

Universidade Federal da Bahia

Componentes:

Andressa Andrade

Nailane Oliveira

Renata Antunes

MANUAL INTEL GALILEO

Salvador - 2016

Sumário

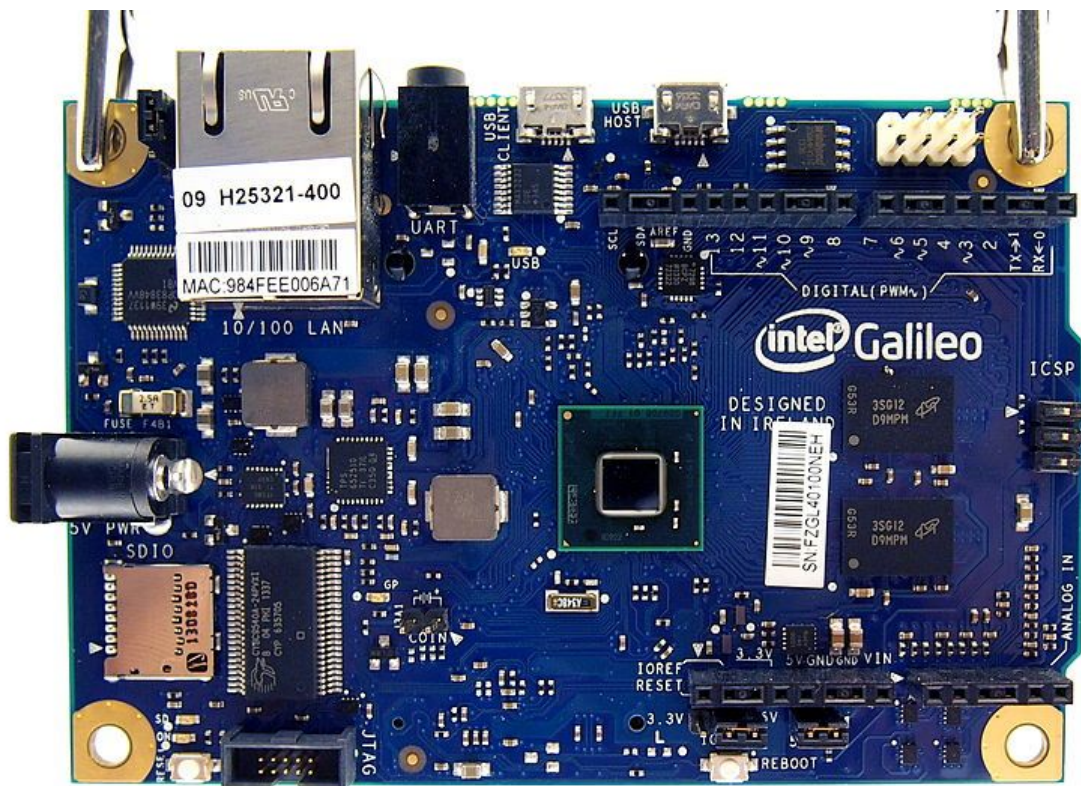
Introdução

Prefacio

Caro leitor, este manual tem como objetivo apresentar de forma dinâmica e intuitiva o grande potencial da placa Intel Galileo geração 2. Neste material iremos abordar as principais características da placa, combinando a sua facilidade de manipulação e programação. Através deste manual será possível desenvolver os conhecimentos técnicos, desde pinagem à processador, desfrutando todo o poder do Intel Galileo 2ª Geração, onde você encontrará explicações detalhadas de alguns exemplos, juntamente com alguns exercícios, tornando-se capacitado para utilizá-la sozinho.

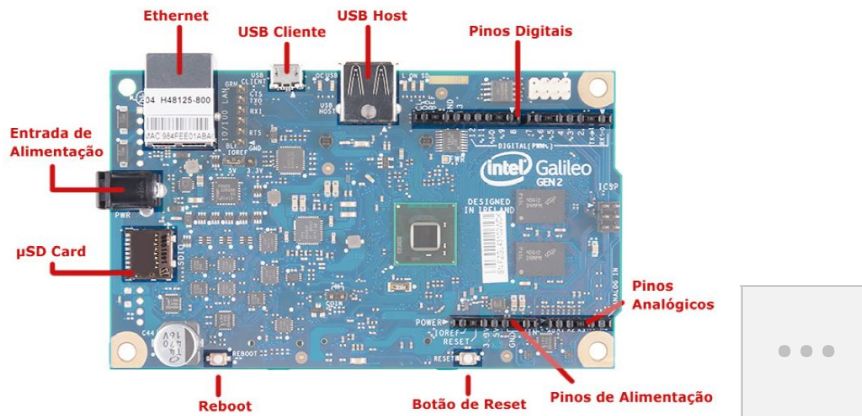
Introdução

O Intel® Galileo 2ª Geração é uma placa de microcontrolador baseado no processador de aplicação Quark SoC X1000 da Intel®, um sistema em um chip (system on a chip) da classe Pentium de 32 bits. É a segunda geração da primeira placa baseada na arquitetura Intel® desenvolvida para ter pinagem compatível, tanto em software como em hardware, com os shields Arduino desenhados para o Uno R3. É possível programar utilizando a IDE do Arduino ou podemos habilitar ssh e usar linhas de comando para programar em linguagens de alto nível, como Python e C. Uma diferença relevante entre as duas gerações é a fonte de alimentação, na primeira versão era 5V, na segunda é de 12V.



Intel Galileo 2ª Geração

1. Conhecendo a placa:



a. Características físicas

- i. Botão de reset
- ii. Botão de reboot
- iii. Pinos de alimentação
- iv. Pinos analógicos
- v. Micro SD Card
- vi. Entrada de Alimentação
- vii. Ethernet
- viii. USB Cliente
- ix. USB Host
- x. Pinos Digitais

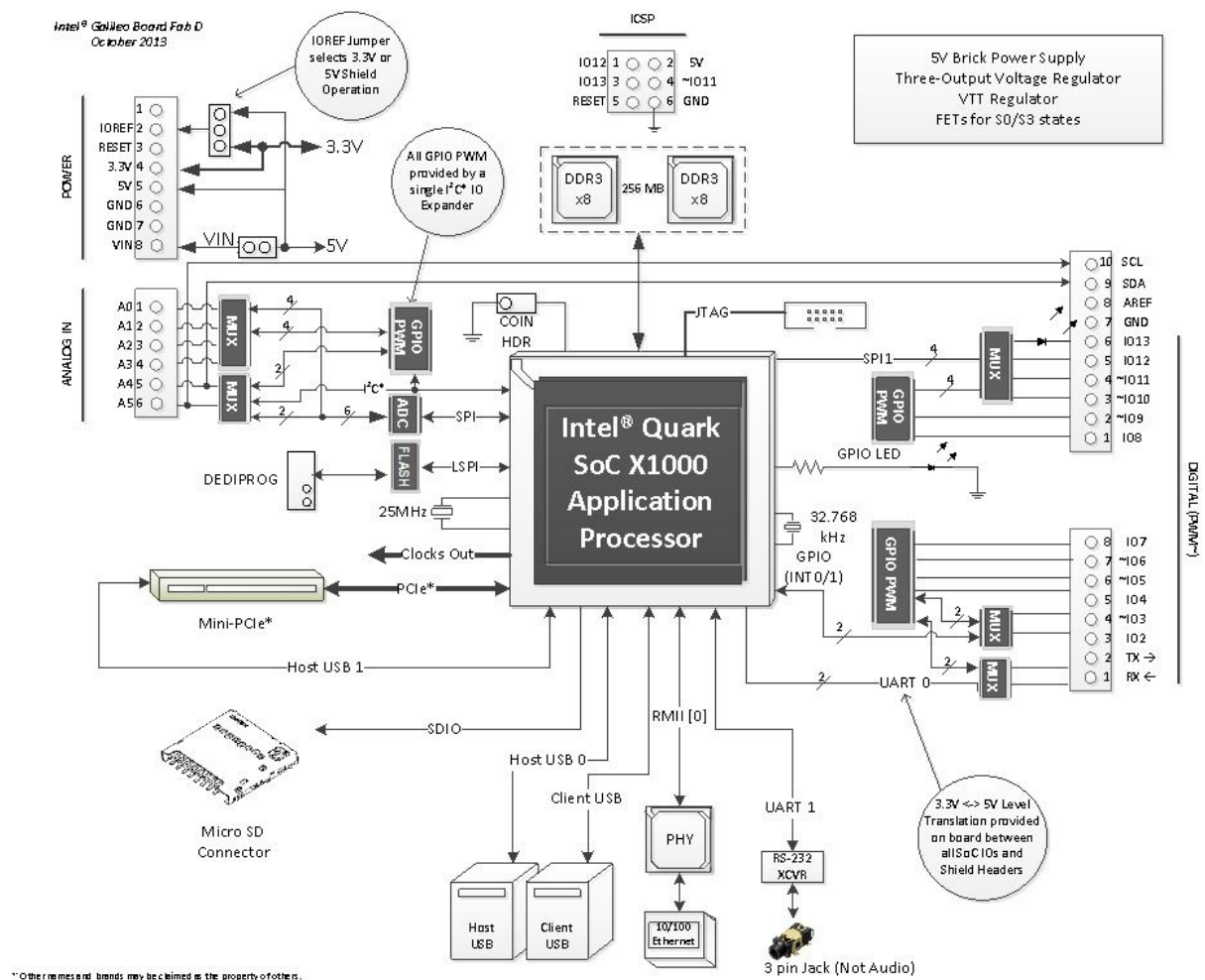
b. Características do processador

- i. Processador Intel Quark SoC X1000
- ii. 16 Kbytes L1 Cachê
- iii. 512 Kbytes SRAM embarcada
- iv. Velocidade de clock 400 MHz
- v. Relógio de tempo real (RTC) integrado com tipo moeda de 3V
- vi. Estados de hibernação de CPU compatível com ACPI

c. Opções de armazenamento

- i. 512 Kbytes SRAM embarcada
- ii. 256 MB de DRAM
- iii. 8 MB de Flash SPI Legacy para armazenar o firmware (bootloader) e a última sketch
- iv. Entre 256 e 512 KB dedicados para armazenagem de sketch
- v. Cartão micro SD opcional pode oferecer até 32 GB de armazenamento
- vi. Armazenamento USB compatível com qualquer drive USB 2.0
- vii. 11 kB de EEPROM programáveis com a biblioteca EEPROM

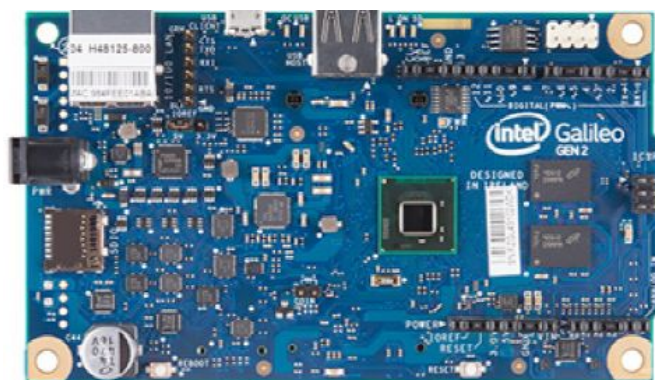
Especificação da placa



Alimentação e conexão da placa

2. Instalando o Software

Para a instalação você precisará da fonte de alimentação 12V que acompanha a placa e de um cabo USB microB que não acompanha o kit.



Fazer o download da última versão do Software do Arduino para o Galileo compatível com o sistema operacional do seu computador. O download pode ser realizado no site:

<https://www.arduino.cc/>

Dentro do arquivo que foi baixado, executar o arquivo arduino.exe que é o ponto de entrada do programa Arduino, a IDE. Clicando duas vezes e o software está instalado.

3. Instalando os drivers

Conecte a placa Intel Galileo no computador e aguarde o Windows iniciar o processo de instalação dos drivers necessários.

Ao abrir a opção para instalação dos drivers, selecionar “Procurar software de driver no computador”.

Dentro da pasta em que o software do Arduino foi instalado, selecionar a mesma pasta para a busca dos drivers serem realizadas, incluindo subpastas e avançar.

Após realizar algumas confirmações, clicar em Instalar.

Dentro de alguns instantes, o Windows concluirá a instalação e a placa estará disponível para uso.

4. Preparando a placa na IDE

Apesar de ter instalado a IDE do Arduíno, juntamente com os drivers, ainda é necessário selecionar a placa dentro do software para dar início aos exercícios e colocar a mão na massa no código.

Executando o arquivo arduino.exe, selecione a porta serial que conectarpa o Galileo em: Ferramentas > Porta Serial. Você deverá selecionar o mesmo número da porta COM em que a placa foi definida no sistema

Depois, deve selecionar qual a sua placa:

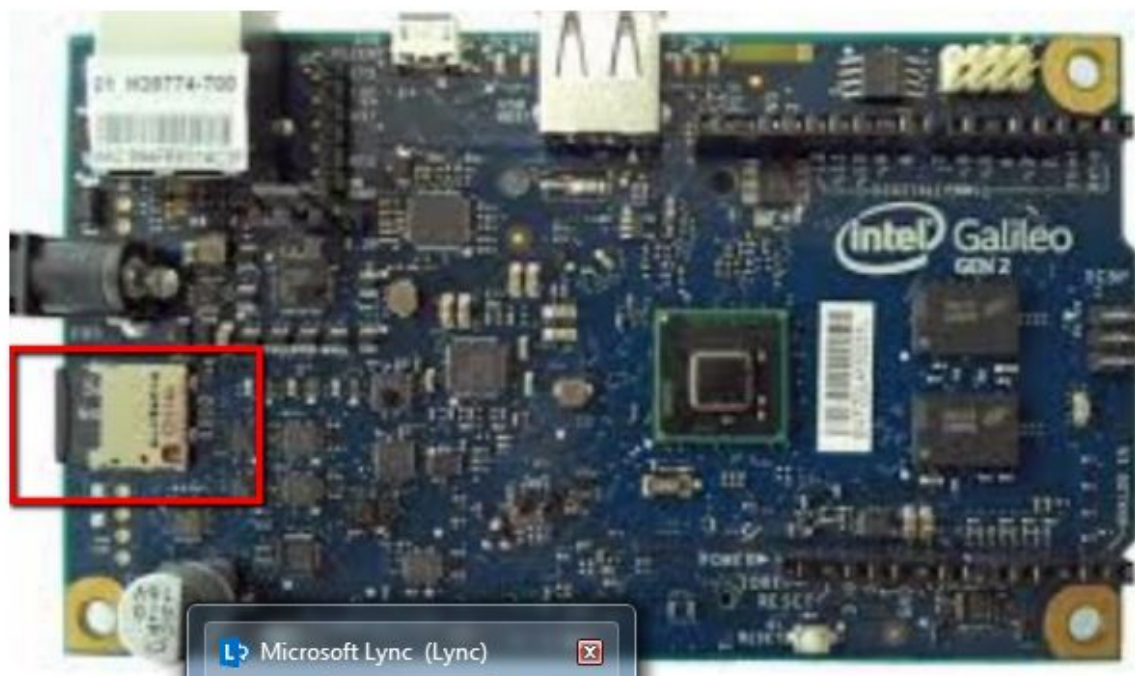
Ferramentas > Placa > Intel Galileo Gen 2

Nota: Para mais informações como descobrir o número da porta COM da placa, você deve ir em Painel de Controle > Gerenciador de Dispositivos > Portas (COM e LPT)



5. Boot utilizando o SD Card (achei interessante colocar um exemplo de como dar o boot com qualquer SD e depois falamos sobre o boot com Linux)

Para dar o boot com o SD Card, é necessário seguir os seguintes passos:



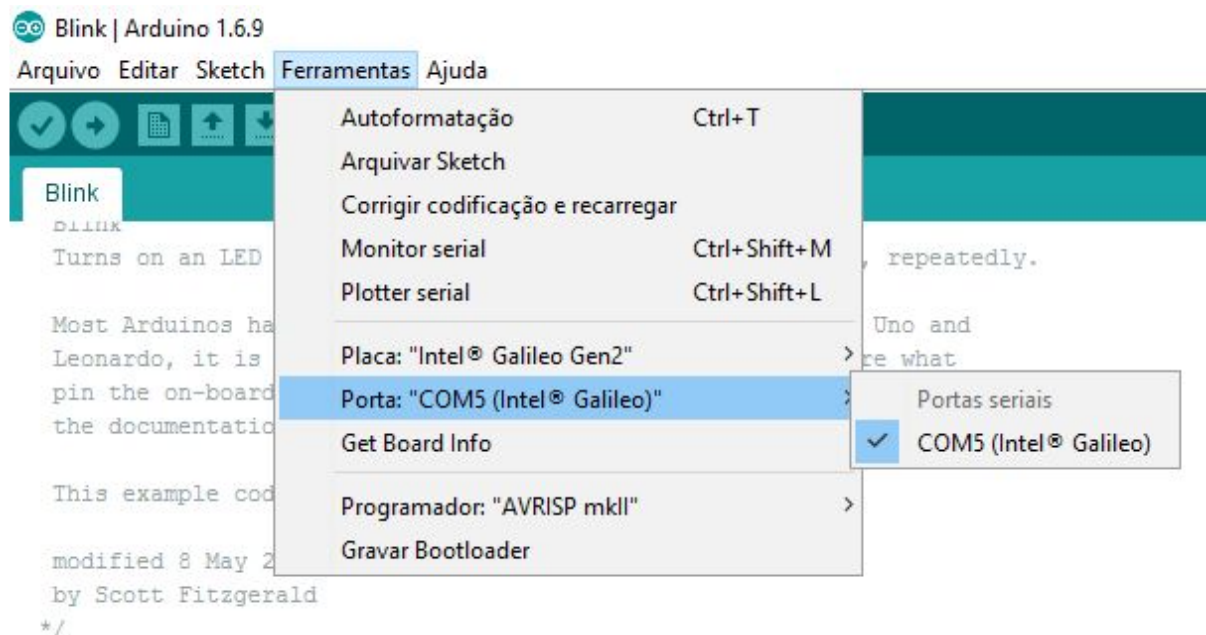
1. Coloque o seu SD Card no slot disponível na placa
2. Conecte a fonte

3. Espere a led do USB ligar. Logo depois, a led do SD deve piscar também. Ela indica que o cartão está sendo lido.
4. Conecte a placa ao computador utilizando um cabo USB.

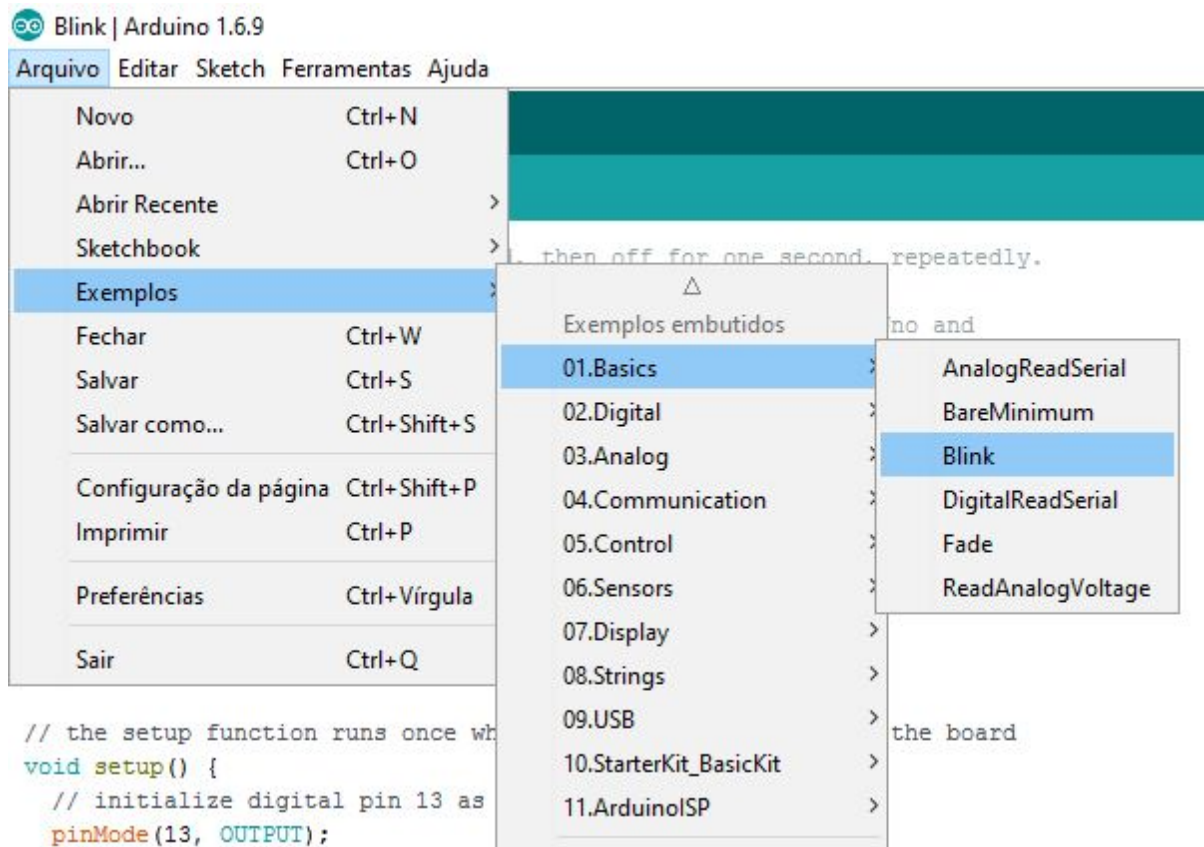
1. Exemplo Blink - O "Hello World" das placas

Vamos iniciar com o exemplo mais fácil de código,

Primeiro precisamos configurar a IDE:



Depois carregamos um exemplo chamado Blink que já vem na IDE



```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
```

2. Exemplo utilizando um botão para acender e apagar um LED, Portas Digitais

É possível utilizar a Placa Intel Galileo Gen 2, para atividades que poderiam ser feitas em um arduino, então pode se pesquisar projetos em Arduino e reproduzir estes nas Placas Intel Galileo. Neste exemplo, é o acionamento de LED utilizando um botão. Montamos um circuito assim:

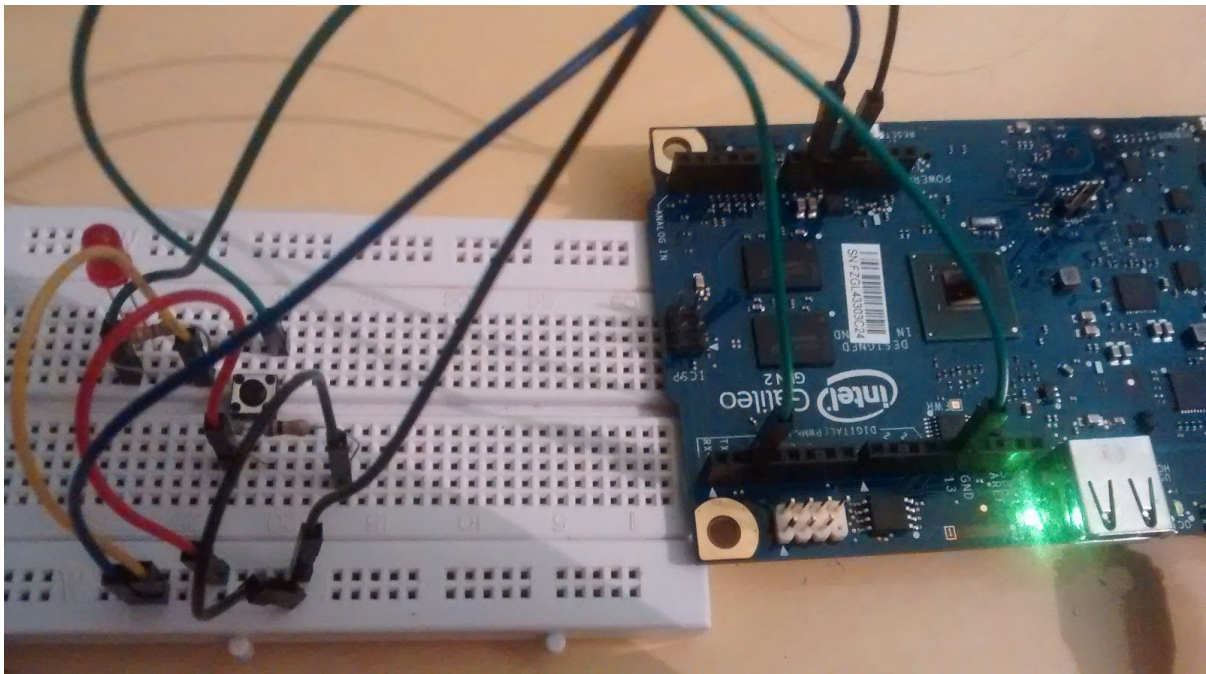
Código:

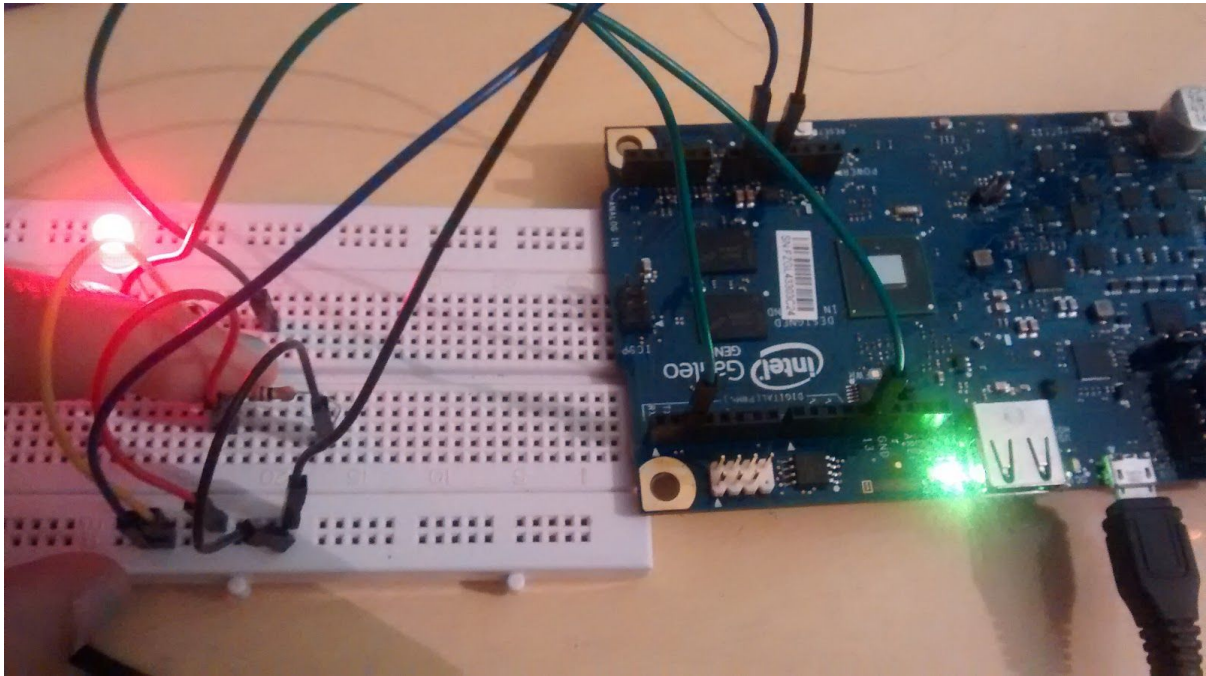
```
// constantes nao sao alteradas.
// Sao usadas aqui para definir os numeros dos pinos:
const int buttonPin = 2;
```

```

// o numero do pino do botão
const int ledPin = 13;
// o numero do pino do LED
// variaveis que devem mudar:
int buttonState = 0;
// variavel para ler o estado do botao
void setup() {
    // inicializa o pino do LED como saida:
    pinMode(ledPin, OUTPUT);
    // inicializa o pino do botao como entrada:
    pinMode(buttonPin, INPUT);
}
void loop(){
    // faz a leitura do valor do botao:
    buttonState = digitalRead(buttonPin);
    // verifica se o botao esta pressionado.
    // em caso positivo, buttonState e HIGH:
    if (buttonState == HIGH) {
        // liga o LED:
        digitalWrite(ledPin, HIGH);
    }
    else {
        // desliga o LED:
        digitalWrite(ledPin, LOW);
    }
}

```





LED acionado ao clicar o botão

3 - Exemplo Utilizando Portas Analógicas

/* Entrada Analogica, Saida Analogica, Saida serial

Le o pino de entrada analogica, mapeia o resultado para um intervalo entre 0 e 1023 e usa o resultado para estabelecer o pulso PWM do pino de saida.

Tambem e possivel acompanhar o resultado atraves do Monitor Serial.

O circuito: - O pino central do Potenciometro conectado ao pino analogico 0.

Os pinos laterais do potenciometro conectados no terra e 5V. - LED conectado no pino digital 9 e no terra.

Criado em 29/12/2008, Modificado em 09/04/2012 por Tom Igoe Este exemplo e de dominio publico. */

// constantes nao sao alteradas:

const int analogInPin = A0;

// Entrada analogica do potenciometro

const int analogOutPin = 9;

// Saida analogica onde o LED esta conectado

int sensorValue = 0;

// leitura do potenciometro

int outputValue = 0;

// leitura da saida PWM (analogica)

void setup() {

 // inicializa a comunicacao serial:

 Serial.begin(9600);

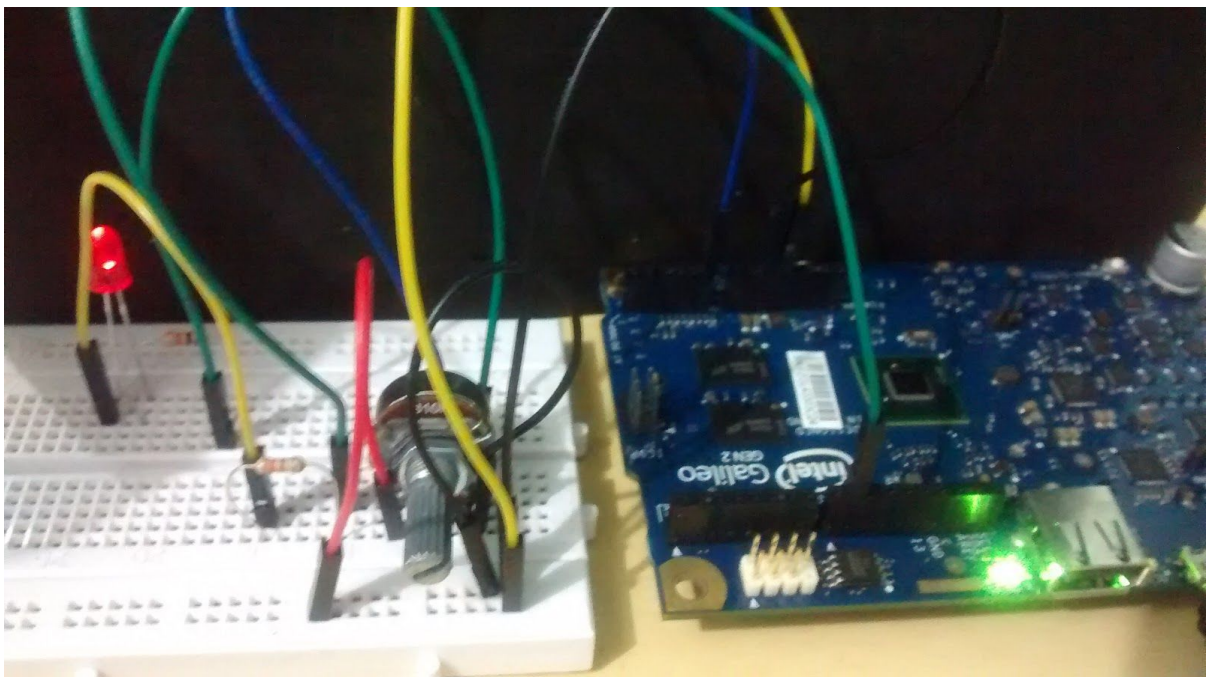
}

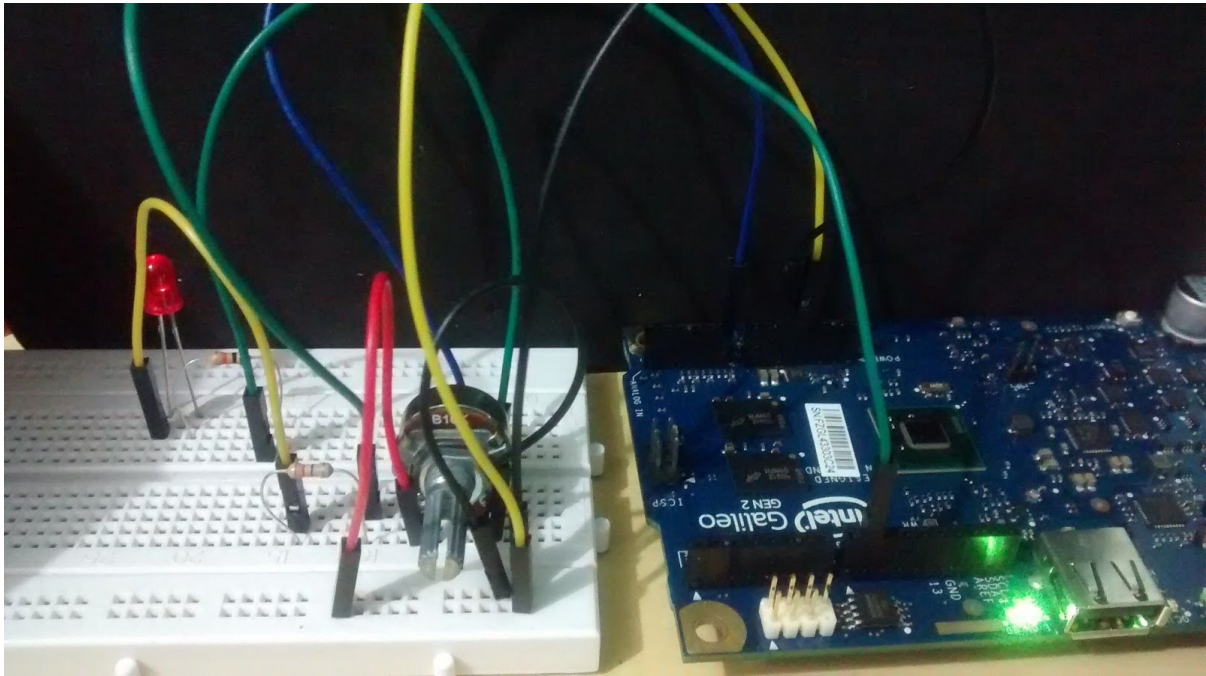
void loop() {

 // faz a leitura da entrada analogica:

 sensorValue = analogRead(analogInPin);


```
// mapeia o resultado da entrada analogica dentro do intervalo de 0 a 1023:  
outputValue = map(sensorValue, 0, 1023, 0, 255);  
// muda o valor da saida analogica:  
analogWrite(analogOutPin, outputValue);  
// imprime o resultado no monitor serial:  
Serial.print("sensor = " );  
Serial.print(sensorValue);  
Serial.print("\t output = ");  
Serial.println(outputValue);  
// Aguarda 2 milissegundos antes do proximo loop:  
delay(2);  
}
```





Ao girar o potenciômetro o LED terá seu brilho aumentado ou diminuído.

4 - Exemplo Linux e Sketches permanentes

Faça o download de uma imagem Linux para o Galileo.

Você pode encontrar a imagem aqui:

<https://software.intel.com/en-us/iot/hardware/galileo/downloads#>

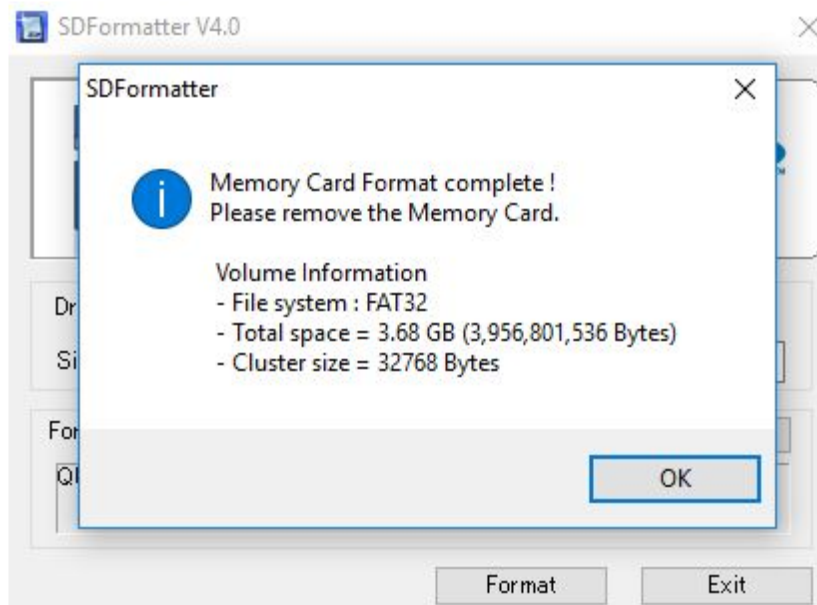
MicroSD* Card Image for Linux* Operating System

Using the microSD card image enables the Wi-Fi* drivers and sketch persistence. The libraries to use with either Eclipse* or the Intel® XDK are included in the EGLIBC-based image.

[Latest Yocto* Poky Image](#) (210 MB)

Preparing your microSD card for [Windows*](#), [OS X*](#), and [Linux*](#)

Tenha um cartão micro SD de pelo menos 1gb, descompacte o arquivo e copie todo o conteúdo para um cartão SD formatado. Um software indicado para a formatação pode ser encontrado nesse link: https://members.sdcard.org/downloads/formatter_4/eula_windows/
A instalação é simples, após basta selecionar a unidade do cartão e clicar em "Format"

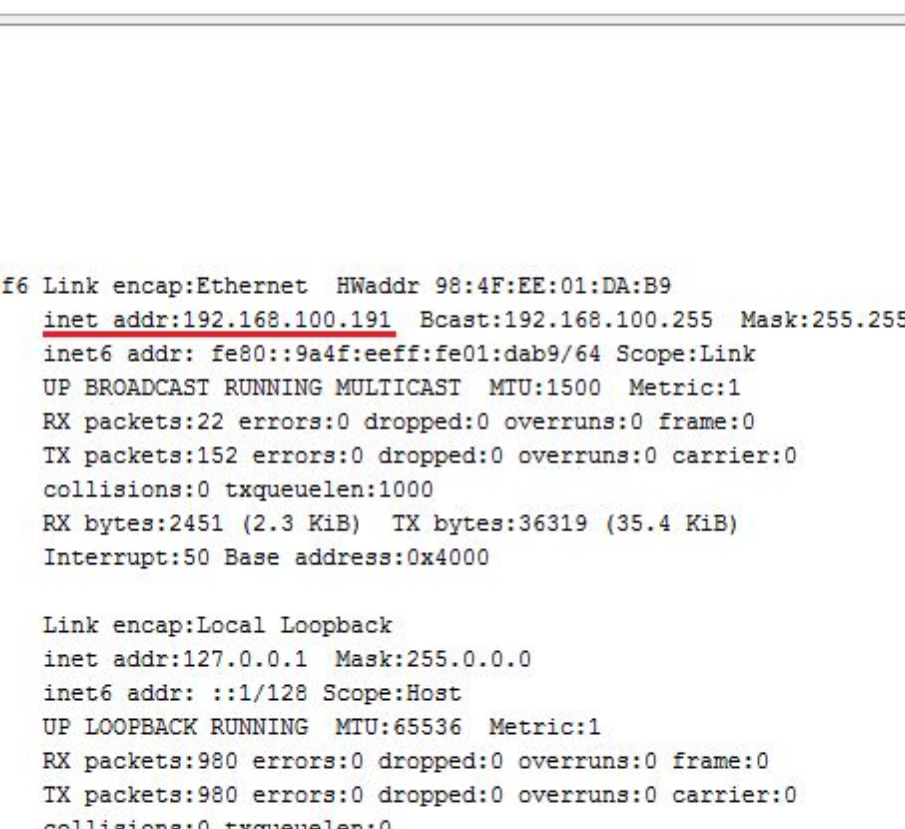


Essa tela será exibida

Desligue o Galileo (desconecte o cabo USB e após a fonte). coloque o cartão no slot microSD e ligue novamente a placa. Aguarde alguns minutos para que o Linux seja carregado, observando a atividade do led do SD que fica à direita do conector USB, ele pisca enquanto o cartão está sendo acessado.

Desta forma, será possível acessar o linux pelo cabo Ethernet
Primeiro passo é descobrir o endereço IP, para isso inserimos o cabo Ethernet e utilizamos um código no arduino:

```
void setup() {  
    Serial.begin(9600);  
    //It'll not continue until you send an 'a' through the serial monitor  
    while(Serial.read()!='a'){  
        Serial.println("hello");  
        delay(1000);  
    }  
    //Display ifconfig result to serial monitor  
    system("ifconfig > /dev/ttyGS0");  
  
    void loop() {  
    }  
}
```



```
COM6

|
Enviar

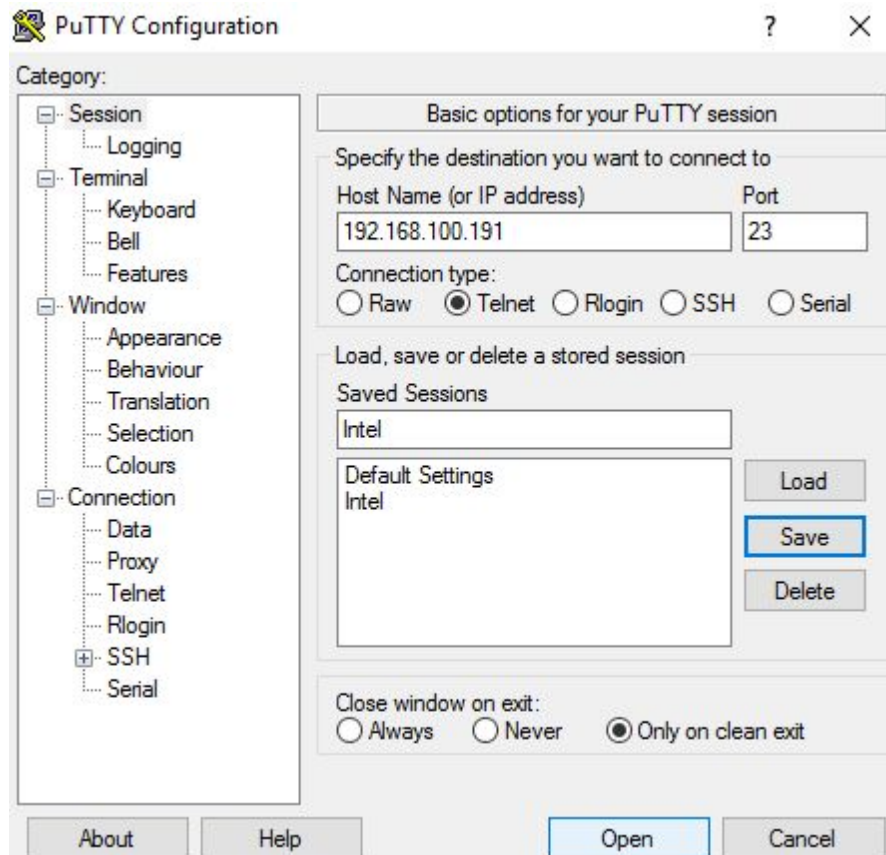
hello
hello
hello
hello
hello
hello
hello
hello
enp0s20f6 Link encap:Ethernet  HWaddr 98:4F:EE:01:DA:B9
    inet addr:192.168.100.191  Bcast:192.168.100.255  Mask:255.255.255.0
    inet6 addr: fe80::9a4f:eeff:fe01:dab9/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
    RX packets:22 errors:0 dropped:0 overruns:0 frame:0
    TX packets:152 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:2451 (2.3 KiB)  TX bytes:36319 (35.4 KiB)
    Interrupt:50 Base address:0x4000

lo
    Link encap:Local Loopback
    inet addr:127.0.0.1  Mask:255.0.0.0
    inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING  MTU:65536  Metric:1
    RX packets:980 errors:0 dropped:0 overruns:0 frame:0
    TX packets:980 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:71200 (69.5 KiB)  TX bytes:71200 (69.5 KiB)

☒ Auto-rolagem
Nenhum final-de-linha
9600 velocidade
```

O próximo passo é instalar o Putty que pode ser encontrado aqui:
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Basta seguir as instruções. Ao iniciar o PuTTY esta tela será exibida:



Em Host Name(or IP address) copie o endereço sublinhado na etapa anterior, o tipo da conexão é Telnet e a porta 23. É possível salvar a configuração para outras ocasiões, digite o nome que deseja em “Saved Sessions” e clique “Salvar”, na próxima vez que acessar, basta selecionar o nome salvo e clicar em “Load”. Clique em “Open”.

testando a imagem do sd: <https://software.intel.com/en-us/get-started-galileo-windows-step3>
<http://www.intel.com/content/www/us/en/support/boards-and-kits/intel-galileo-boards/000021501.html>

Referencias

<http://www.ft.unicamp.br/lapet/wp-content/uploads/Estudo-da-placa-de-prototipagem-Intel-Galileo-Gen2.pdf>

https://www.thingworx.com/wp-content/uploads/academic-program_weather-app-galileo_configure.pdf

<http://blog.filipeflop.com/embarcados/python-linux-no-intel-galileo.html>