

# Lição 1: Temporização, sensoriamento e comunicação a um salto

## Minicurso: Desenvolvimento de aplicações para RSSFs usando a linguagem nesC

Prof. Silvana Rossetto

<sup>1</sup>Projeto (CIA)<sup>2</sup> — Ambiente de experimentação remota de IoT  
6 de outubro de 2011

### Apresentação

- **Objetivo principal:** *Construir uma aplicação básica de sensoriamento com leitura periódica de um sensor, transmissão das leituras a um salto e exibição dos valores sensorizados via leds.*
- **Objetivos intermediários:**
  - introduzir conceitos básicos sobre a linguagem nesC e sobre o modelo de programação do TinyOS;
  - compilar e executar aplicações nesC nas plataformas micaz e telosb;
- **Aplicações de referência:**
  - Blink/BlinkAppC, tutorials/BlinkTaskAppC
  - Sense/SenseAppC
  - tutorials/BlinkToRadio/BlinkToRadioAppC

### Atividade 1

- *Descrição:* Compilar e executar a aplicação Blink/BlinkAppC.nc
- *Tempo previsto:* 15min atividades + 20min exposição de conceitos.
- *Objetivos:* Apresentar as ferramentas de compilação, carga e execução de aplicações nesC/TinyOS nas plataformas micaz e telosb e introduzir os conceitos básicos da linguagem nesC.

### Execução e estudo da aplicação

1. Acesse o diretório TOSROOT/apps/Blink.
2. Acople uma plataforma micaz (chave em OFF) no board MIB520, e conecte o board a uma porta USB do seu computador.
3. Compile e instale a aplicação BlinkAppC fazendo:  
`make micaz install mib510, /dev/ttyUSB0`<sup>1</sup>
4. Desacople a plataforma do board, mude a chave para ON e observe a execução da aplicação.
5. Conecte uma plataforma telosb (sem as pilhas) a uma porta USB do seu computador.
6. Instale a aplicação BlinkAppC nessa plataforma fazendo:  
`make telosb install bsl, /dev/ttyUSB0`<sup>2</sup>

*Obs.: A versão em C da aplicação e os códigos executáveis gerados para cada plataforma alvo são guardados no respectivo diretório (micaz ou telosb), dentro da pasta build*<sup>3</sup>.

<sup>1</sup>Para ter permissão de escrita no dispositivo, execute como super-usuário o comando `chmod 666 /dev/ttyUSB0`

<sup>2</sup>Caso necessário, execute o comando `motelist` para checar quais portas possuem motes telosb acoplados.

<sup>3</sup>O comando `make clean` remove todos os arquivos criados na compilação.

**Exposição/Comentários** Acompanhe a exposição do instrutor sobre a Atividade-1 (vide *slides-licao-1.pdf*)

## Atividade 2

- *Descrição*: Estudar o código, compilar e executar a aplicação `tutorials/BlinkTask/BlinkTaskAppC.nc`.
- *Tempo previsto*: 5min atividades + 10min exposição de conceitos.
- *Objetivo*: Mostrar o uso de tarefas postergadas em nesC.

## Estudo e execução da aplicação

1. Acesse o diretório `TOSROOT/apps/tutorials/BlinkTask`.
2. Estude o conteúdo dos arquivos desse diretório para compreender a diferença entre as aplicações `BlinkAppC` e `BlinkTaskAppC`.
3. Compile e execute a aplicação nas plataformas `micaz` e `telosb`.

**Exposição/Comentários** Acompanhe a exposição do instrutor sobre a Atividade-2 (vide *slides-licao-1.pdf*)

## Atividade 3

- *Descrição*: Estudar o código, compilar e executar a aplicação `Sense/SenseAppC.nc`.
- *Tempo previsto*: 15min atividades + 15min exposição.
- *Objetivo*: Mostrar como coletar medidas dos sensores.

## Estudo e execução da aplicação

1. Acesse o diretório `TOSROOT/apps/Sense`.
2. Estude o conteúdo dos arquivos desse diretório e tente compreender como a aplicação funciona.
3. Compile e execute a aplicação na plataforma `micaz`.

## Modificação da aplicação para usar um sensor real

1. Faça uma cópia do diretório da aplicação.
2. Altere o componente de sensoreamento de `DemoSensorC` para `PhotoC`.
3. Altere a implementação da aplicação para que o mote exiba nos seus leds a faixa do valor sensoreado (comece com as seguintes faixas:  $> 300$  acende led 0,  $> 500$  acende led 1,  $> 700$  acende led 2).
4. Compile e instale a aplicação na plataforma `micaz` fazendo:  

```
SENSORBOARD=mda1004 make micaz install mib510,/dev/ttyUSB0
```

*(Atenção para a necessidade de uso da variável de sistema `SENSORBOARD`, indicando qual placa de sensor implementará o componente de sensoreamento.)*
5. Acople o sensor `mda100` no mote e observe a execução da aplicação (tente fazer a luminosidade variar ao redor do mote).

*Obs.: Os componentes de implementação dos sensores acoplados à plataforma `telosb` não estão disponíveis na versão corrente do TinyOS.*

<sup>4</sup>Pode variar de acordo com o dispositivo de sensoreamento usado, no caso do sensor MTS300, fazer: `SENSORBOARD=mts300`

**Exposição/Comentários** Acompanhe a exposição do instrutor sobre a Atividade-3 (vide *slides-licao-1.pdf*)

#### Atividade 4

- *Descrição:* Estudar o código, compilar e executar a aplicação `tutorials/BlinkToRadioAppC.nc`
- *Tempo previsto:* 15min atividades + 30min exposição.
- *Objetivo:* Mostrar como enviar e receber mensagens entre nós da rede.

#### Estudo e execução da aplicação

1. Acesse o diretório `TOSROOT/apps/tutorials/BlinkToRadio`
2. Estude o conteúdo dos arquivos desse diretório e tente compreender como a aplicação funciona.
3. Altere o código das mensagens para a sua data de aniversário (arquivo `BlinkToRadio.h`)
4. Compile e instale a aplicação em dois motes `mica2` e observe a execução da aplicação <sup>5</sup>.
5. Altere o destinatário das mensagens. Ao invés de enviar a mensagem para qualquer nó da rede, usando a constante `AM_BROADCAST_ADDR`, escolha dois identificadores específicos para os seus nós e em cada um instale o código da aplicação que envia as mensagens para o outro nó. Por default, o `ID=1` é associado a cada nó durante a instalação da aplicação. Para alterar esse valor, altere a linha de comando para:  

```
make mica2 install.ID mib510,/dev/ttyUSB0
```
6. Verifique se os nós continuam trocando mensagens.
7. Recarregue a aplicação em um dos nós, alterando apenas o seu ID, verifique o que aconteceu.

**Exposição/Comentários** Acompanhe a exposição do instrutor sobre a Atividade-4 (vide *slides-licao-1.pdf*)

#### Atividade 5

- *Descrição:* Desenvolver uma aplicação básica de sensoramento em uma RSSF.
- *Tempo previsto:* 30min.
- *Objetivo:* Aplicar os conhecimentos adquiridos para construir novas aplicações TinyOS.

#### Construção e execução da aplicação

1. Crie uma aplicação que colete dados de um sensor de luminosidade a cada 0,5seg e envie esses dados para um nó específico da rede. Defina um código para as suas mensagens. (*ATENÇÃO: essa aplicação não deve receber mensagens da rede.*)
2. Crie outra aplicação que receba as mensagens da rede com o código usado no passo anterior e exiba a informação recebida nos leds. (*ATENÇÃO: essa aplicação não deve enviar mensagens para a rede.*)
3. Instale as aplicações em duas plataformas `mica2` e avalie a execução.

---

<sup>5</sup>No caso de plataformas `mica2` incluir a diretiva de compilação `PFLAGS=-DCC1K_DEF_FREQ=916400000`

## **Fechamento**

- *Descrição*: Discussão sobre as dificuldades encontradas e os resultados obtidos.
- *Tempo previsto*: 10min.

## **Referências bibliográficas e informações adicionais**

1. Lição 1 do TinyOS-2.x (Introdução)
2. Lição 2 do TinyOS-2.x (Módulos e modelo de execução do TinyOS)
3. Lição 3 do TinyOS-2.x (Comunicação no TinyOS)
4. Lição 5 do TinyOS-2.x (Sensoriamento no TinyOS)