Wrocław University of Technology

Group 4
Relational routing in WSN

MoteWorks Tutorial

TABLE OF CONTENTS

1. - INTRODUCTION

In this document, we are going to describe the necessary software and its configuration for starting to work in the project “Routing in Wireless Sensor Network (WSN)”. After following this manual, you will be able to reproduce and check our researches and achievements.

2. - USING A VIRTUAL MACHINE

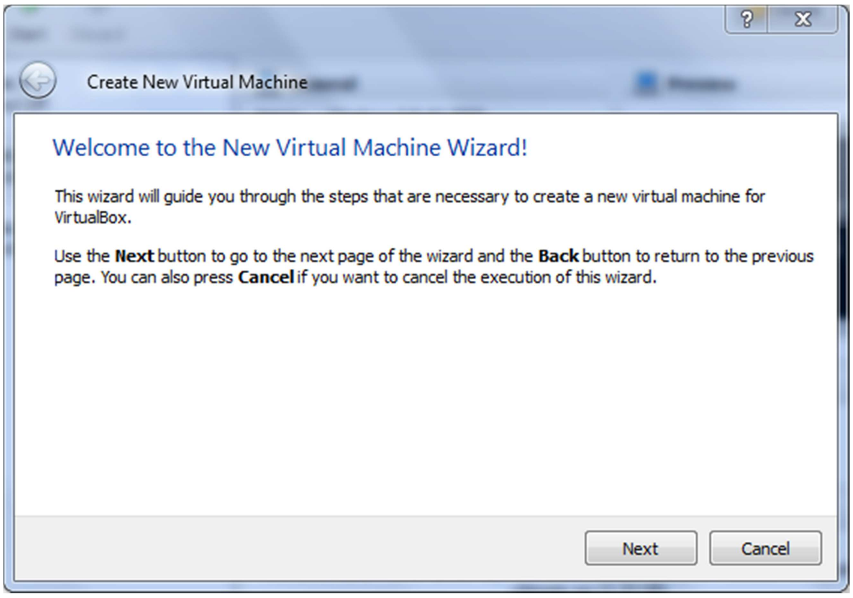
The platform for which is destined this manual is Windows, particularly, Windows XP 32 bits – English (USA). The main reason is that the software provided by Crossbow is not compatible with newer versions of Windows, so the installation process fails. Considering that, we propose using virtualization environment for running Windows XP.

What you will need:

- Virtual Machine. We recommend Oracle VM Virtualbox (link)
- Windows XP ISO file or CD-ROM
- Enough space on your HDD (~10 GB)
- Enough RAM (~512 MB)

Instructions:

1) Open up VirtualBox and click the 'New' button with a blue sun icon.
2) Click next in this following dialog box that will appear:



Welcome to the New Virtual Machine Wizard!

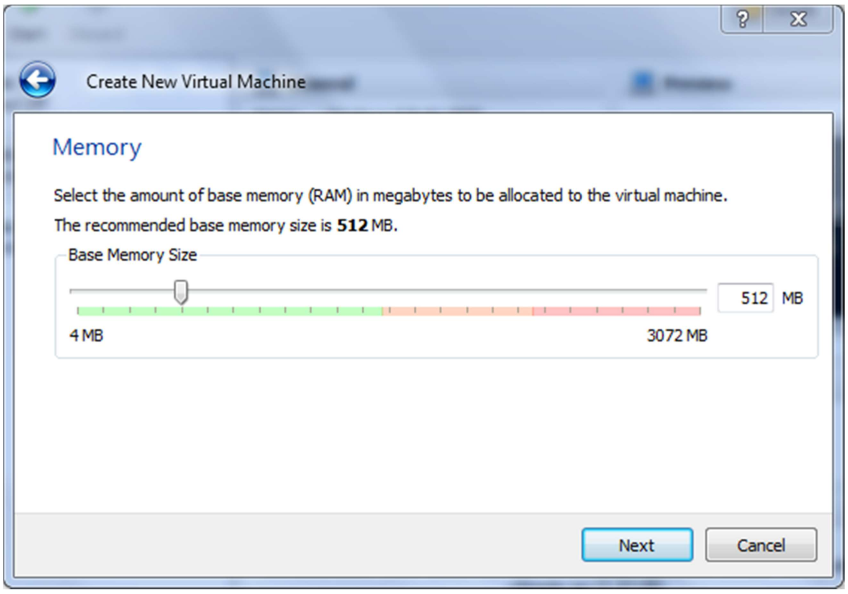
This wizard will guide you through the steps that are necessary to create a new virtual machine for VirtualBox.

Use the **Next** button to go to the next page of the wizard and the **Back** button to return to the previous page. You can also press **Cancel** if you want to cancel the execution of this wizard.

3) Type the name of the Virtual Machine and select Windows XP as the OS type, and click Next.

*If you input Windows XP as the name of your VM, the OS Type will auto-adjust to Windows XP.

4) Next, select the amount of RAM in this window (~512MB) and click Next:



Memory

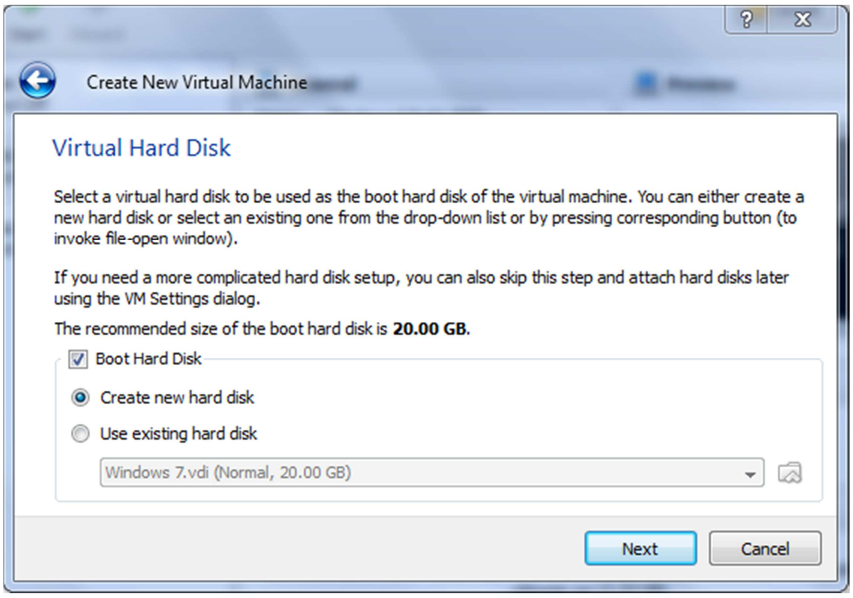
Select the amount of base memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended base memory size is **512** MB.

Base Memory Size: 512 MB

*If you don't have enough RAM to spare, put a bit lesser than 512 MB, but features might be disabled.

5) Click Next again in this window to create a new virtual disk:



Virtual Hard Disk

Select a virtual hard disk to be used as the boot hard disk of the virtual machine. You can either create a new hard disk or select an existing one from the drop-down list or by pressing corresponding button (to invoke file-open window).

If you need a more complicated hard disk setup, you can also skip this step and attach hard disks later using the VM Settings dialog.

The recommended size of the boot hard disk is **20.00 GB**.

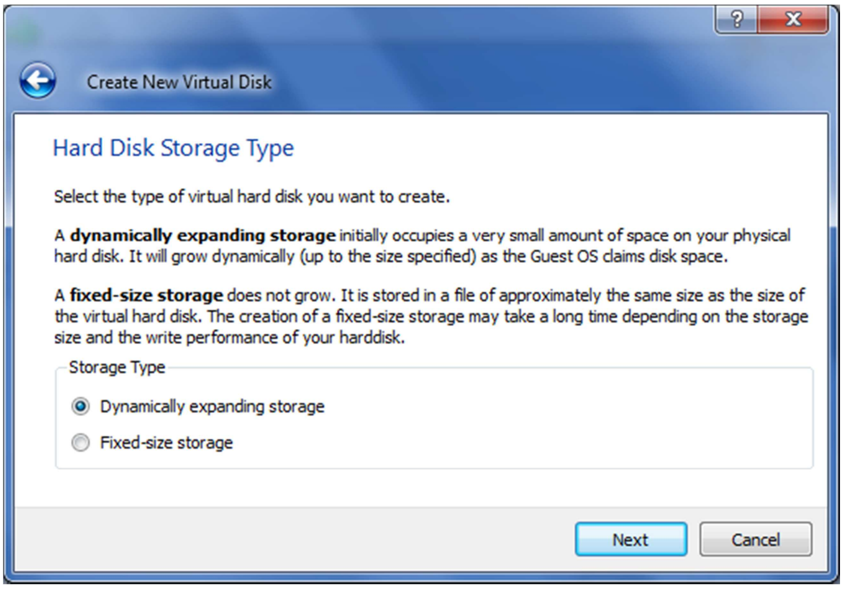
☑ Boot Hard Disk

● Create new hard disk

○ Use existing hard disk

Windows 7.vdi (Normal, 20.00 GB)

6) A new window will pop-up. Click Next.
7) Select 'Dynamically expanding storage in this window:



Create New Virtual Disk

Hard Disk Storage Type

Select the type of virtual hard disk you want to create.

A **dynamically expanding storage** initially occupies a very small amount of space on your physical hard disk. It will grow dynamically (up to the size specified) as the Guest OS claims disk space.

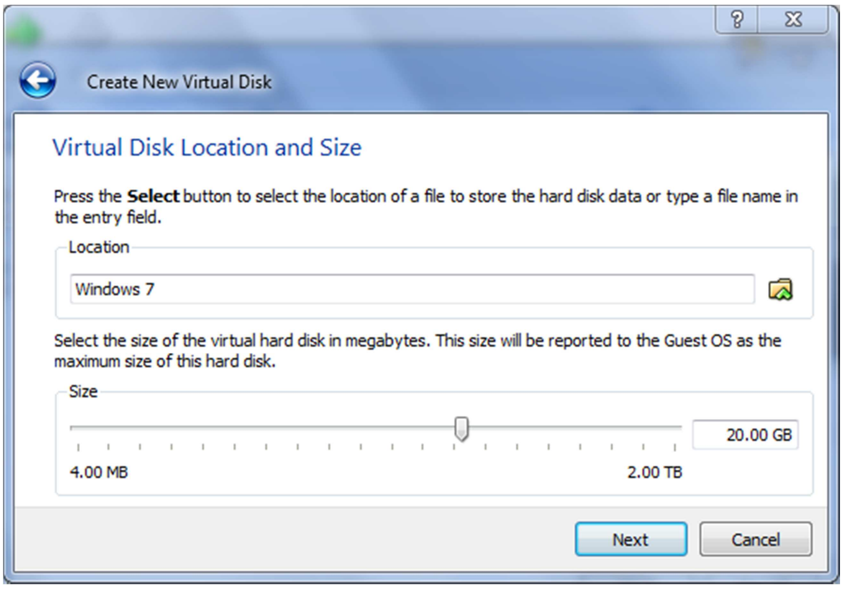
A **fixed-size storage** does not grow. It is stored in a file of approximately the same size as the size of the virtual hard disk. The creation of a fixed-size storage may take a long time depending on the storage size and the write performance of your harddisk.

Storage Type

● Dynamically expanding storage

○ Fixed-size storage

8) Click Next in the window.
9) Select 10 GB and click Next in this window:



Create New Virtual Disk

Virtual Disk Location and Size

Press the **Select** button to select the location of a file to store the hard disk data or type a file name in the entry field.

Location: Windows 7

Select the size of the virtual hard disk in megabytes. This size will be reported to the Guest OS as the maximum size of this hard disk.

Size: 20.00 GB

10) Click Finish in the remaining two windows.
11) Now is the time to go to the Settings page of your new VM and add or disable settings, according to your choice.
12) Double-click on your new Virtual Machine, and a few dialog boxes will popup, ignore them, but not the one that says: 'First Run Wizard' .
13) Navigate to ISO file/CD-ROM by clicking the folder icon with a green arrow and click Next once you're done.
14) Windows XP will now install. Installation will take not much time (probably lesser than 10 minutes).
15) The setup will ask you some info sometimes during the installation process, so don't leave your computer.
16) Done.

3. - INSTALLATION OF THE ENVIRONMENT - MOTEWORKS

MoteWorks is provided with a set of software development tools for custom Mote applications, including custom sensor board drivers, sensor signal conditioning and processing and message handlers. MoteWorks includes an optimized cross-compiler for the target mote platform and an advanced editor for TinyOS application development. MoteWorks automatically installs and configures these development tools for quick set-up and rapid start of development.

3.1. - WHAT YOU NEED FOR INSTALLATION

Requirements:

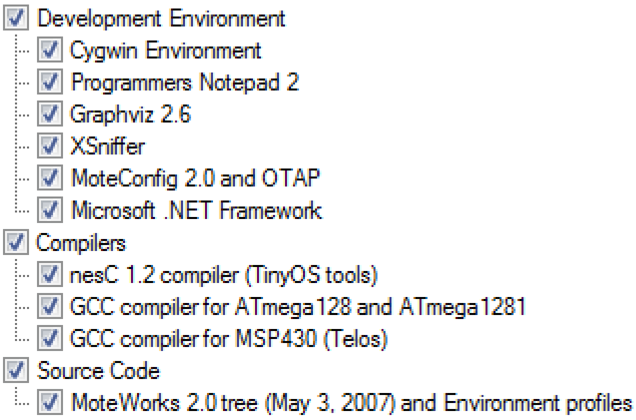
- Crossbow's MoteWorks CD-ROM (link)
- A Windows-based PC
  - Operating System: Microsoft Windows (XP, 2000, NT)
  - 1 GB or more of free space in destination drive
  - 550 MB or more of space in C drive, regardless of destination drive

The MoteWorks InstallShield Wizard setup offers the following software packages:

- **TinyOS**: An event-driven OS for wireless sensor networks
- **MoteWorks Tools:** Tools for debugging
- **nesC compiler** - An extension of C-language designed for TinyOS
- **Cygwin** - A Linux like environment for Windows
- **AVR Tools** - A suite of software development tools for Atmel's AVR processors
- **Programmer's notepad** - IDE for code compilation and debugging
- **XSniffer** - Network Monitoring Tool for the RF environment
- **MoteConfig** - GUI environment for Mote Programming and OTAP
- **Graphviz** - To view files made from make docs
- **PuTTY and TortoiseCVS** - Source access through CVS server for Enterprise Users

3.2. - INSTALLATION PROCESS

1) Navigate to folder "MoteWorks" and execute "MoteWorks_2.0.F_Setup.exe" (*2.0.F* is the current version).
2) At the License Agreement page, you should read and check on "I accept the agreement. Click on Next.
3) Specify the destination directory for the MoteWorks (default is *C:\Crossbow*) and click on Next. MoteWorks **should not** be installed to *C:\Program Files\Crossbow*
4) In the Select MoteWorks Components dialog, select Full installation from the drop down (recommended) Click on Next.

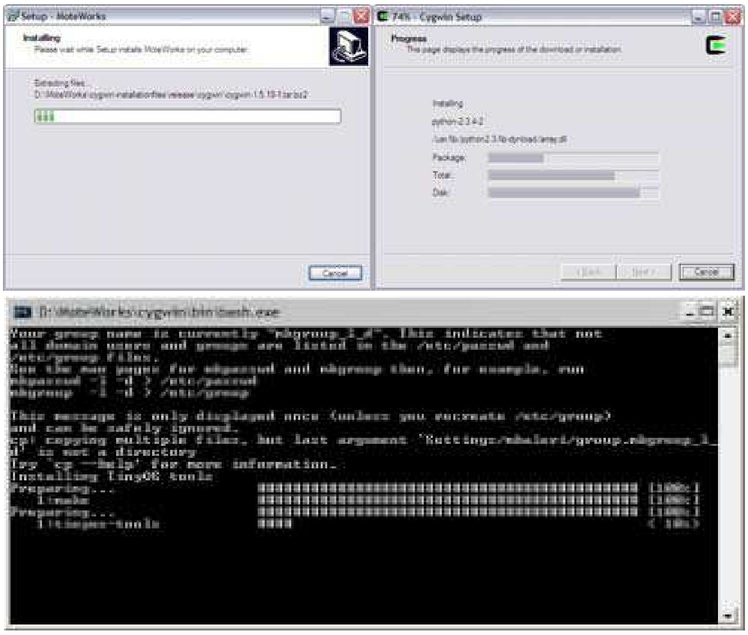


☑ Development Environment
- ☑ Cygwin Environment
- ☑ Programmers Notepad 2
- ☑ Graphviz 2.6
- ☑ XSniffer
- ☑ MoteConfig 2.0 and OTAP
- ☑ Microsoft .NET Framework

☑ Compilers
- ☑ nesC 1.2 compiler (TinyOS tools)
- ☑ GCC compiler for ATmega128 and ATmega1281
- ☑ GCC compiler for MSP430 (Telos)

☑ Source Code
- ☑ MoteWorks 2.0 tree (May 3, 2007) and Environment profiles

5) The windows shown will appear as the installation progresses. Wait patiently for further instructions.


Setup - MoteWorks
Installing
Please wait while Setup installs MoteWorks on your computer.
Extracting files...
Cancel
74% - Cygwin Setup
Progress
This page displays the progress of the download or installation.
Installing
python-2.3.4-2
Package:
Total:
Disk:
< Back
Next >
Cancel
D:\MoteWorks\cygwin\bin\bash.exe

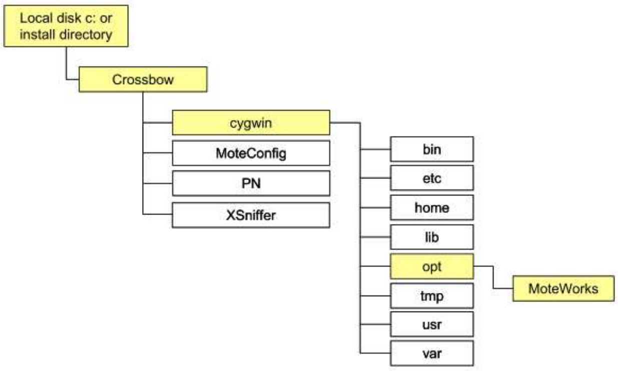
```
Your group name is currently "mkgroup_l_d". This indicates that not
all domain users and groups are listed in the /etc/passwd and
/etc/group files.
See the man pages for mkpasswd and mkgroup then, for example, run
mkpasswd -l -d > /etc/passwd
mkgroup  -l -d > /etc/group

This message is only displayed once (unless you recreate /etc/group)
and can be safely ignored.
cp: copying multiple files, but last argument `Settings/mbalari/group.mkgroup_l_
d' is not a directory
Try `cp --help' for more information.
Installing TinyOS tools
Preparing...                ########################################### [100%]
   1:make                   ########################################### [100%]
Preparing...                ########################################### [100%]
   1:tinyos-tools           ####                                        ( 10%)
```

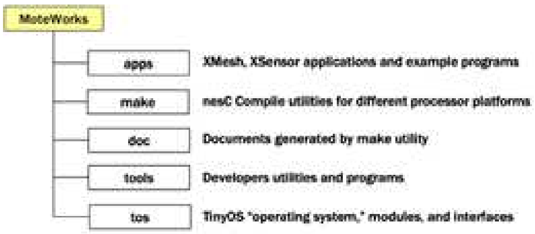
6) The next step is the installation of Programmer’s Notepad. At the License Agreement page, you should read and check on “I accept the agreement” before you can proceed further. Click on Next. (Notepad gets installed under the default folder *C:\ Crossbow\pn*).

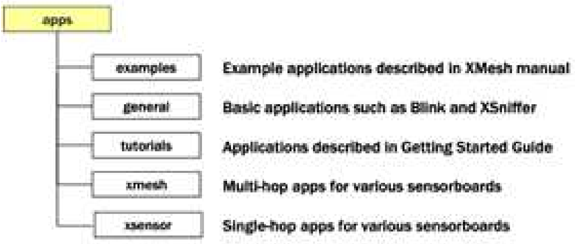
7) The next step is the installation of MoteConfig. Click Next on the welcome window. The installer will guide you through the rest of the process. (MoteConfig gets installed under the default folder *C:\ Crossbow\MoteConfig*).

3.3. - MOTEWORKS INSTALLATION STRUCTURE

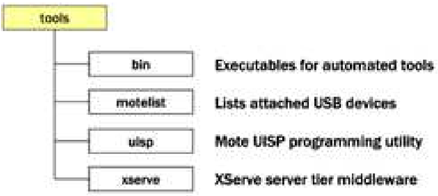
All the MoteWorks components such as *apps/*, *doc/, tools/*, and *tos/* directories are located under `<install dir>`*/cygwin/opt/MoteWorks/*. In addition the Makefile is in this folder. The environment variables for TOSROOT is set to `<install dir>`. Typically the default `<install dir>` is the *C:\ Crossbow*.

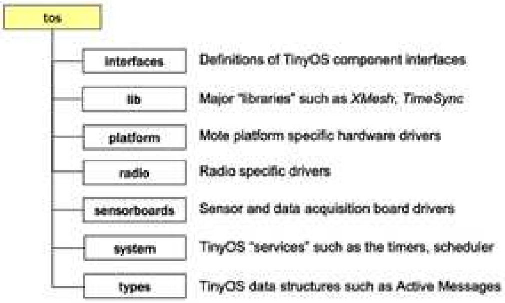



MoteWorks
apps
XMesh, XSensor applications and example programs
make
nesC Compile utilities for different processor platforms
doc
Documents generated by make utility
tools
Developers utilities and programs
tos
TinyOS "operating system," modules, and interfaces


apps
examples
Example applications described in XMesh manual
general
Basic applications such as Blink and XSniffer
tutorials
Applications described in Getting Started Guide
xmesh
Multi-hop apps for various sensorboards
xsensor
Single-hop apps for various sensorboards

(c) apps and subdirectories


tools
bin
Executables for automated tools
motelist
Lists attached USB devices
uisp
Mote UISP programming utility
xserve
XServe server tier middleware


tos
interfaces
Definitions of TinyOS component interfaces
lib
Major "libraries" such as XMesh, TimeSync
platform
Mote platform specific hardware drivers
radio
Radio specific drivers
sensorboards
Sensor and data acquisition board drivers
system
TinyOS "services" such as the timers, scheduler
types
TinyOS data structures such as Active Messages

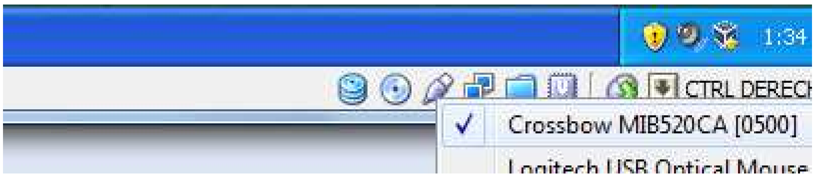
3.4. – CYGWIN

Cygwin is a Unix/Linux emulation environment for Microsoft Windows. It consists of two parts: A DLL (cygwin1.dll) which acts as a Linux API emulation layer providing substantial linux API functionality; a collection of tools, which provide a Linux look and feel. The Cygwin tools are ports of the popular GNU development tools for Microsoft Windows. Cygwin is an optional user interface for compiling and downloading Mote applications in MoteWorks. The Cygwin shell can be started by double clicking on the icon located on your desktop. You should see a new command prompt window.

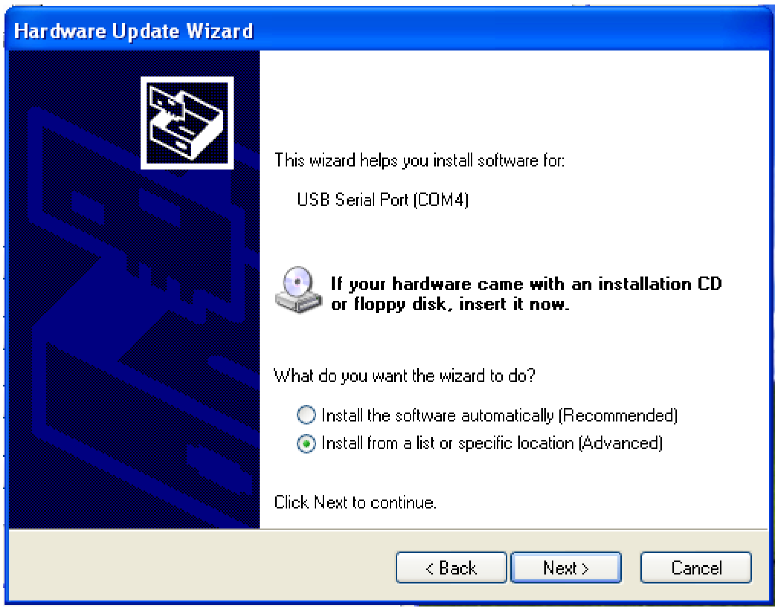
3.5. - CONFIGURING USB

Our platform will be MIB520. It uses the FTDI FT2232C to use the USB port as a virtual COM port. Hence you need to install FT2232C VCP drivers:

- Plug the USB and click Crossbow MIB520CA [0500]



- When you plug a MIB520 into your PC for the first time, Windows detects and reports it as a new hardware. Please select “Install from a list or specific location (Advanced)” and browse to “MIB520 Drivers” folder of the MoteWorks CDROM. The install shield wizard will guide you through the installation process.


Hardware Update Wizard

This wizard helps you install software for:

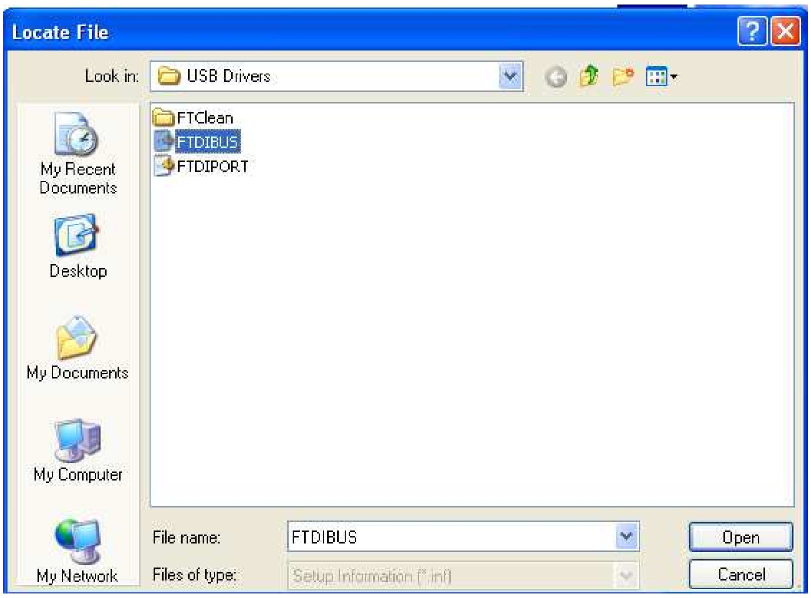
USB Serial Port (COM4)

If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

○ Install the software automatically (Recommended)
● Install from a list or specific location (Advanced)

Click Next to continue.

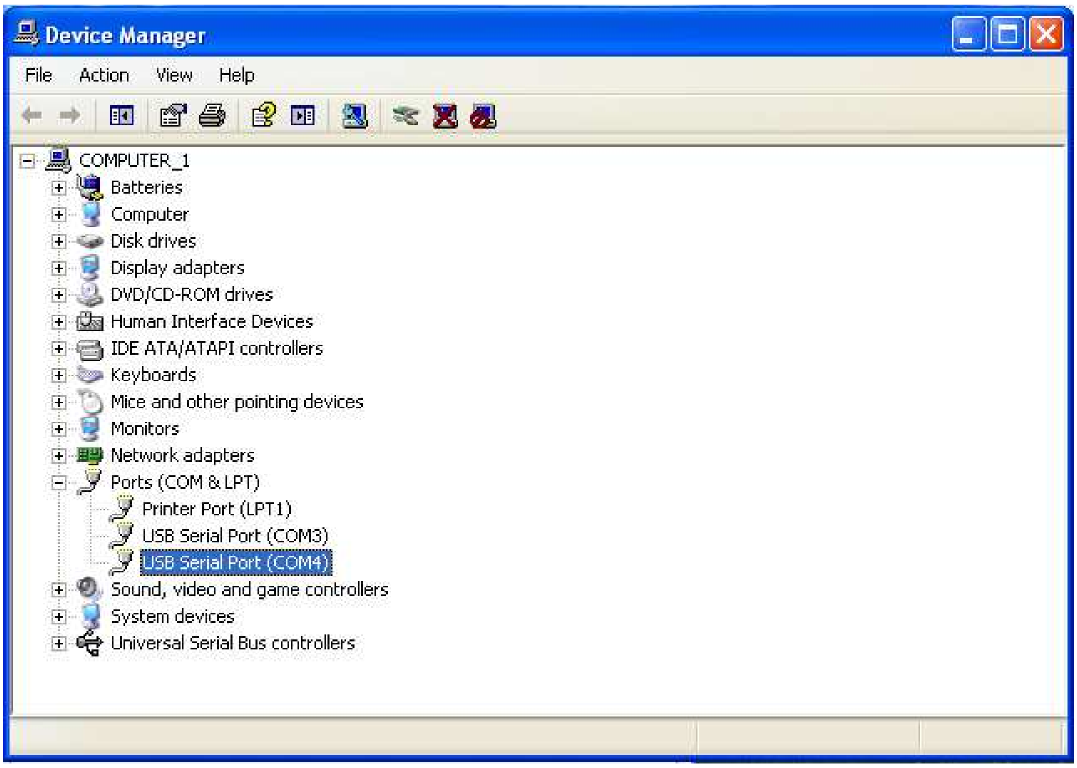


Look in: USB Drivers

File name: FTDIBUS

Files of type: Setup Information (*.inf)

- When the drivers are installed, you will see two serial ports added under the Control Panel>System>Hardware. Make a note of the assigned COM port numbers. In this case, COM3 and COM4.



COMPUTER_1
- Batteries
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
  - Printer Port (LPT1)
  - USB Serial Port (COM3)
  - USB Serial Port (COM4)
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers

- The two virtual serial ports for MIB520 are COMx and COM(x+1); COMx is for Mote programming and COM(x+1) is for Mote communication.

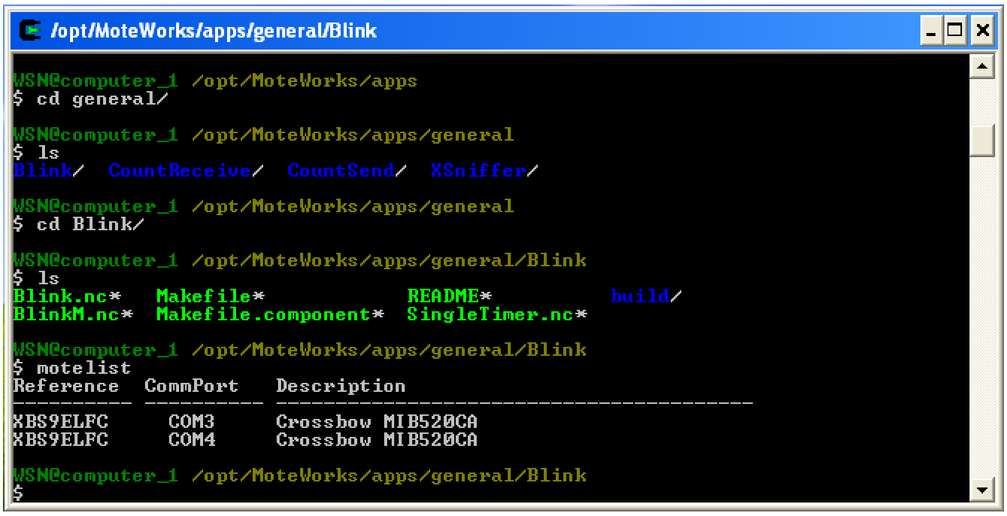
4. – PROGRAMMING BOARDS

4.1. - COMPILING MOTEWORKS APPLICATIONS

The syntax for compiling (building) application code in a Cygwin window is of the form:

```
make <platform>
```

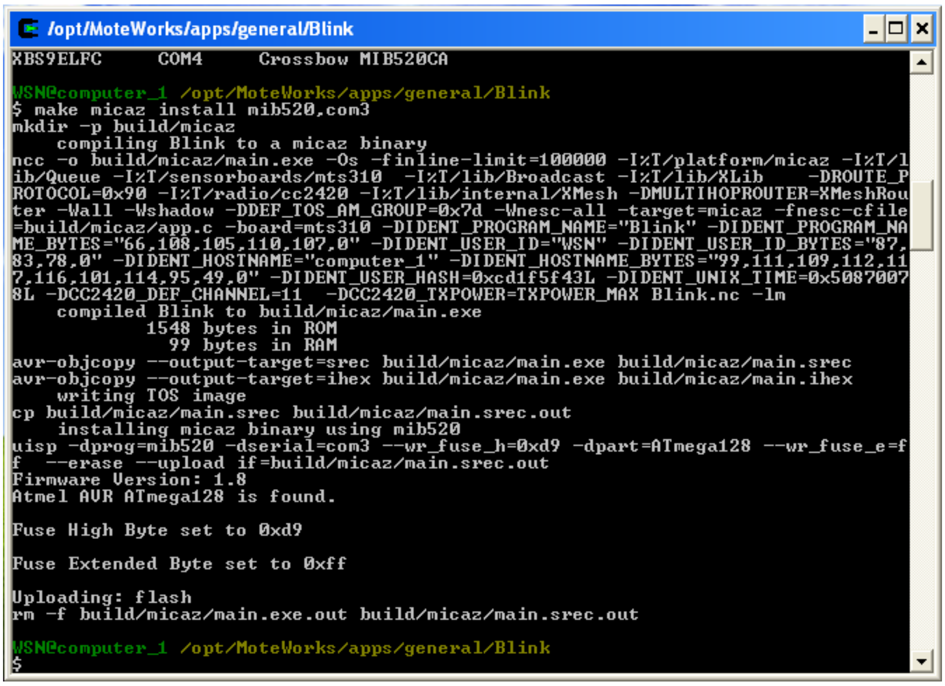
In our case, <platform> is MIB520 (MICAz). Write `motelist` command and you will see the number of the COM ports assigned to the mote:



```
WSN@computer_1 /opt/MoteWorks/apps
$ cd general/

WSN@computer_1 /opt/MoteWorks/apps/general
$ ls
Blink/  CountReceive/  CountSend/  XSniffer/

WSN@computer_1 /opt/MoteWorks/apps/general
$ cd Blink/

WSN@computer_1 /opt/MoteWorks/apps/general/Blink
$ ls
Blink.nc*    Makefile*            README*           build/
BlinkM.nc*   Makefile.component*  SingleTimer.nc*

WSN@computer_1 /opt/MoteWorks/apps/general/Blink
$ motelist
Reference   CommPort   Description
----------  ----------  ----------------------------------------
XBS9ELFC      COM3     Crossbow MIB520CA
XBS9ELFC      COM4     Crossbow MIB520CA

WSN@computer_1 /opt/MoteWorks/apps/general/Blink
$
```

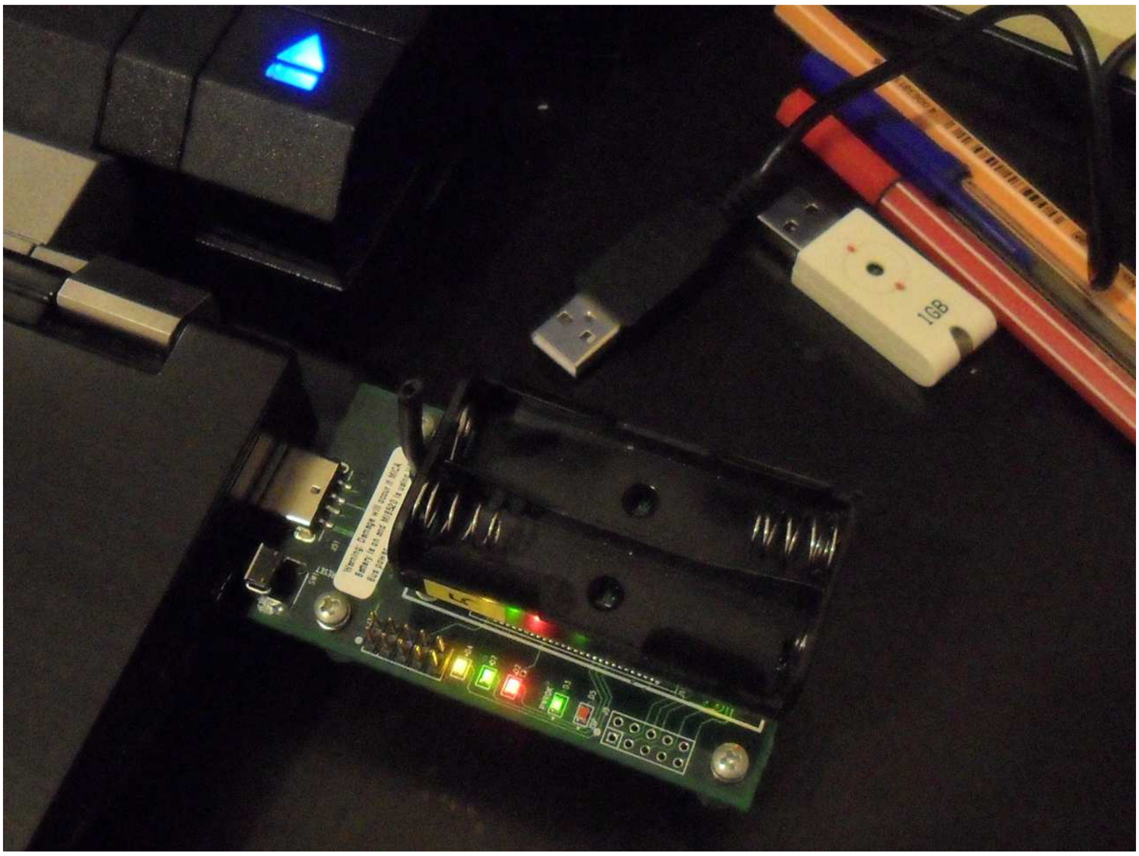
This example is for programming a MICAz from a MIB510 that is connected to a PC's serial port COM3.

```
make micaz install mib520,com3
```



```
XBS9ELFC      COM4      Crossbow MIB520CA

WSN@computer_1 /opt/MoteWorks/apps/general/Blink
$ make micaz install mib520,com3
mkdir -p build/micaz
    compiling Blink to a micaz binary
ncc -o build/micaz/main.exe -Os -finline-limit=100000 -I%T/platform/micaz -I%T/l
ib/Queue -I%T/sensorboards/mts310  -I%T/lib/Broadcast -I%T/lib/XLib    -DROUTE_P
ROTOCOL=0x90 -I%T/radio/cc2420 -I%T/lib/internal/XMesh -DMULTIHOPROUTER=XMeshRou
ter -Wall -Wshadow -DDEF_TOS_AM_GROUP=0x7d -Wnesc-all -target=micaz -fnesc-cfile
=build/micaz/app.c -board=mts310 -DIDENT_PROGRAM_NAME="Blink" -DIDENT_PROGRAM_NA
ME_BYTES="66,108,105,110,107,0" -DIDENT_USER_ID="WSN" -DIDENT_USER_ID_BYTES="87,
83,78,0" -DIDENT_HOSTNAME="computer_1" -DIDENT_HOSTNAME_BYTES="99,111,109,112,11
7,116,101,114,95,49,0" -DIDENT_USER_HASH=0xcd1f5f43L -DIDENT_UNIX_TIME=0x5087007
8L -DCC2420_DEF_CHANNEL=11  -DCC2420_TXPOWER=TXPOWER_MAX Blink.nc -lm
    compiled Blink to build/micaz/main.exe
            1548 bytes in ROM
              99 bytes in RAM
avr-objcopy --output-target=srec build/micaz/main.exe build/micaz/main.srec
avr-objcopy --output-target=ihex build/micaz/main.exe build/micaz/main.ihex
    writing TOS image
cp build/micaz/main.srec build/micaz/main.srec.out
    installing micaz binary using mib520
uisp -dprog=mib520 -dserial=com3 --wr_fuse_h=0xd9 -dpart=ATmega128 --wr_fuse_e=f
f  --erase --upload if=build/micaz/main.srec.out
Firmware Version: 1.8
Atmel AVR ATmega128 is found.

Fuse High Byte set to 0xd9

Fuse Extended Byte set to 0xff

Uploading: flash
rm -f build/micaz/main.exe.out build/micaz/main.srec.out

WSN@computer_1 /opt/MoteWorks/apps/general/Blink
$
```

Now, the nesC application is installed and running in the mote:



REFERENCES

[1] Oracle VirtualBox Webpage -- https://www.virtualbox.org

[2] TinyOS Webpage -- http://www.tinyos.net

[3] nesC compiler Sourceforge Webpage -- http://nescc.sourceforge.net

[4] MoteWorks Getting Started Guide 7430-0102-01 D – Crossbow guide link