



Índice

Índice	2
Introdução	3
Hardware	3
Bibliotecas	3
Instalação	4
Pré-requisitos	4
Hardware	4
Bibliotecas	6
Via ficheiro .ZIP	6
Via manual	6
Utilizar o BAGA	8
Biblioteca BAGA	9
Exemplo DisplayData	10
Exemplo DataLogger	11
Outros exemplos	12

Introdução

O BAGA é um *kit* de desenvolvimento baseado em Arduino™, orientado à recolha de dados meteorológicos para suporte às atividades agrícolas.

Uma das características deste *kit* é ser programável a partir da plataforma Arduino™, podendo ser programado para efetuar diferentes tarefas, tirando partido dos diversos sensores presentes na placa. Junto com o *kit* é disponibilizada uma biblioteca de funções para facilitar a sua configuração e utilização.

Alguns termos utilizados neste manual derivam da filosofia Arduino e das linguagens de programação:

- **Biblioteca** – Bloco de software que visa criar uma ou mais camadas de abstração entre programador e o software de mais baixo nível, permitindo assim a utilização do hardware de uma forma mais simples e intuitiva.
- **Sketch** – Ficheiro com código fonte que faz uso das funções (métodos) de uma ou mais bibliotecas de forma a criar um programa funcional para execução no hardware.

Hardware

O hardware BAGA disponibiliza:

- Sensores
 - Humidade relativa
 - Temperatura
 - Pressão atmosférica
 - Radiação solar
 - Radiação UV (implementação futura)
- Comunicação
 - USB-CDC (porta série emulada)
 - UART
 - RF (implementação futura)
- Armazenamento
 - Slot para cartão MicroSD

Bibliotecas

A biblioteca BAGA utiliza algumas bibliotecas de suporte, nomeadamente as bibliotecas Si7020 para o sensor de humidade relativa e temperatura, e a BMP280 para o sensor de pressão atmosférica e temperatura.

São ainda usadas bibliotecas *open source* de utilização gratuita, como por exemplo, a biblioteca SdFat que dá suporte à utilização de cartão SD.

Instalação

Pré-requisitos

Para utilizar a placa BAGA será necessário o seguinte equipamento:

- Computador com 1 porta USB disponível
- Cabo USB Male A- Male B
- 2 Baterias do tipo AA 1.5V
- Cartão MicroSD
- Arduino IDE
- Bibliotecas Globaltronic - <http://github.com/Globaltronic/BAGA>
 - BAGA
 - Si7020
 - BMP280
- Bibliotecas externas:
 - SdFat - <http://github.com/greiman/SdFat>

Hardware

Na primeira utilização do BAGA, após ligação do dispositivo ao computador via USB, este deverá ser reconhecido e instalado automaticamente como **Arduino Leonardo**.

Caso este não seja instalado automaticamente, é necessário instalar o controlador manualmente. Para isso será necessário abrir o **Gestor de Dispositivos** e identificar o **BAGA (Arduino Leonardo)** na lista de **Outros Dispositivos** (Figura 1).

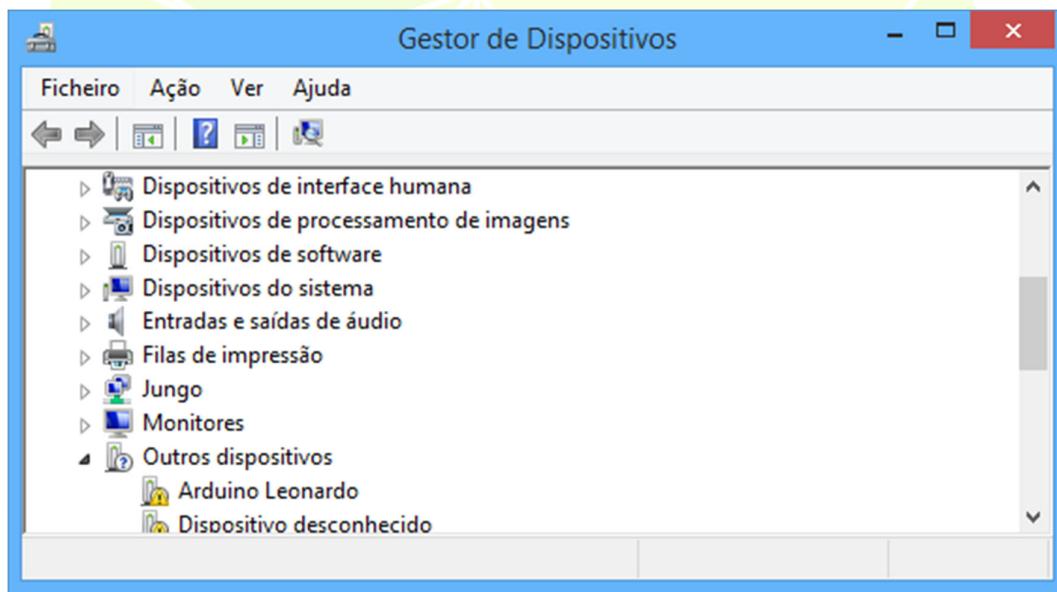


Figura 1 - Gestor de Dispositivos

Clicar com o botão direito no dispositivo “**Arduino Leonardo**” e escolher a opção “**Atualizar controlador de Software...**”.

Irá surgir uma nova janela para atualizar o controlador e deverá ser escolhida a opção “**Procurar software do controlador no computador**”.

Na nova janela (Figura 2) deve procurar a pasta de instalação do Arduino IDE (normalmente em **C:** ou **C:\Programas** ou no local para onde descompactou a mesma, se optou pela versão zip do IDE) e selecionar a pasta **Drivers** que está dentro desta.

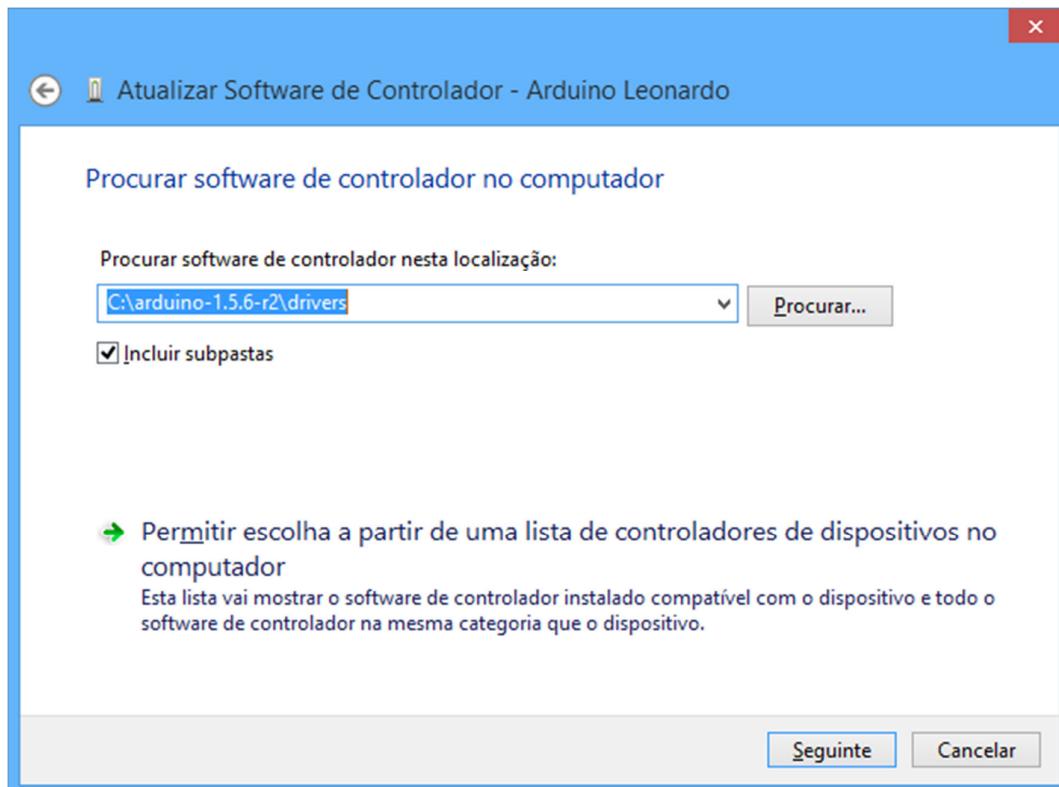


Figura 2 - Atualizar Software de Controlador

Ao clicar “**Seguinte**” o controlador deverá ser instalado e apresentada uma mensagem de sucesso.

Se o *software* for instalado corretamente, a lista de dispositivos deverá passar a apresentar o dispositivo **Arduino Leonardo** no grupo **Portas (COM e LPT)** (Figura 3).

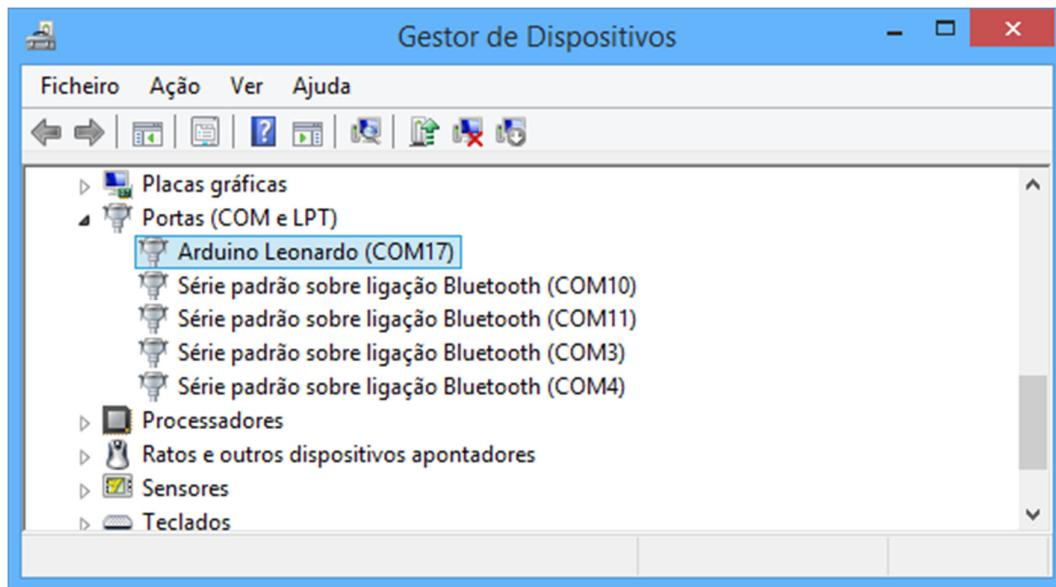


Figura 3 - Gestor de Dispositivos

Bibliotecas

Para utilizar a placa BAGA deverão ser instaladas as bibliotecas listadas no capítulo **Instalação >> Pré-requisitos**.

Via ficheiro .ZIP

1. Abrir o Arduino IDE
2. Na barra de menu superior, abrir: **Sketch >> Include Library >> Add .ZIP Library**
3. Selecionar o ficheiro **BAGA.zip** e clicar em abrir (deverá ser mostrada uma mensagem de sucesso)
4. Repetir os passos 2 e 3 para instalar as bibliotecas **Si7020** (sensor de temperatura e humidade relativa) e **BMP280** (sensor de pressão atmosférica e temperatura).
5. Reiniciar o **Arduino IDE** para atualizar o menu de exemplos.

Via manual

Geralmente, uma biblioteca Arduino é composta por uma pasta com uma estrutura idêntica à da Figura 4, que inclui:

- Pasta de exemplos - Opcional
- Ficheiro **<biblioteca>.cpp**
- Ficheiro **<biblioteca>.h**
- **keywords.txt** - Opcional
- **library.properties** - Opcional

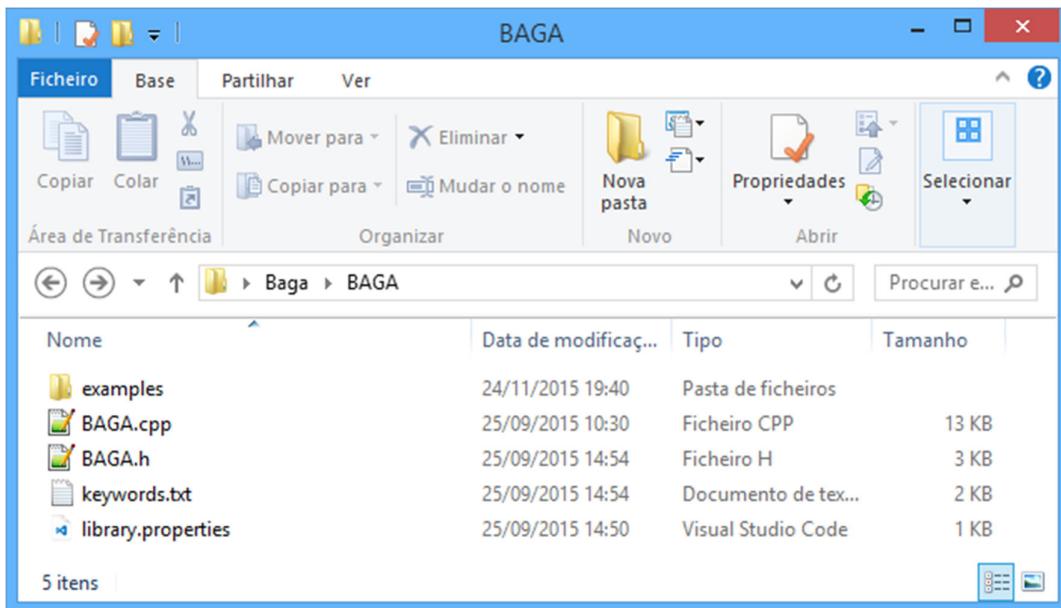


Figura 4 - Biblioteca BAGA

Exemplo de instalação da biblioteca **BAGA** via manual:

1. Descomprimir o ficheiro da biblioteca a instalar (Figura 5)

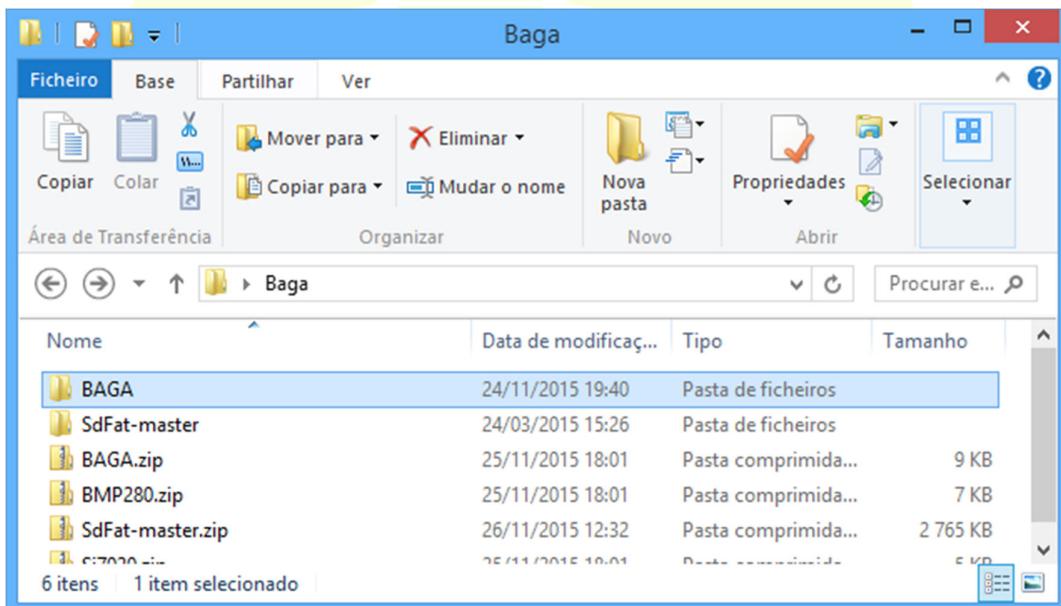


Figura 5 - Biblioteca BAGA

2. Identificar se a pasta corresponde a uma biblioteca Arduino ou um conjunto de bibliotecas verificando a estrutura da pasta (Figura 6).

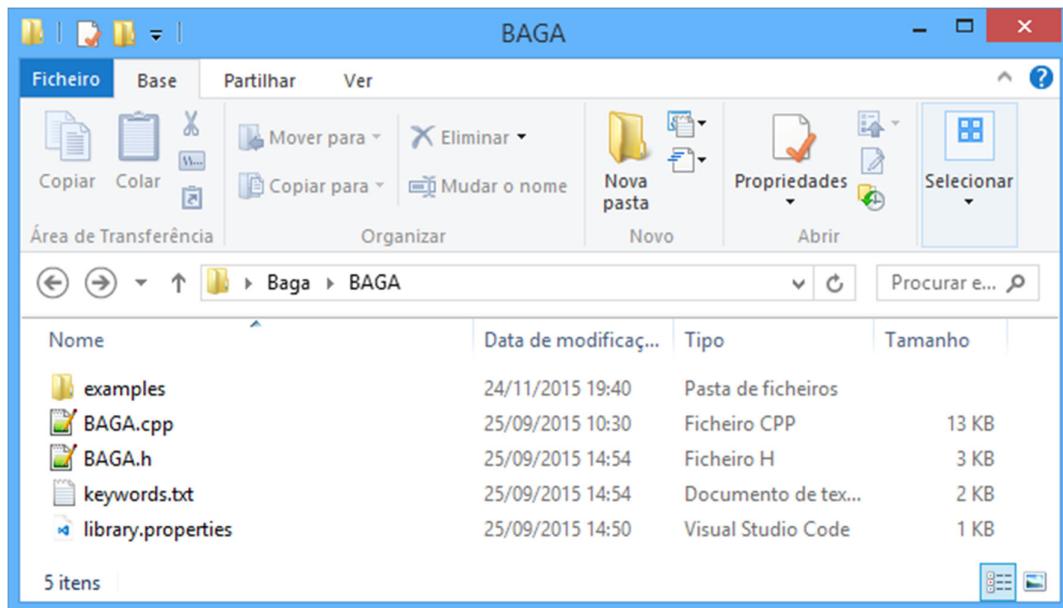


Figura 6 - Estrutura da pasta da biblioteca BAGA

3. Copiar a biblioteca para a pasta de bibliotecas Arduino em **Documentos/Arduino/libraries**.

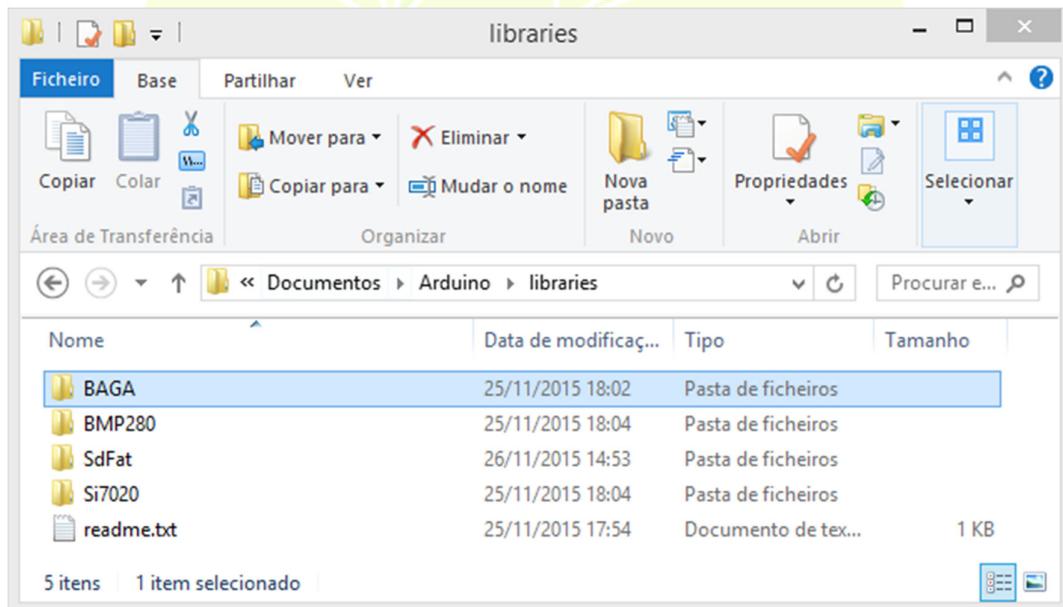


Figura 7 - Pasta de bibliotecas Arduino

4. Quando o **Arduino IDE** for reiniciado, a nova biblioteca já deverá estar listada.

Utilizar o BAGA

Após a instalação do *hardware*, este deve ser devidamente identificado e selecionado nas configurações do Arduino IDE. Para isso, no menu **Tools >> Board**, deve ser escolhida a opção

Arduino Leonardo. De seguida, em **Tools >> Port**, deve ser selecionada a porta identificada como “**COMx (Arduino Leonardo)**” que deverá corresponder à porta listada no Gestor de Dispositivos (Figura 3).

Biblioteca BAGA

Abaixo está uma lista de métodos e propriedades da biblioteca BAGA e a respetiva função.

Método	Descrição
begin	Inicializar a placa BAGA
readSensors	Ler informação de todos os sensores e armazenar nas variáveis internas. Todos os valores lidos utilizando os métodos abaixo, correspondem aos valores da última leitura efetuada com este método.
readTemperatureC	Ler a temperatura (em Celcius)
readTemperatureF	Ler a temperatura (em Fahrenheit)
readTemperatureK	Ler a temperatura (em Kelvin)
readHeatIndexC	Ler a temperatura aparente (em Celcius)
readHeatIndexF	Ler a temperatura aparente (em Fahrenheit)
readHeatIndexK	Ler a temperatura aparente (em Kelvin)
readAbsHumidity	Ler a humidade absoluta (em g/m ³)
readRelHumidity	Ler a humidade relativa (em %)
readAbsPressuremBar	Ler a pressão atmosférica absoluta (em Milibar)
readAbsPressureBar	Ler a pressão atmosférica absoluta (em Bar)
readAbsPressurePa	Ler a pressão atmosférica absoluta (em Pascal)
readAbsPressureKPa	Ler a pressão atmosférica absoluta (em Kilopascal)
readAbsPressurePsi	Ler a pressão atmosférica absoluta (em PSI)
readRelPressureBar	Ler a pressão atmosférica relativa (em Milibar)
readRelPressurePa	Ler a pressão atmosférica relativa (em Pascal)
readRelPressureKPa	Ler a pressão atmosférica relativa (em Kilopascal)
readRelPressurePsi	Ler a pressão atmosférica relativa (em PSI)
readSolarRadiation	Ler sensor de radiação solar (em Watt/m ²)
readBatteryVoltage	Ler tensão das baterias (em milivolts)
setLedOn	Activar o LED SYS
setLedOff	Desactivar o LED SYS

Antes de criar um novo projeto para a placa BAGA, devem ser testados os *sketches* de exemplo incluídos na biblioteca fornecida para conhecer melhor o funcionamento da mesma.

Os sketches de exemplo podem ser carregados a partir do submenu da biblioteca em **File >> Examples**.

Para criar um novo sketch de raiz e utilizar a biblioteca BAGA, basta clicar em **File >> New** e incluir as bibliotecas a partir de **Sketch >> Include Library**.

Exemplo DisplayData

Um dos exemplos fornecidos é o **DisplayData**. Este exemplo consiste na recolha periódica de informação dos sensores e apresentação, da mesma, através da porta USB da placa (que emula uma porta COM). O led de sistema é ativado durante a leitura dos sensores.

Conforme apresentado na Figura 8, para utilizar a biblioteca BAGA é necessário incluir as bibliotecas:

- Wire – Comunicação I²C
- BMP280 – Sensor de pressão atmosférica e temperatura
- Si7020 – Sensor de humidade relativa e temperatura
- BAGA – Placa BAGA

DebugSerial – É a porta de comunicação utilizada para enviar as informações dos sensores para o computador. Neste caso poderá ser definida como Serial (porta emulada por USB) ou Serial1 que é uma porta UART (TTL 5V) no conector J3 da placa.

```

1 #include <Wire.h>
2 #include <BMP280.h>
3 #include <Si7020.h>
4 #include <BAGA.h>
5
6 // Select the desired debug serial port:
7 // * Serial (USB-CDC)
8 // * Serial1 (J3 connector on board)
9
10#define DebugSerial    Serial
11
12
13
14 /**
15 * BAGA A1
16 * Data Display Example
17 *
18 * This example reads data periodically and prints it to the :
19 *
<           >

```

Arduino Leonardo on COM17

Figura 8 - Sketch DisplayData

A Figura 8 e a Figura 9 apresentam excertos do *sketch* de exemplo **DisplayData**.

O intervalo entre medições pode ser configurado alterando a variável seguinte:

updateIntervalMilliseconds – Intervalo de tempo entre leituras, em milissegundos.

```

File Edit Sketch Tools Help
DisplayData | Arduino 1.6.5
BAGA baga; // BAGA instance
int updateIntervalMilliseconds = 5000; // Read data every 5 s
char strOk[7] = "[ OK ]";
char strFail[7] = "[FAIL]";
void setup() {
    // put your setup code here, to run once:
    byte result;
    DebugSerial.begin(9600); // Initialize serial port
    while (!DebugSerial)
    {
        ; // wait for serial port to connect. Needed for Leonardo
    }
    DebugSerial.println("##### BAGA #####");
    result = baga.begin();
}

```

Arduino Leonardo on COM17

Figura 9 - Sketch DisplayData

Exemplo DataLogger

O exemplo **DataLogger** efetua igualmente recolha de informação dos sensores, armazenando a mesma num ficheiro na raiz do cartão MicroSD, em formato CSV.

Comparativamente ao exemplo **DisplayData**, serão necessárias bibliotecas adicionais que permitem a utilização do cartão MicroSD, neste caso as bibliotecas **SPI** e **SdFat**. Em cada *sketch*, apenas devem ser incluídas as bibliotecas estritamente necessárias de forma a otimizar o uso da memória do dispositivo.

A biblioteca usada para utilização do cartão MicroSD é a biblioteca **SdFat** em alternativa à biblioteca **SD** pré-instalada com o Arduino IDE, por esta ocupar mais memória.

```

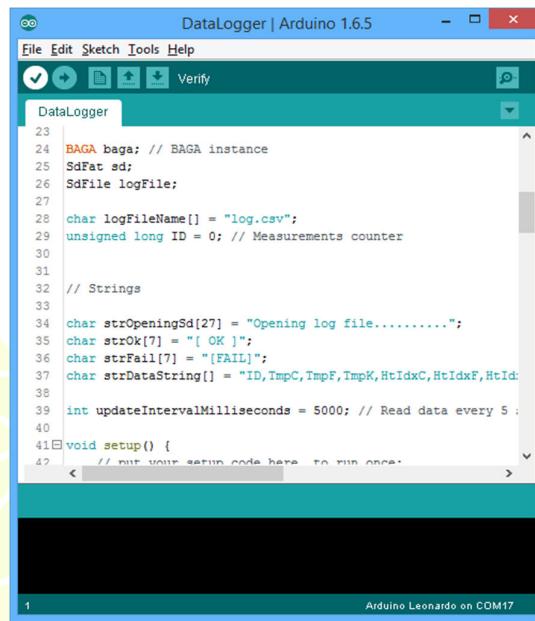
File Edit Sketch Tools Help
DataLogger | Arduino 1.6.5
#include <Wire.h>
#include <BMP280.h>
#include <S17020.h>
#include <BAGA.h>
#include <SPI.h>
#include <SdFat.h>
/**
 * BAGA A1
 * Data Logger Example
 *
 * This example reads data periodically and saves it to the MicroSD card.
 *
 * HW Compatibility: A2
 */
// Select the desired debug serial port:
// * Serial (USB-CDC)
// * Serial1 (J3 connector on board)

```

Arduino Leonardo on COM17

Figura 10 - Sketch DataLogger

A variável **logFileName** define o nome do ficheiro onde os dados serão guardados.



```

23
24 BAGA baga; // BAGA instance
25 SdFat sd;
26 SdFile logFile;
27
28 char logFileName[] = "log.csv";
29 unsigned long ID = 0; // Measurements counter
30
31 // Strings
32
33 char strOpeningSd[27] = "Opening log file.....";
34 char strOk[7] = "[ OK ]";
35 char strFail[7] = "[FAIL]";
36 char strDataString[] = "ID,TmpC,TmpF,TmpK,HtIdxC,HtIdxF,HtId:
37
38 int updateIntervalMilliseconds = 5000; // Read data every 5 :
39
40
41 void setup() {
42     // put your setup code here - to run once:

```

Figura 11 - Sketch DataLogger

Outros exemplos

As bibliotecas BMP280 me Si7020 disponibilizam também alguns exemplos para uso individual dos respetivos sensores, e podem ser encontrados nos devidos submenus em **File >> Examples**.