Trabalho 2 - 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, que será usada para a formação de competências na área de gerência de sistemas operacionais embarcados. A Intel Galileo usa o sistema operacional Yocto (https://www.yocto project.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 LaTeX.

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 Observações 4
2.5 Como será a avaliação 4

3 Critérios de avaliação **4**

4 Datas **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.
Datasheet: http://www.intel.com/content/www/us/en/embedded/products/galileo/galileo-g1-datasheet.html

1.1 Começando a programar

A Galileo suporta a programação no estilo Arduíno (mas não apenas). Para iniciar com o estilo Arduíno, instale o Arduíno 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 Arduíno, o programa possui duas funções:

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

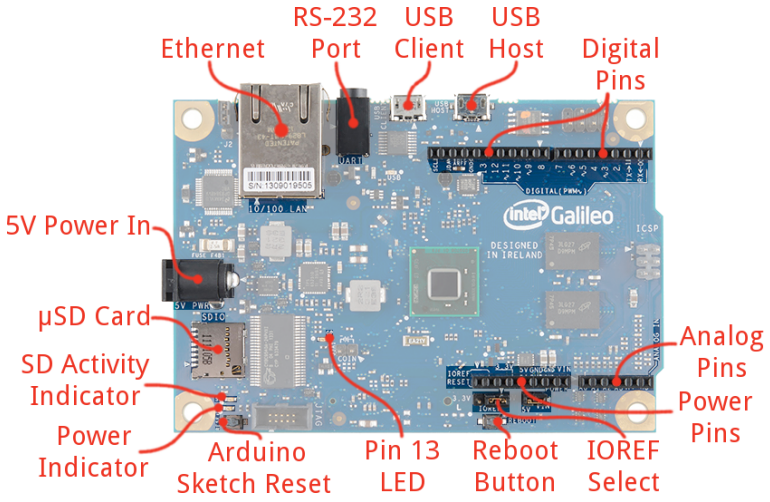

Figura 1. Diagrama da Intel Galileo Gen 1. fonte: https://learn.sparkfun.com/tutorials/galileo-getting-started-guide

```
}

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

}
```

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

Algumas referências sobre o estilo de programação Arduíno:

- 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 Arduíno. 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*, *pstree* 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 Arduíno - no entanto, ele é muito mais que um Arduíno.

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 desenvolvido um processo (ou thread) *Leitor* que lê continuamente os dados de temperatura e luminosidade e armazenar em uma área de dados. Devem ser armazenados o horário da leitura e o valor lido. O processo *Leitor* deve ser o único responsável pelo acesso aos sensores. O processo Leitor deve realizar a leitura acessando a *Linux Sysfs interface*. Ver http://www.malinov.com/Home/sergey-s-blog/intelgalileo-programminggpiofromlinux.

2.2 Fase 2

Deve ser implementado um processo *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. O servidor Web deve acessar os dados gravados pelo processo *Leitor*. A página deve indicar o horário da leitura e o valor de todas as leituras disponíveis.

2.3 Fase 3

Deve ser desenvolvido um servidor que permite gerenciar as leituras realizadas. O servidor deve responder às seguintes mensagens:

1. ERASE-ALL: apaga todas as leituras disponíveis.
2. GET-ALL: transmite todas as leituras.
3. GET <n>: transmite a *n-ésima* leitura <n>.
4. SET-PERIOD <p>: configura o intervalo entre leituras, com *p* dado em milisegundos.
5. GET-PERIOD: transmite o intervalo entre leituras.

O teste deste módulo será realizado com um cliente TELNET em um computador remoto.

2.4 Observações

As funções desenvolvidas nas Fases 1, 2 e 3 devem funcionar simultaneamente. O servidor Web deve atender um número arbitrário de conexões simultâneas.

Importante: Não esqueça de resolver os problemas de condições de corrida entre os 3 processos (ou threads) desenvolvidos.

2.5 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 3 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.

4 Datas

Entrega do relatório e apresentação: 27/11/2017.