



## **Problema 15: Acelerômetro Triaxial**

Lucas Locatelli Helena Nº 9041079  
Natanael Dias Correia Nº 8957263

Depto. De Engenharia Elétrica e de Computação  
Sel0352 – Instrumentação Eletrônica II  
Prof. Dr. Edson Gesualdo

São Carlos, 2017

## Problema

Usar um acelerômetro triaxial para medir as diversas acelerações que o carro do BAJA sofre durante a corrida. O acelerômetro será usado somente em testes.

## Projeto

Para solução do problema foi escolhido o módulo ADXL345 para captar as acelerações e um Arduino Uno para gravar os dados no cartão de memória, o motivo dessa escolha foi o tamanho compacto dos componentes, o baixo consumo de energia, a portabilidade e o preço de construção.

### ADXL345



Figura 1 Módulo ADXL345.

O ADXL345 é um acelerômetro triaxial pequeno, compacto e com baixo consumo de energia, tem uma alta resolução em suas medições (13 bits) medindo até  $\pm 16g$  (3.9 mg/LSB), com interfaces de comunicação digital SPI e I<sup>2</sup>C. As acelerações obtidas pelo acelerômetro saem em função de g (aceleração da gravidade).

## Ethernet Shield 5100

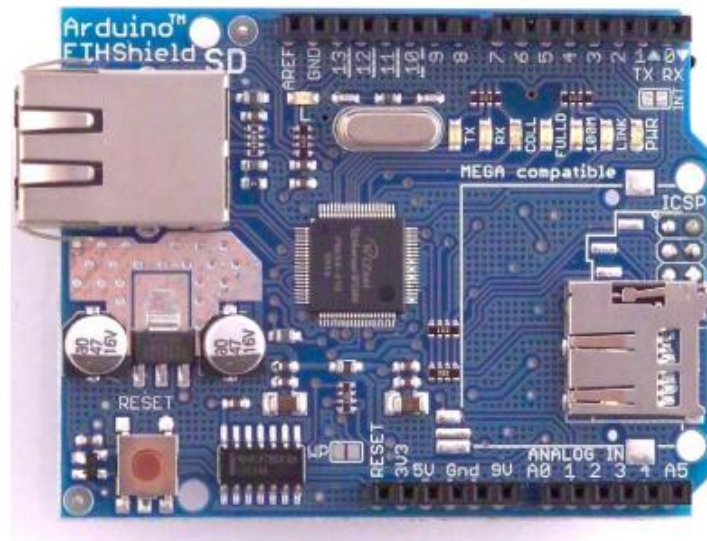


Figura 2 Ethernet shield 5100.

É um shield para o Arduino baseado no chip ethernet Wiznet W5100, porém nessa aplicação ele é utilizado por seu slot para cartão micro-SD, para guardar os dados coletados, ele se comunica com o Arduino por SPI.

## Arduino Uno



Figura 3 Arduino UNO.

Arduino é uma plataforma aberta de prototipagem eletrônica, é composta por Hardware, projetada com um microcontrolador Atmel AVR e uma linguagem de programação baseada em

Wiring. É usada nesse projeto para ler os dados no acelerômetro, tratá-los e armazenar no cartão SD.

## Montagem

Foram utilizados dois resistores de pull-up de 20 k $\Omega$ , pois a comunicação por I<sup>2</sup>C requer. O circuito final montado está demonstrado na figura, onde o arduino foi programado para gravar os dados provenientes do acelerômetro no cartão de memória. O código feito pode ser encontrado no final deste projeto.

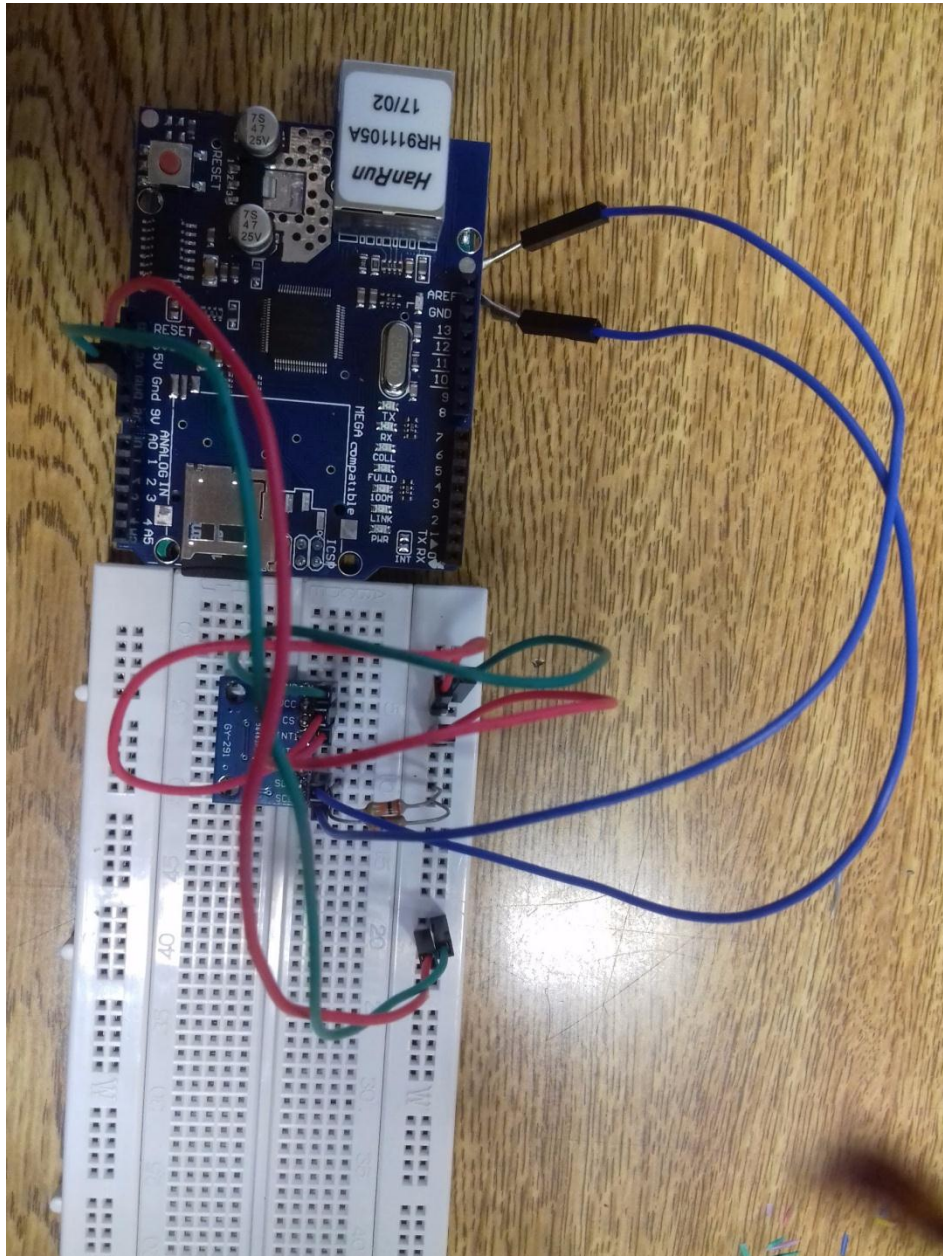


Figura 4 Circuito Final montado



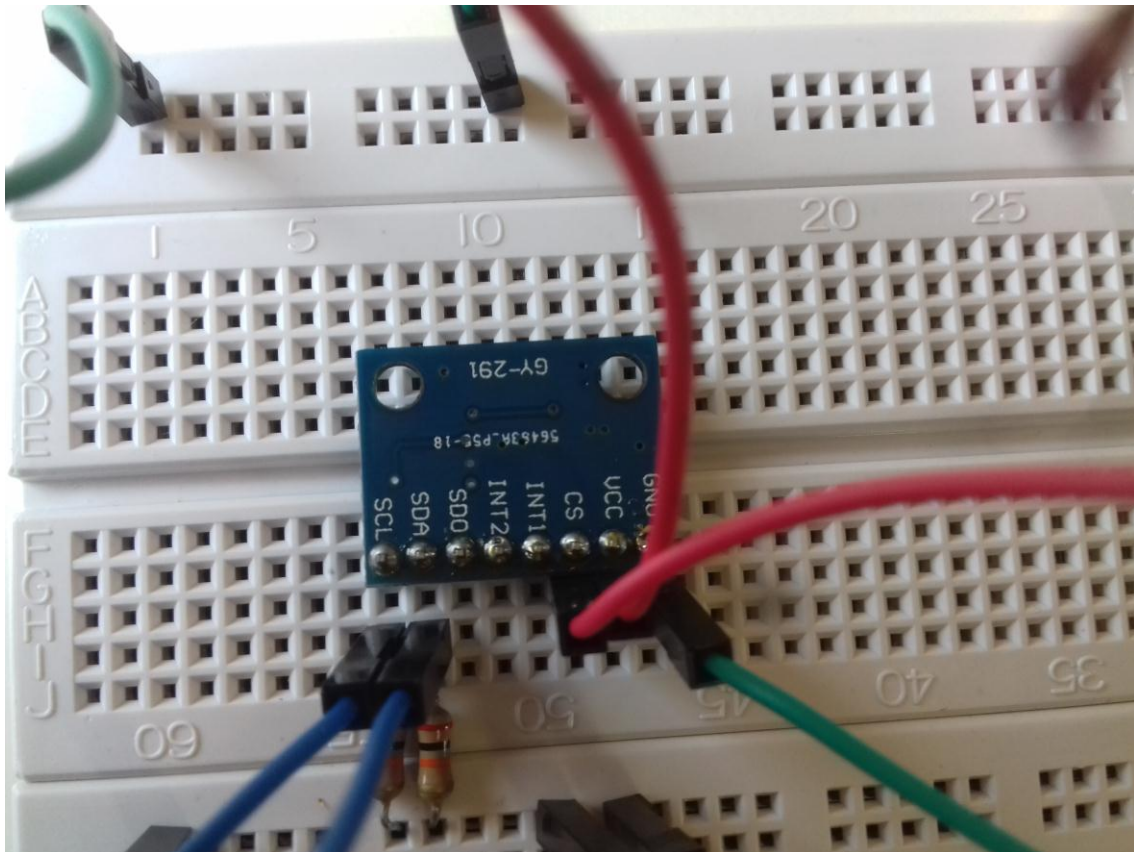


Figura 5: zoom sobre as conexões com o acelerômetro

Na figura acima vemos de perto as conexões com o acelerômetro. CS em nível alto seleciona o modo I2C. Neste caso, a strings de dados saem por SDA que vão para o arduino. SCL funciona como clock. Ambos SCL e SDA mandam ou zero ou alta impedância nas suas saídas. Por conta disso, temos que utilizar nas suas saídas dois resistores de pull-up de 20 kohms conectados com a tensão VCC. Para “criar” o estado 1 e ainda evitar o curto-circuito. As entradas INT2 e INT1 são de interrupções e não as utilizamos neste projeto.

## Resultados

Fizemos um código para programar o arduino para salvar no cartão de memória. Após a programação, realizamos alguns testes com o acelerômetro, todos os dados foram salvos no cartão SD em um arquivo.txt, onde os valores obtidos das acelerações nos três eixos ficam em função de g. Neste caso pode ser inserido no MatLab para tratamentos dos dados, para mudar e obter os valores absolutos das acelerações.

## Instalação no carro do baja

Feito os testes em protoboard apresetado na Figura 5 acima, para comprovar o correto funcionamento do circuito, nós teríamos apenas que imprimir um case para fixar o circuito no carro do baja. Para essa instalação nós precisaríamos de uma bateria 9V junto com um adaptador para conectá-la ao arduino conforme figura 5 abaixo. A bateria é recarregável e

possui uma carga alta, assim não teria o risco de descarregá-la durante os testes e além disso, o sistema consome pouca energia.

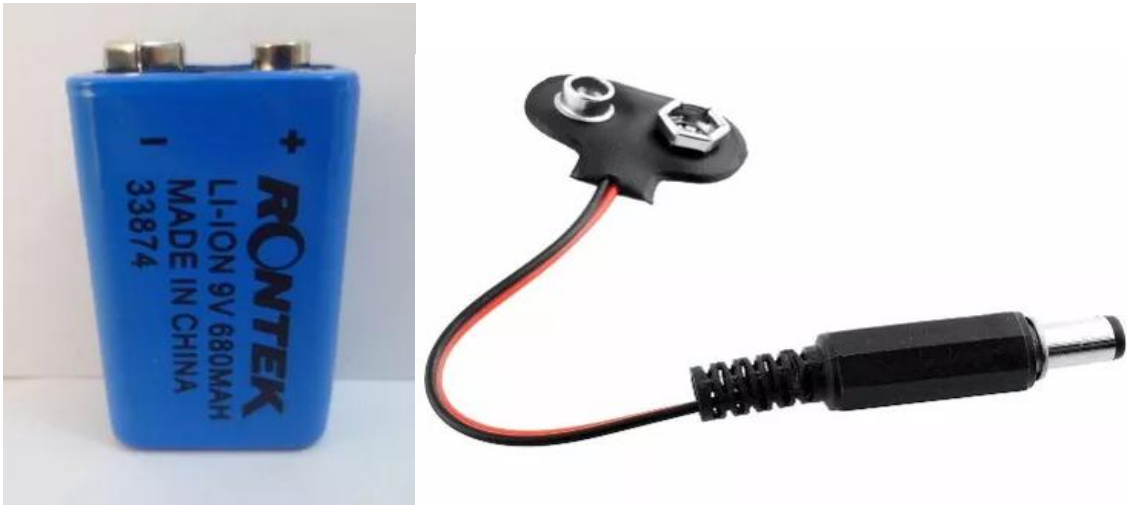


Figura 6 Bateria e a daptador escolhido.

## Preço do projeto

O preço do projeto foi estipulado abaixo:

Acerelômetro ADXL345: 20 R\$

Arduino UNO: 45R\$

Shield Ethernet: 35 R\$

Bateria 9V recarregável: 60 R\$

Adaptador para bateria: 4 R\$

Total: 164 Reais

## Conclusão

Os resultados obtidos foram satisfatórios e os componentes funcionaram dentro das especificações de seus data sheets. O preço do projeto ficou em torno de 164 reais usando uma bateria recarregável, e portanto, mais caras que as baterias comuns. Para trabalhos futuros utilizaríamos essa bateria de 9V para alimentar o sistema e a impressão de um case para o circuito para fixá-lo ao carro.

## Referências:

DataSheet do ADXL345: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>

# Apêndice

Código utilizado para salvamento dos dados do acelerômetro no cartão de memória:

```
#include <Wire.h>
#include <SPI.h>
#include <SD.h>

#define accel_module (0x53)
byte values[6] ;
char output[512];
File myFile;

void setup(){
  Wire.begin();
  Serial.begin(9600);

  while (!Serial) {
  }

  Serial.print("Initializing SD card...");
  if (!SD.begin(4)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");

  Wire.beginTransmission(accel_module);
  Wire.write(0x2D);
  Wire.write(0);
  Wire.endTransmission();
  Wire.beginTransmission(accel_module);
  Wire.write(0x2D);
  Wire.write(16);
  Wire.endTransmission();
  Wire.beginTransmission(accel_module);
  Wire.write(0x2D);
  Wire.write(8);
  Wire.endTransmission();
}

void loop(){
  int xyzregister = 0x32;
  int x, y, z;
  float x1,y1,z1;

  Wire.beginTransmission(accel_module);
  Wire.write(xyzregister);
  Wire.endTransmission();

  Wire.beginTransmission(accel_module);
  Wire.requestFrom(accel_module, 6);

  int i = 0;
  while(Wire.available()){
```

```
values[i] = Wire.read();
i++;
}
Wire.endTransmission();

x = (((int)values[1]) << 8) | values[0];
y = (((int)values[3])<< 8) | values[2];
z = (((int)values[5]) << 8) | values[4];
sprintf(output, "%d %d %d", x, y, z);
Serial.print(output);
Serial.write(10);

myFile = SD.open("Data.txt", FILE_WRITE);
// if the file opened okay, write to it:
if (myFile) {
  Serial.print("Writing to test.txt...");
  myFile.print("x: ");
  myFile.print(x);
  myFile.print("y: ");
  myFile.print(y);
  myFile.print("z: ");
  myFile.println(z);
  // close the file:
  myFile.close();
  Serial.println("done.");
} else {
  // if the file didn't open, print an error:
  Serial.println("error opening test.txt");
}
delay(1000);
}
```