

# Roteiro de instalação de um testbed para aplicações de redes de sensores sem fio

**GT-Tel: Grupo de Trabalho - Testbed para espaços Inteligentes  
Março de 2015**

## ÍNDICE

	<b>página</b>
1. Sistemas Operacionais.....	03
2. Instalação Testbed.....	04
2.1)1ª Solução: Instalação.....	04
2.1.1)Instalação Linux.....	04
2.1.1.1)Programas.....	04
2.1.1.2)Portal Testbed.....	06
2.1.1.3)Aplicativo TestbedControl.....	07
2.1.1.4)Configuração de Rede do Linux.....	07
2.1.2)Instalação Windows.....	08
2.1.2.1) Programas.....	08
2.1.2.2) Instalação TBControl Windows.....	09
2.1.2.3) Configuração de Rede do Windows.....	09
3. Configuração dos nós.....	10
3.1)MicaZ+sensor MTS300+MIB 600.....	10
3.2)Telosb+silex.....	11
3.3) Raspberry Pi + Telosb.....	12
4. Banco de dados.....	13
4.1)Informações básicas.....	13
4.2)Criação/Inserindo um nó.....	14
4.3) Configuração do nó - BD.....	15
4.3.1) Micaz + MIB 600.....	15
4.3.2) Telosb+silex.....	15
4.3.3) Telosb+Raspberry Pi.....	15
5. Configurar TBControl como serviço.....	16
6. Inicializar MV do VirtualBox no boot do host.....	17

### **Sistemas operacionais utilizados**

- Ubuntu 14.04.01 64 bits
- Windows 7 em máquina virtual (Configuração apenas para Silex+telosb)

## **Instalação do Testbed**

### **2.1)1ª Solução: Instalação**

#### **2.1.1)Instalação Linux**

##### **2.1.1.1)Programas**

1. Instalar Lua
  - a. `sudo apt-get install lua5.1`
  - b. `sudo apt-get install lua-md5`
2. Instalar Apache
  - a. `sudo apt-add-repository ppa:ondrej/apache2`
  - b. `sudo apt-get update`
  - c. `sudo apt-get install apache2`
3. Habilitar Lua no Apache
  - a. `sudo a2enmod lua`
4. Reiniciar o serviço do Apache
  - a. `sudo service apache2 restart`
5. Instalar luarocks
  - a. `sudo apt-get install luarocks`
6. Instalar o Git
  - a. `sudo apt-get install git`
7. Instalar Sailor e luasec
  - a. `sudo luarocks install luasec`  
`OPENSSL_LIBDIR=/usr/lib/x86_64-linux-gnu/`
  - b. `sudo luarocks install cgilua 5.1.4-2`
  - c. `sudo luarocks install sailor 0.2.1-1`
8. Instalar Postgresql
  - a. `sudo apt-get install postgresql libpq-dev`
9. Instalar suporte Postgres para Lua
  - a. `sudo luarocks install luasql-postgres`  
`PGSQL_INCDIR=/usr/include/postgresql/`
10. Criar senha do postgres
  - a. `sudo -u postgres psql`
  - b. `\password`
  - c. `<defina uma senha do postgres>`
  - d. `\q`

## 11. Instalar Uisp Tools

### a. Adicione os seguintes repositórios:

i. `sudo gedit /etc/apt/sources.list`

### ii. Adicione no final do arquivo:

`deb http://tinyos.stanford.edu/tinyos/dists/ubuntu lucid`

`main`

`deb http://tinyos.stanford.edu/tinyos/dists/ubuntu`

`maverick main`

`deb http://tinyos.stanford.edu/tinyos/dists/ubuntu natty`

`main`

### b. `sudo apt-get update`

### c. `sudo apt-get install tinyos-tools`

## 12. Complementos

### a. `sudo apt-get install binutils-avr`

### b. `sudo apt-get install binutils-msp430`

### c. `sudo apt-get install python-psycpg2`

## 13. Java 1.7 (Oracle/Sun)

### a. Adicione o seguinte repositório

i. `sudo add-apt-repository ppa:webupd8team/java`

### b. `sudo apt-get update`

### c. `sudo apt-get install oracle-jdk7-installer`

### d. Execute no terminal o comando: “java –version” e verifique se a versão do Java é 1.7.0\_xx

i. Caso não seja a versão 1.7 faça:

1. `sudo apt-get install oracle-java7-set-default`

### 2.1.1.2)Portal Testbed

1. Baixar aplicação Testbed
  - a. `cd ~`
  - b. `git clone https://github.com/projetotestbed/Testbed`
2. Habilitar todas as permissões
  - a. `sudo chmod 777 -R ~/Testbed`
3. Criar o banco de dados
  - a. `cd ~/Testbed/BD`
  - b. `psql -h localhost -U postgres < TBDB.bck`
  - c. <digitar senha do postgres definida no item acima>
4. Configurar conf.lua
  - a. Edite o arquivo `~/Testbed/TBportal/conf/conf.lua` e altere as informações conforme as suas configurações.
5. Copiar aplicação web para dentro de `/var/www/html`
  - a. `sudo cp ~/Testbed/TBportal /var/www/html/TBportal -r`
  - b. `sudo cp ~/Testbed/Site /var/www/html/Site -r`
  - c. Apagar tudo do arquivo `/var/www/html/index.html` e adicionar a seguinte linha: `<META http-equiv="refresh" content="1;URL=http://IP/Site">`
6. Configurar o `apache2.conf` - mudar `*allow overwrite*` para `*all*` dentro da sessão `*/var/www*` em `/etc/apache2/apache2.conf`
  - a. Abra o arquivo `/etc/apache2/apache2.conf` em seu editor de preferência
  - b. Em `<Directory /var/www/>` substitua `AllowOverride None` por `AllowOverride all`
7. Dar permissão de escrita para o Apache escrever arquivo de sessão
  - a. `sudo chown www-data:www-data /var/www/html/TBportal/runtime/tmp`
8. Reiniciar o serviço do Apache
  - a. `sudo service apache2 restart`

### **2.1.1.3)Aplicativo de Controle do Testbed**

1. Na pasta ~/Testbed/TBControl/files
  - a. Defina o caminho correto no arquivo startTBrelay.sh
  - b. Configure o arquivo TBDB.cfg conforme as configurações do banco de dados
2. Edite o arquivo "TestbedControl.properties" para corrigir os parâmetros do banco de dados, o caminho do programa e o tempo de checagem do status do nó
3. Copie "TestbedControl.properties" para o diretório root do usuário como ".TestbedControl.properties" (note o ponto incluído antes do nome)
4. crie um diretório e de permissão para a pasta.
  - a. mkdir ~/Testbed/TBControl/logs
  - b. sudo chmod 777 -R ~/Testbed/TBControl/logs

### **2.1.1.4)Configuração da Rede Linux**

1. Adaptador de Rede 1 (Conexão externa)
  - a. Configuração padrão: com IP fixo ou dinâmico
2. Adaptador de Rede 2 (Conexão interna - switch)
  - a. Endereço IP: 192.168.0.100
  - b. Netmask: 255.255.255.0

## 2.1.2) Instalação Windows

\*Windows 7 instalado em uma máquina virtual no VirtualBox (necessário apenas no caso de usar o hardware Silex para carga remota de código nas plataformas TelosB)

### 2.1.2.1) Programas

1. Instale os seguintes programas:

a. Cygwin

i. <http://cygwin.com/install.html>

ii. Obs.: instale os pacotes de python para cygwin.

b. Python 2.7.7

i. <https://www.python.org/download/>

c. Pyserial:

i. <https://pypi.python.org/pypi/pyserial>

ii. Obtenha o pacote .tar.gz e o copie para a pasta home do cygwin

iii. Extraia o arquivo

1. Em um terminal do cygwin, redirecione para a pasta “home” e digite:

a. `tar -vzxf nomedoarquivo.tar.gz`

iv. Mude o diretorio do cygwin para a pasta descompactada e execute:

1. `python setup.py install`

d. SX Virtual Link (software da placa Silex)

i. Obtenha a versão mais recente do software e instale:

ii. <http://silexamerica.com/campaign/sx-virtual-link-do>

e. Driver para TelosB

i. Conecte um mote TelosB na máquina virtual

ii. Obtenha o driver VCP mais recente para Windows:

1. <http://www.ftdichip.com/Drivers/VCP.htm>

iii. Siga o tutorial abaixo:

1. [http://www.ftdichip.com/Support/Documents/AppNotes/AN\\_119\\_FTDI\\_Drivers\\_Installation\\_Guide\\_for\\_Windows7.pdf](http://www.ftdichip.com/Support/Documents/AppNotes/AN_119_FTDI_Drivers_Installation_Guide_for_Windows7.pdf)

iv. Obs.: Caso a máquina virtual exiba a seguinte mensagem: “No devices connected”, siga os seguintes procedimentos:

1. Obtenha o software “Extension Pack” conforme a versão do VirtualBox



- a. <https://www.virtualbox.org/wiki/Downloads>
2. Abra seu VirtualBox e clique em “File” -> “Preference”
3. Na janela aberta selecione “Extensões”, escolha o ícone azul e em seguida o pacote obtido
4. Execute em um terminal do Linux o seguinte comando:
  - a. `usermod NOME -a -g vboxusers`
  - b. Substitua o campo “NOME” para o nome do seu usuário

#### **2.1.2.2) Instalação TBControl Windows**

1. Crie um diretório na pasta raiz do usuário do cygwin e copie os arquivos da pasta “TBControl\_cygwin”
  - a. <https://github.com/projetotestbed/Testbed>
2. Edite o arquivo “bsl\_data\_server.py” e corrija o caminho “/home/tbdev/TBControl/files/tos-bsl.exe” (linha 228) para o caminho correto conforme o diretório criado em no item 1
3. Abra com um editor de texto o arquivo “start\_SXxxxxxx” e corrija o caminho “/home/tbdev...” para o caminho correto (linha 5)

#### **2.1.2.3) Configuração de Rede do Windows**

1. Na parte de ‘rede’ das configurações da máquina virtual faça as seguintes modificações:
  - a. Em adaptador 1: deixe em modo Bridge e aponte para a eth0 (rede conectada ao switch)
  - b. Em adaptador 2: deixe em modo NAT
2. Configurando IP fixo para o Windows.
  - a. Com o Windows iniciado, abra as configurações de rede no protocolo IPv4 e faça as seguintes modificações na rede conectado ao switch:
    - i. Endereço IP: 192.168.0.101
    - ii. Máscara da sub-rede: 255.255.255.0
    - iii. Gateway padrão: 192.168.0.100
    - iv. DNS preferencial: 192.168.0.100

## Configuração dos nós

### 3.1)MicaZ+sensor MTS300+MIB 600

1. Configurando o IP e porta da mib 600
  - a. Conecte a mib600 na mesma rede de sua máquina
  - b. Faça: `arp -s ip_desejado mac_address_da_mib600`
  - c. `telnet ip_desejado 1`
  - d. `telnet ip_desejado 9999` e configure o IP conforme a sua rede interna
    - i. Na nossa solução utilizamos como IP: 192.168.0.x e Máscara da sub-rede: 255.255.255.0
2. Configurando o canal de controle (canal 1) e o canal de dados (canal 2) da placa mib 600:
  - a. Acesse a mib600 fazendo: `telnet IP 9999` (onde 9999 é a porta de configuração da mib):
    - i. Canal 1:  
Baudrate 115200, I/F Mode 4C, Flow 00  
Port 10001  
Connect Mode : C0  
Auto increment source port disabled  
Remote IP Adr: --- none ---, Port 00000  
Disconn Mode : 00  
Flush Mode : 22
    - ii. Canal 2:  
Baudrate 57600, I/F Mode 4C, Flow 00  
Port 10002  
Connect Mode : C0  
Auto increment source port disabled  
Remote IP Adr: --- none ---, Port 00000  
Disconn Mode : 00  
Flush Mode : 22
    - iii. Caso as configurações estejam diferentes, altere conforme os dados acima

### 3.2)Telosb+Silex

1. Configurações no Windows:
  - a. Configurando IP da Silex:
    - i. Obs.: para facilitar a configuração, desconecte todos os outros TelosB já conectados ao switch
    - ii. Insira o CD de instalação do Silex
      1. Selecione a primeira opção do CD, e na próxima janela selecione a primeira opção
      2. Coloque para detectar o MAC e selecione *next*
      3. Configure o IP do Silex conectado como 192.168.0.x, a máscara sub-rede como 255.255.255.0 e o gateway padrão: 0.0.0.0
    - iii. Abra o programa SX Virtual Link
    - iv. Com o botão direito do mouse, clique em *connect*
  - b. No terminal do cygwin:
    - i. Mude para o diretório TBControl/files
    - ii. Execute: ./motelist-windows.exe
      1. Anote o COMx
    - iii. Execute: ls /dev/ttyS\*
      1. Anote o /dev/ttySx
  - c. Na pasta TBControl/files, edite o arquivo tb\_motes da seguinte forma:
    - i. <ID\_da\_placa\_silex> = 1000i,COMx,/dev/ttySx,115200, onde:
      1. <ID\_da\_placa\_silex>: pode ser encontrada pelo SX Virtual Link (seu formato é SXxxxxxx)
      2. O valor de i em 1000i tem que ser ímpar
  - d. Para cada mote TelosB adicionado é preciso fazer uma cópia do arquivo start\_SXxxxxxx.bat e renomeá-lo substituindo o "SXxxxxxx" pelo ID da Silex
  - e. Abra novamente o SX Virtual Link
    - i. Com o botão direito selecione "properties", na segunda aba ("connect") ative as duas primeiras opções de "Auto Connect" e marque a opção "start designed ...", selecione "Browse" e escolha o arquivo star\_SXxxxxxx que foi renomeado anteriormente

### 3.3) Raspberry Pi + Telosb

#### 1. Instalando o sistema operacional:

- a) Baixe a ultima versão do Raspbian (Versão utilizada: 2014-09-09):
  - i) <http://www.raspberrypi.org/downloads/>
- b) Baixe o seguinte arquivo para instalar o Debian no Raspberry Pi:
  - i) <http://www.softpedia.com/get/CD-DVD-Tools/Data-CD-DVD-Burning/Win32-Disk-Imager.shtml>
- c) Descompacte a imagem do Raspbian e no program Win32-Disk-Imager selecione a imagem do Raspbian e o local do SD Card. Escolha a opção 'Write'.

#### 2) Configurando o Raspbian pela primeira vez:

- a) Conecte os seguintes componentes ao RPi: SDCard, cabo ethernet, teclado e por ultimo o conecte a fonte de energia.
- b) No primeiro boot, irá abrir uma janela de configuração do Raspberry Pi (raspi-config) realize as seguintes configurações:
  - i) Escolha a primeira opção: "1 - expand Filesystem"
  - ii) Na opção "8 advanced option" habilite o acesso ssh
- c) Login e senha padrão:
  - i) Login: pi
  - ii) senha: raspberry

#### 3) Configurando o TBControl no Raspberry Pi:

- a) Baixe e copie a pasta TBControl\_RPi para a pasta 'home/pi'
  - i) <https://github.com/projetotestbed/Testbed>
    - 1) Obs.: Deverá ficar algo como: /home/pi/TBControl/files
    - 2) Caso o diretório fique diferente modifique o seguinte caminho:
      - a) no arquivo bsl\_data\_server.py (TBControl/files) altere a linha 228(linha após #call tos-bsl) para o caminho correto do arquivo tos-bsl.py
    - 3) Edite o arquivo tb\_motes da seguinte forma:
      - a) mote1 = 10001,COM3,/dev/USB0,115200
    - 4) Inicie o TBControl do RPi:
      - a) ./mote1.sh

#### 4) Configuração de rede do RPi:

- a) Edite o arquivo da interface de rede da seguinte forma:
  - i) `sudo nano /etc/network/interfaces`
    - 1) altere a linha 'iface eth0 inet dhcp' para
      - a) `iface eth0 inet static`  
`address 192.168.0.x`  
`netmask 255.255.255.0`  
`gateway 0.0.0.0`

## Banco de dados

### 4.1) Informações Básicas

1. Abaixo segue a descrição de algumas tabelas importantes. Utilizadas na criação ou inserção de novas informações no Banco de Dados :
2. Tabela “*nodestatus*”
  - a. Tabela com os possíveis status dos nós
  - b. Atributos:
    - i. “statusid”: ID do status
    - ii. “code”: Nome do status
    - iii. “description”: Descrição do status
3. Tabela “*nodetype*”
  - a. Tabela com os possíveis tipos de nós presente no testbed
  - b. Atributos:
    - i. “nodetype”: ID do tipo do nó
    - ii. “short”: Descrição mais simples do nó
    - iii. “description”: Descrição completa sobre o nó
4. Tabela “*networks*”
  - a. Tabela com as informações das Redes que estão configuradas o testbed
  - b. Atributos:
    - i. “netid”: ID da rede
    - ii. “netversion”: versão da rede
    - iii. “nname”: Nome da rede
    - iv. “description”: Descrição da Rede
    - v. “vdescription”: Descrição da versão da rede
5. Tabela “*network*”
  - a. Tabela que contém todos os nós configurados em todas as redes
  - b. Atributos:
    - i. “netid”: ID da rede
    - ii. “netversion”: versão da rede
    - iii. “nodeid”: ID do nó
    - iv. “nodetype”: Tipo do nó
    - v. “statusid”: status do nó
    - vi. “location”: Localização do nó
    - vii. “posx”: localização no eixo x
    - viii. “posy”: localização no eixo y
    - ix. “posz”: localização no eixo z
    - x. “ctlchannel”: informações para o canal de controle
    - xi. “datachannel”: informações para o canal de dados

## 4.2) Criação/inserindo um nó

1. Primeira configuração:
  - a. No processo de instalação do Testbed o Banco de Dados se encontra totalmente vazio
  - b. Para carregar no banco de dados as informações básicas faça:
    - i. Editar o arquivo “ResetAllData.lua” no diretório “~/Testbed/DB” e altere os dados conforme a configuração do seu Testbed.
    - ii. Carregar os dados básicos com o comando abaixo: (obs: este comando apaga todos os dados existentes e cria novos dados.)
      1. lua ResetAllData.lua
2. Inserindo um novo nó em uma rede
  - a. A inserção de um novo nó pode ser realizada utilizando um software de gerenciamento de Banco de dados (pgAdmin), por linha de comando ou editando o arquivo “AddNode.lua” na pasta “~/Testbed/DB” conforme as configurações do seu Testbed
    - i. A inserção de um novo nó em uma Rede é realizada na tabela “network”
    - ii. Utilizando pgAdmin
      1. No pgadmin abra a tabela “network” (localizado em: portal->Schemas->public->Tables)
      2. Adicione uma nova linha para um nó conforme as suas configurações
    - iii. Linha de comando
      1. ~\$ psql -h localhost -d “dbName” -U “User”
        - a. onde
          - i. dbName = Nome do Banco de dados(padão: portal2)
          - ii. User: usuário do postgresql(padão: postgres)
        2. Por comandos SQL(insert,update,delete) modifique a tabela conforme as suas configurações

### **4.3) Configuração do nó - BD**

1. Configurações básicas no Banco de Dados para adicionar um determinado tipo de nó na rede.

#### **4.3.1) Micaz+MIB 600**

1. nodetype = 1 (padrão)
2. ctlchannel = "-dprog=stk500 -dhost= "IP" "
3. datachannel = "1000i "IP" 10002"
4. Onde:
  - a. i: ID do micaz
  - b. IP: IP da MIB 600

#### **4.3.2) Telosb+silex**

1. nodetype = 2 (padrão)
2. Em ctlchannel configure como:
  - a. "192.168.0.101 <porta\_do\_telosb>"
3. Em datachannel configure como:
  - a. "1000i 192.168.101 <porta>"
4. Onde:
  - a. i: ID da Telosb
  - b. <porta\_do\_telosb> é a porta configurada no arquivo tb\_mote(no Windos: ~/TBControl/files)
  - c. <porta> é a <porta\_do\_telosb>+1

#### **4.3.3) Adicionando Telosb+Raspberry Pi**

1. nodetype = 2 (padrão)
2. ctlchannel = "IP 10001> "
3. datachannel = "1000i "IP" 10002"
4. Onde:
  - a. i: ID do telosb
  - b. IP: IP do RPi

### Configurar TBControl como serviço

1. Vá para o diretório TBControl/files
  - a. `cd ~/Testbed/TBControl/files`
2. Edite o arquivo tbcontrol no diretório files e coloque os valores corretos para:  
TBCUSER=tbdev  
TBCPATH=/home/tbdev/Testbed/TBControl/  
DATABASE=portal2  
DBUSER=postgres  
PGPWD=postgres

Sendo:

- \* TBCUSER - Usuário linux
- \* TBCPATH - Caminho do diretório onde encontra-se o TBControl.jar
- \* DATABASE - Nome do banco de dados no postgres
- \* DBUSER - usuário do banco de dados
- \* PGPWD - senha do banco de dados (criado no item acima: "criar senha do postgres")

3. Copie o arquivo para a pasta de serviços do Linux
  - a. `sudo cp tbcontrol /etc/init.d/.`
4. Configurar o serviço para iniciar no boot da máquina
  - a. `sudo update-rc.d tbcontrol defaults 95`
5. Configurar o tbcontrol para reiniciar toda semana
  - a. No terminal execute:
    - i. `crontab -e`
  - b. No arquivo que irá abrir adicione no final do arquivo a seguinte linha:
    - i. `10 02 * * sun /etc/init.d/tbcontrol restart`
  - c. Execute:
    - i. `sudo /etc/init.d/cron restart`
6. Iniciar/Parar TBControl:
  - a. Iniciar: `sudo service tbcontrol start`
  - b. Parar: `sudo service tbcontrol stop`



### **Inicializar MV do VirtualBox no boot do host**

1. Configurar o start/stop das VMs como serviço no linux
  - a. Vá para o diretório TBControl/files
    - i. `cd ~/Testbed/TBControl/files`
  - b. Edite o arquivo vmboot no diretório files e coloque o valor correto para:  
`VBOXUSER=tbdev`
- Sendo:
  - \* VBOXUSER - Usuário linux
  - c. Copie o arquivo para a pasta de serviços do linux
    - i. `sudo cp vmboot /etc/init.d/`
  - d. Configurar o serviço para iniciar no boot da máquina
    - i. `sudo update-rc.d vmboot defaults 95`

\* obs) No shutdown o script salva os estados das VMs que estiverem ligadas.

No boot o script só inicializa as VMs que tiverem seus estados salvos. Se alguma VM não inicializar no boot, tente salvar o estado dela manualmente antes do shutdown.