

Trabalho 1 - Sistemas Operacionais Embarcados

Carlos Marcelo Pedroso

*Universidade Federal do Paraná,
Departamento de Engenharia Elétrica*

E-mail: pedroso@eletrica.ufpr.br

ABSTRACT: Este documento especifica atividades a serem desenvolvidas para disciplina de Sistemas Operacionais Embarcados do curso de Engenharia Elétrica com Ênfase em Sistemas Embarcados da Universidade Federal do Paraná. Esta atividade compreende o uso da placa Intel Galileo (<http://www.intel.com/content/www/us/en/embedded/products/galileo/galileo-g1-datasheet.html>), que será usada para a formação de competências na área de gerência de processos. A Intel Galileo usa o sistema operacional Yocto (<https://www.yoctoproject.org/>), que é uma versão do Linux bastante usada em sistemas embarcados. Deverá ser apresentado um relatório respondendo as questões solicitadas, bem como código fonte em linguagem C em sistemas Unix. O formato do relatório está disponível na página da disciplina, bem como um *template* em L^AT_EX.

Sumário

1	Iniciando com a Intel Galileo Gen 1	1
1.1	Começando a programar	1
2	Descrição do Trabalho	3
2.1	Fase 1	3
2.2	Fase 2	3
2.3	Fase 3	3
2.4	Como será a avaliação	3
3	Critérios de avaliação	4

1 Iniciando com a Intel Galileo Gen 1

Atenção: a sequência de ligação da Galileo deve ser: 1) Conectar a fonte de alimentação e 2) Conectar a USB. *Proceder de outro modo irá queimar a placa.*

As placas foram fornecidas sem fonte de alimentação. A Galileo usa uma fonte de 5V, que pode ser obtida através de uma adaptação usando um carregador de telefone celular e um cabo USB modificado. Para fazer o cabo modificado, corte o lado do cabo que seria conectado no telefone celular e identifique os pinos 1 e 4 (respectivamente vermelho e preto). O pino vermelho é +5V DC. O pino preto é o terra. Será necessário um conector de alimentação tipo jack macho com diâmetro externo de 5.5mm e interno de ... A parte externa do conector de alimentação deve ser conectada ao terra e parte interna deve ser conectada o +5V DC. Certifique-se da montagem antes de alimentar a placa.

1.1 Começando a programar

A Galileo suporta a programação no estilo Arduino (mas não apenas). Para iniciar com o estilo Arduino, instale o Arduino Intel IDE - disponível em <https://software.intel.com/en-us/get-started-arduino-install>. Após instalar, ligue a Galileo e rode os diversos programas de teste disponíveis (inicie com <https://software.intel.com/en-us/get-started-arduino-blink>).

No estilo Arduino, o programa possui duas funções:

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:
```

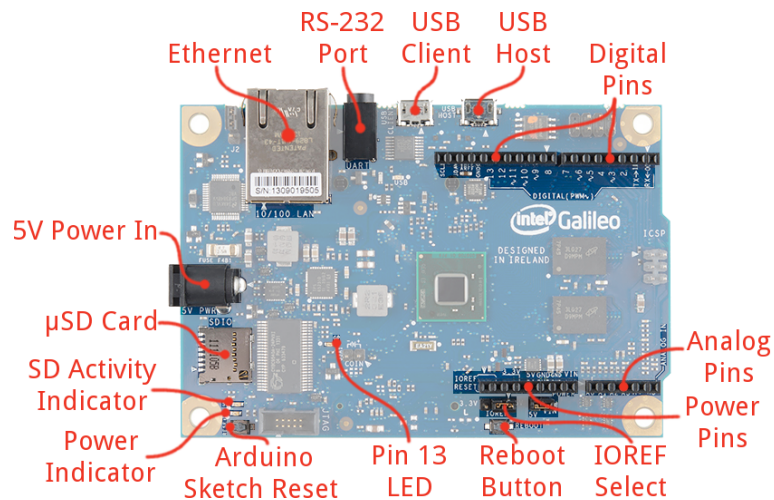


Figura 1. Diagrama da Intel Galileo Gen 1. fonte: <https://learn.sparkfun.com/tutorials/galileo-getting-started-guide>

}

A Intel Arduino IDE possui opções para compilar e transferir o programa para o Galileo. Anote as observações da aula.

Algumas referências sobre o estilo de programação Arduino:

- <http://web.stanford.edu/class/me20n/week3/Arduino%20Programming%20Language.pdf>
- <http://www.playrobot.com/robotpress/wp-content/uploads/2015/08/Arduino-Reference.pdf>
- <http://www.hfremote.us/files/Arduino.pdf>

No entanto, o Galileo não é só um Arduino. Ele possui um Linux embarcado. Escreva o seguinte programa:

```
void setup() {
  system("ifconfig eth0 192.168.1.200 netmask 255.255.255.0 up");
  system("telnetd -l /bin/sh");
}

void loop() {
}
```

A função *system* irá executar uma linha de comando, como você estivesse digitando comandos no terminal. Neste caso, o *ifconfig* irá configurar o endereço IP da interface. O *telnetd* é o servidor de comandos remotos, que aguarda conexão na porta 23. Substitua o

endereço por um endereço de sua rede e teste o programa. Usando um terminal, acesse o sistema operacional embarcado usando o cliente telnet. Confira os processos em execução (comando *ps*, *ps tree* e *top*). Confira a organização do sistema de arquivos (comando *ds -h*, *ls -lh*) e navegue no sistema de arquivos. Verifique como o Galileo imita um Arduino - no entanto, ele é muito mais que um Arduino.

2 Descrição do Trabalho

O trabalho está dividido em três fases.

2.1 Fase 1

O Galileo deve monitorar a temperatura e a luminosidade do ambiente. Deve ser implementado um servidor Web no Galileo, aguardando conexões na porta 80, que pode ser acessada por um Cliente Web em qualquer lugar na Internet, onde a leitura dos valores deve ser continuamente atualizada. A página deve indicar o horário da leitura e o número da leitura em relação às leituras anteriores.

2.2 Fase 2

Os valores lidos de temperatura e luminosidade devem ser enviados para um servidor, que irá armazená-los em um arquivo texto. Para isso, deve ser desenvolvido um programa cliente TCP que irá enviar o arquivo ao servidor, e um programa servidor deve ser desenvolvido utilizando a biblioteca Sockets para receber os dados e armazenar em um arquivo texto. O arquivo texto deve possuir uma linha para cada leitura, sendo a primeira coluna para hora, a segunda para a temperatura e a terceira para luminosidade. Deve ser desenvolvido um protocolo para transmissão dos dados.

2.3 Fase 3

As funções desenvolvidas nas Fases 1 e 2 devem funcionar simultaneamente. O servidor Web deve atender um número arbitrário de conexões simultâneas. O sistema deve ser configurado de tal forma que o processo que envia os dados para o servidor seja mais prioritário do que o servidor Web embarcado.

2.4 Como será a avaliação

1. Apresentação do projeto para o professor.
2. Relatório com uma revisão sobre os recursos disponíveis no Intel Galileo Gen. 1 e sistema operacional Yocto. Na sequência, o relatório deverá explicar como cada uma das fases foi implementada, explicando a solução adotada, com o código fonte e explicações.

3 Critérios de avaliação

O trabalho deve ser implementado em equipes de até 2 pessoas. Não serão admitidos trabalhos desenvolvidos por equipes maiores. A avaliação será realizada da seguinte considerando os seguintes critérios:

1. Qualidade do relatório: 10 pontos;
2. Funcionamento da Fase 1: 30 pontos;
3. Funcionamento da Fase 2: 30 pontos;
4. Funcionamento da Fase 3: 30 pontos;

O funcionamento parcial será pontuado proporcionalmente em cada fase. Em caso de cópias, as equipes envolvidas terão grau zero.