



# Galileo Gen2

Walter Fetter Lages

fetter@ece.ufrgs.br

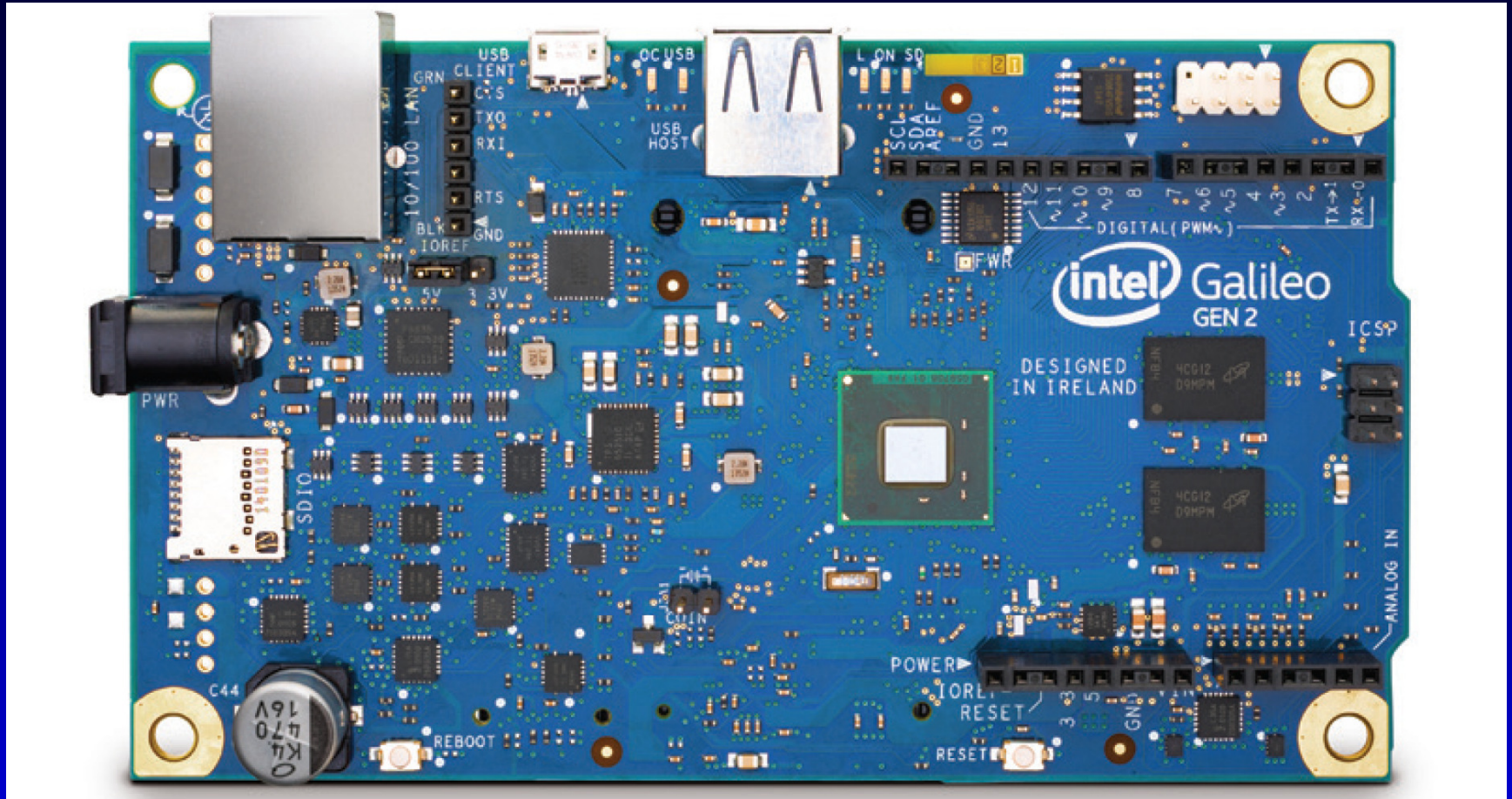
Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Sistemas Elétricos de Automação e Energia

ENG10032 Microcontroladores

# Galileo Gen2





# Galileo Gen2

- Intel Galileo Gen2 Development Board
- Intel Quark SoC X1000
  - Pentium de 32 bits @ 400MHz
- *Open Source Hardware*
- Compatível com Arduino Uno R3
  - 20 I/O digitais (12 nativas)
  - 6 entradas analógicas
  - 6 PWMs com resolução de 12 bits
  - 1 mestre SPI
  - 2 UARTs (1 compartilhada com console)
  - 1 mestre I2C



# Conexões

- *Shield* Arduino
- UART FTDI 3.3V
- ICSP
- JTAG
- Ethernet
- USB 2.0 host
- USB 2.0 client
- Mini-PCI Express 1x
- Jack de alimentação



# Alimentação & Reset

- Jack 7 a 15 V
- *header* para alimentação do RTC
- Botão de *reset* para *shields* e Ethernet
- Botão de *reboot* para o Quark

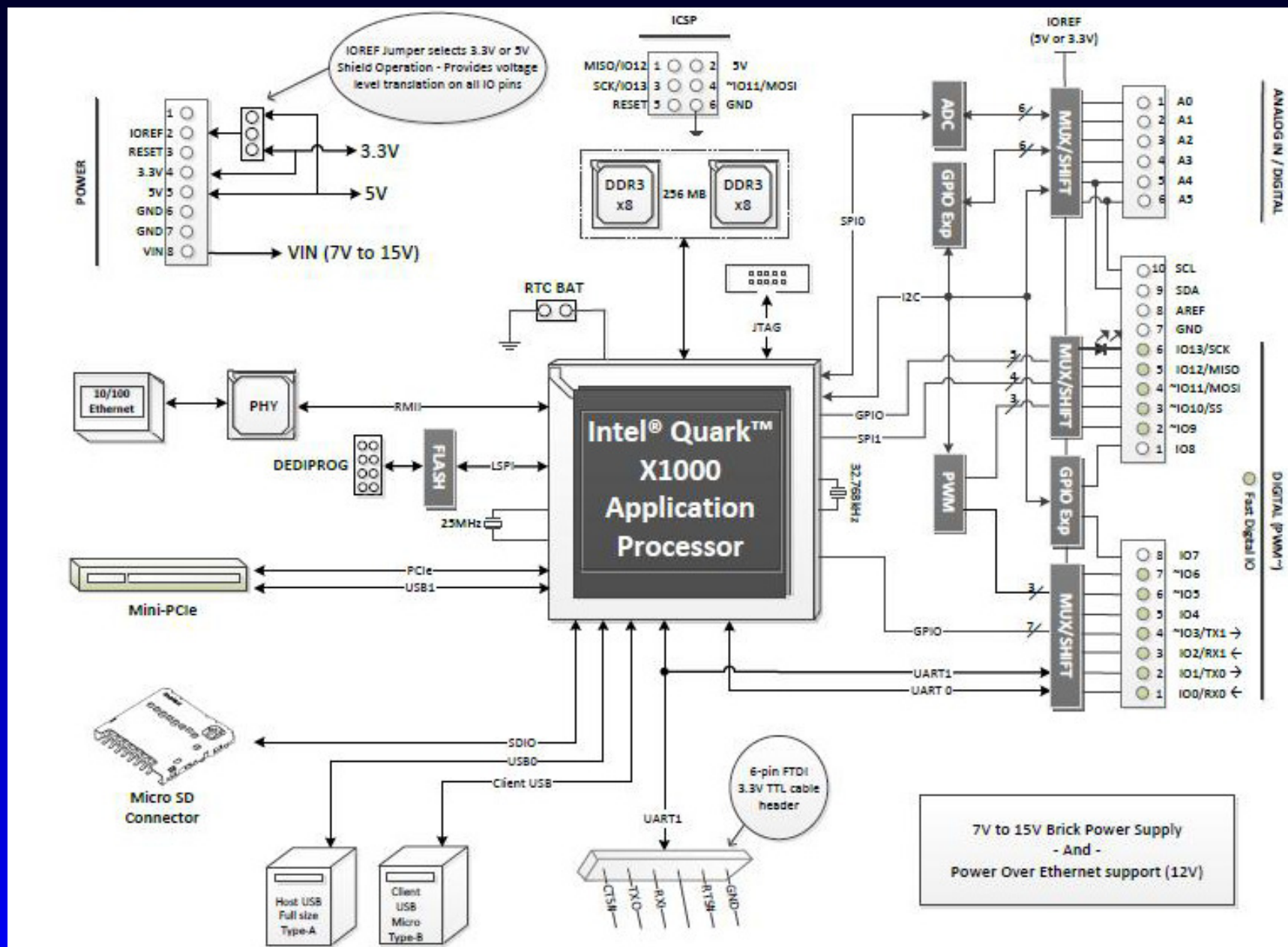


# Armazenamento

- 8 MB Flash NOR
- 256 MB DDR3 @ 800 MT/s
- SD Card até 32 GB
- USB
- 8 kb EEPROM



# Diagrama de Blocos



# UART de Console do Galileo Gen2

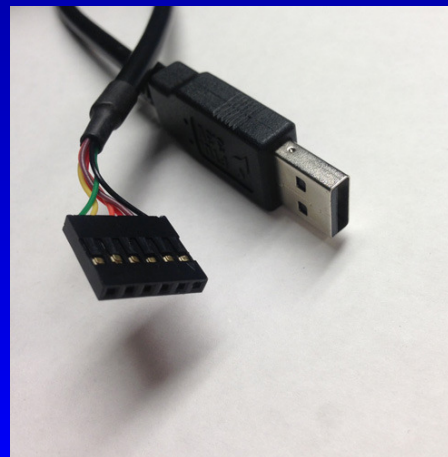
- Sinais RS-232, porém com níveis TTL 3.3 V
- Emula um DTE
- GND, #RTS, NC, RXI, TXO, #CTS
- É diferente do Galileo
  - *Plug* de 3,5 mm, TX, RX, GND
- No Linux aparece como `/dev/ttyS1`





# Cabo FTDI TTL-232R-3V3

- Conversor USB para serial
- Sinais RS-232, porém com níveis TTL 3.3 V
- Existem versões de 5 V e 3.3 V
- Emula um DTE
- GND, #CTS, 5V, TXD, RXD, #RTS
- Conectado no Galileo Gen2 cria um null-modem
- No *host* aparece como `/dev/ttyUSB?`



# Conversor USB-Serial

- Sinais RS-232, porém com níveis TTL 3.3 V
- Existem versões de 5 V e 3.3 V selecionáveis
- Emula um DTE
- GND, #CTS, 3.3V, TXD, RXD, #DTR
- Conectado no Galileo Gen2 cria um null-modem, embora com sinalização #DTR/#CTS
- No *host* aparece como `/dev/ttyUSB?`





# Notas

- Durante o *boot* a Galileo Gen2 desativa o #RTS
  - Para usar o menu do GRUB, é necessário desativar o controle de fluxo por *hardware* no emulador de terminal

# *Boot da Galileo Gen2*

- A Galileo Gen2 possui um *kernel* Linux e um *initrd* na *Flash*
- O sistema de arquivos é RAMFS
- Não é persistente entre *boots*
- Para alterações nas configurações ou no *kernel* serem persistentes é necessário criar um novo *initrd* e gravá-los na *flash*
- Felizmente a Galileo Gen2 também podem dar *boot* a partir do cartão micro SD ou de *pendrive*
  - A imagem disponível no site da Intel foi feita para cartão micro SD e não funciona bem com *pendrive*

# Criação do Cartão SD

- Identificar o dispositivo do cartão SD:

```
fdisk -l | grep Disk
```

- Normalmente o dispositivo é `/dev/sdb` quando é usado um adaptador USB ou `/dev/mmcblk0` quando é usado um *slot* SD
- O dispositivo varia dependendo dos dispositivos de armazenamento de massa existentes na máquina
- Baixar a imagem de

[https://downloadmirror.intel.com/26418/eng/Galileo\\_Poky\\_SW\\_image\\_20160606.zip](https://downloadmirror.intel.com/26418/eng/Galileo_Poky_SW_image_20160606.zip)

# Criação do Cartão SD

- Gravar imagem no cartão:

```
unzip -p Galileo_Poky_SW_image_20160606.zip | \  
dd of=/dev/sdb bs=3M conv=fsync status=progress
```

ou

```
unzip -p Galileo_Poky_SW_image_20160606.zip | \  
dd of=/dev/mmcbk0 bs=3M conv=fsync status=progress
```

- São criadas duas partições no cartão, uma FAT32 para os arquivos de *boot* e outra ext3 para o /
- Inserir o cartão na Galileo Gen2
- Dar *boot*





# Acesso ao FS do Galileo em um PC

- É possível montar o sistema de arquivos do Galileo em um PC com Linux
- É possível também executar um `chroot` para este sistema de arquivos e executar programas que estão lá
- Útil para desenvolvimentos em casa, sem acesso ao Galileo
- Se o cartão for conectado através de um adaptador USB:  

```
mount -t ext3 /dev/sdb2 /mnt
```
- Se o cartão for conectado através de um *slot* SD:  

```
mount -t ext3 /dev/mmcb1k0p2 /mnt
```