Compartilhe com:
- 6 Relacionado

Lista de materiais

Você precisará de quase nada para essa brincadeira! Vamos ao mais simples possível, sendo 3 itens:

- SD card reader (<https://www.autocorerobotica.com.br/modulo-sdcard>)
- Arduino (exemplificado com o UNO, mas basta atentar-se ao wiring de qualquer outro) (<https://www.autocorerobotica.com.br/placa>)

- Jumpers macho-fêmea (<https://www.autocorerobotica.com.br/cabos-dupont-fxm-pacote-com-20pcs>)

Barramento SPI

É melhor iniciar a explicação pelo barramento porque o SPI é um barramento serial um pouco mais complexo que o UART ou I²C. De vantagem, você tem um barramento mais rápido que os demais e acaba sendo o ideal para escrita e leitura do cartão SD porque certamente você não querará aguardar tempo demasiado pela carga dos dados.

A comunicação SPI contempla 4 pinos, dos quais 2 são elegíveis, enquanto outros 2 são forçosamente estáticos. Os pinos são esses descritos na tabela a seguir:

Nome	Pino do Arduino	Descrição
MOSI	Elegível. Sugerido: 11	MOSI significa: Master Output Slave Input, ou seja, saída do master para a entrada do slave.
MISO	Elegível. Sugerido: 12	MISO significa: Master Input Slave Output, ou seja, a saída do slave vai para a entrada do master.
SCK	Dependente do Arduino. No Uno o pino é o 13	SCK é Serial Clock. Cada Arduino possui o barramento SPI em um lugar diferente e será necessário consultar o pinout conforme a board utilizada.
CS	Dependente do Arduino. No Uno o pino é o 4	CS significa Chip Select. Em alguns casos você pode encontrar a descrição SS, que significa Slave Select.

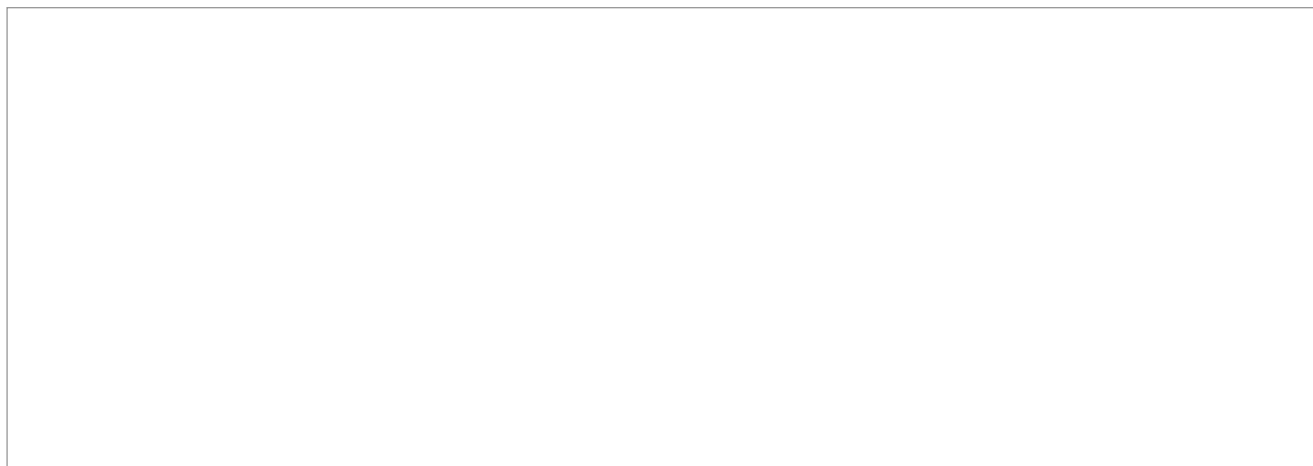
O barramento SPI suporta diversos dispositivos simultaneamente, mas no nosso caso estamos focando exclusivamente na utilização do SD card.

Biblioteca

Se você procurar através do Library Manager da IDE do Arduino, você certamente encontrará ao menos 2 bibliotecas para tocar áudio a partir do SD card. Já se procurar diretamente nos exemplos no submenus de File->Examples-SD, diversas opções estarão disponíveis, dentre elas, escrever e ler no SD card. Abra-o para que nos mantenhamos em sintonia.

No início desse sketch você notará que coincidentemente o wiring sugere o mesmo esquema descrito na tabela supracitada. Porém esse pinout/wiring é próprio do Uno. Supondo que você queira utilizar o Arduino Leonardo, você deverá previamente fazer uma consulta no google por “Leonardo pinout”. Ou veja o esquema nesse meu outro post (<http://www.dobitaobyte.com.br/modulo-nrf24l01-com-arduino/>) onde trato de comunicação SPI com um Arduino Uno e um Leonardo.

De resto, são algumas poucas e ridículamente simples linhas que o sketch utiliza para mostrar o quão simples é escrever no SD card. Vou dispor o código do exemplo devidamente adulterado para me favorecer de alguma forma:



```
1  /*
2  ** MOSI - pin 11
3  ** MISO - pin 12
4  ** CLK - pin 13
5  ** CS - pin 4
6  */
7
8  #include <SPI.h>
9  #include <SD.h>
10
11 //criar um objeto File para manipular um arquivo
12 File myFile;
13
14 void setup() {
15   Serial.begin(9600);
16   while (!Serial);
17
18   Serial.print("Inicializa o SD card...");
19
20   if (!SD.begin(4)) {
21     Serial.println("Nao inicializado. Veja no texto adiante sobre isso.");
22     return;
23   }
24   Serial.println("Beleza! Seguindo...");
25
26   /* A biblioteca SD tem um metodo para abrir arquivos e esse arquivo aberto
27   sera armazenado no objeto myFile, descrito ao inicio desse codigo. Somente
28   um arquivo pode ser aberto por vez, portanto nao se esqueca de fecha-lo
29   antes de abrir um novo ou voce tera problemas.
30   */
31
32   myFile = SD.open("bB.txt", FILE_WRITE);
33
34   Se o arquivo foi aberto com sucesso, escreve nele
35   if (myFile) {
36     Serial.print("Escrevendo para bB.txt");
37     myFile.println("dobitaabyte.com.br");
38
39     // Terminou de escrever, fecha-se o arquivo:
40     myFile.close();
41
42     Serial.println("Escrevinhado.");
43   }
44   else {
45     // Se nao deu certo, comeca a dar merda desse ponto
46     Serial.println("Nao foi possivel abrir o arquivo");
47   }
48
49   // Agora vamos abri-lo para leitura
50   myFile = SD.open("bB.txt");
51   if (myFile) {
52     Serial.println("Conteudo de bB.txt:");
53
54     // Le todo o arquivo...
55     while (myFile.available()) {
56       Serial.write(myFile.read());
57     }
58     //Finalizado, fecha-se o arquivo
59     myFile.close();
60   }
61   else {
62     // Se deu erro acima, aqui tambem dara.
63     Serial.println("Nao pude abrir o arquivo.");
64   }
65 }
66
67 void loop() {
68   // Aqui poderia ser colocada uma rotina para leitura de sensores e entao os dados
69   // poderiam ser gravados, precedendo-os pelo timestamp. Ai esta seu data logger!
70 }
71 }
```

Troubleshooting

Não existem muitos erros prováveis, o que facilita a depuração do erro. Dentre as mais comuns estão:

Alimentação

Se o Arduino estiver com uma fonte externa de baixa corrente, provavelmente você poderá não suprir a alimentação de forma eficiente e isso poderá gerar erros estranhos.

A alimentação do módulo deve vir preferencialmente de uma tensão externa ao Arduino, de forma regulada. Isso porque o consumo do módulo pode chegar à 200mA!

Sobrecarga

Se você estiver com uma fonte com carga suficiente para o Arduino, mas estiver sobrecarregados de periféricos (sensores, LEDs etc), então você poderá obter erros semelhantes ao anteriormente citado.

Defeito no SD card

Se o cartão tiver algum problema físico em sua estrutura, gerará erros tal qual acontece em um celular ou em um notebook/PC.

Wiring

