Establishing a Connection

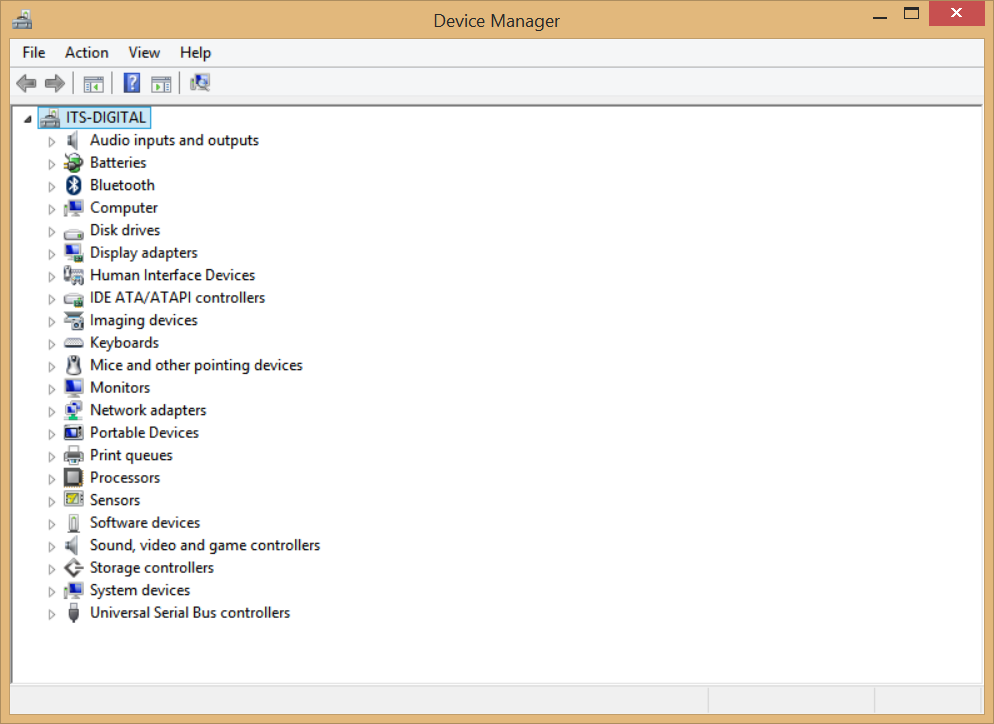
This ~~section of the report~~ tutorial will explain the process of establishing a Bluetooth connection between Mathworks® MATLAB® and the Lego® Mindstorms® NXT ‘Brick’. The Lego® Mindstorms® NXT ‘Brick’ will simply be referred to as the *Brick*, and Mathworks® MATLAB® simply as *MATLAB*, from this point onwards.

This section focuses on setting up a connection to the Brick from a 64-bit Windows operating system.

# Establishing a Wired Connection

A wired connection has to be made before a Bluetooth connection can be successfully established for future use.

**Note:** Before proceeding, enter Device Manager and ensure that all drivers (Bluetooth or USB) that are associated with the Brick remain uninstalled or are deleted.



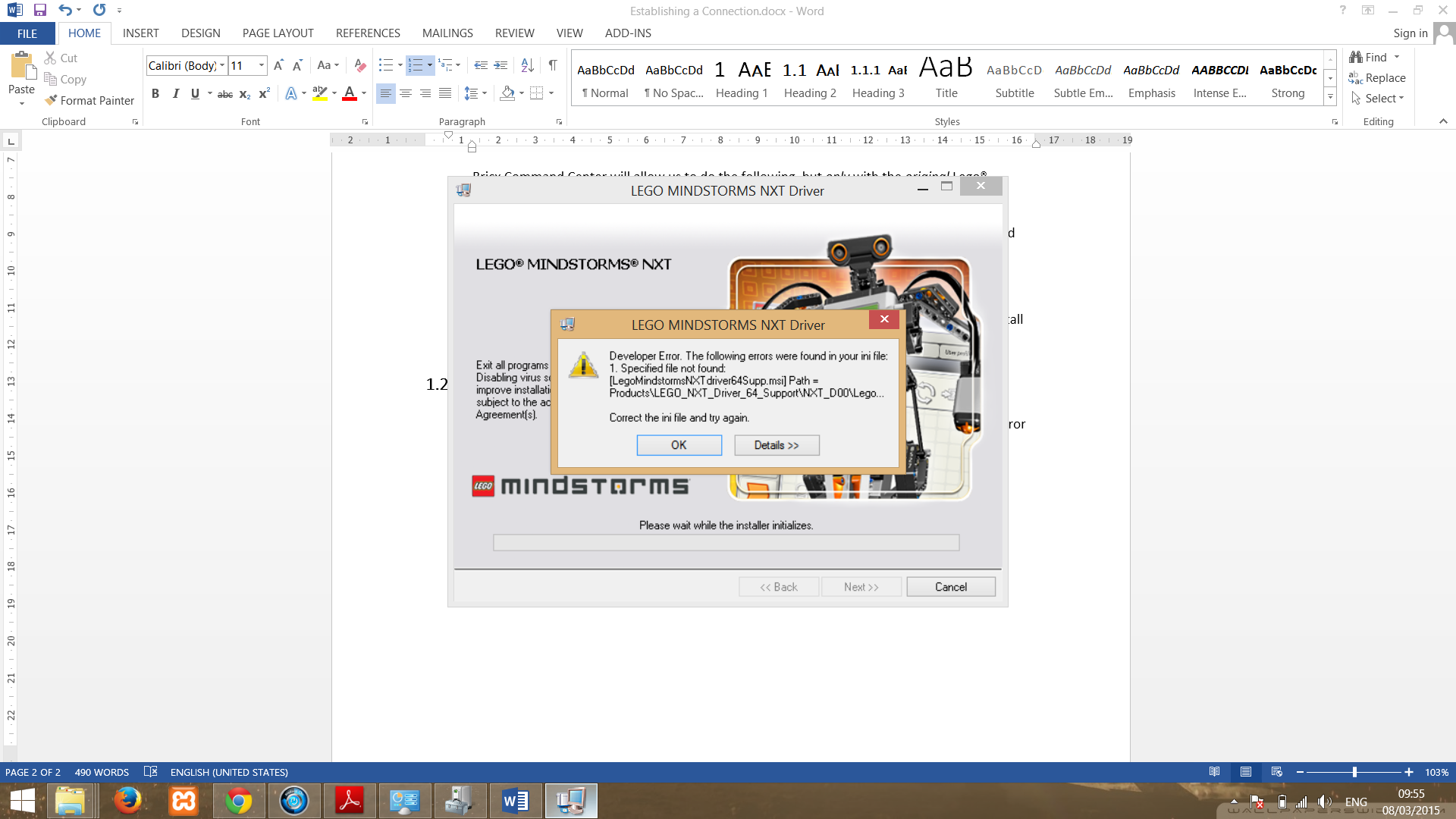
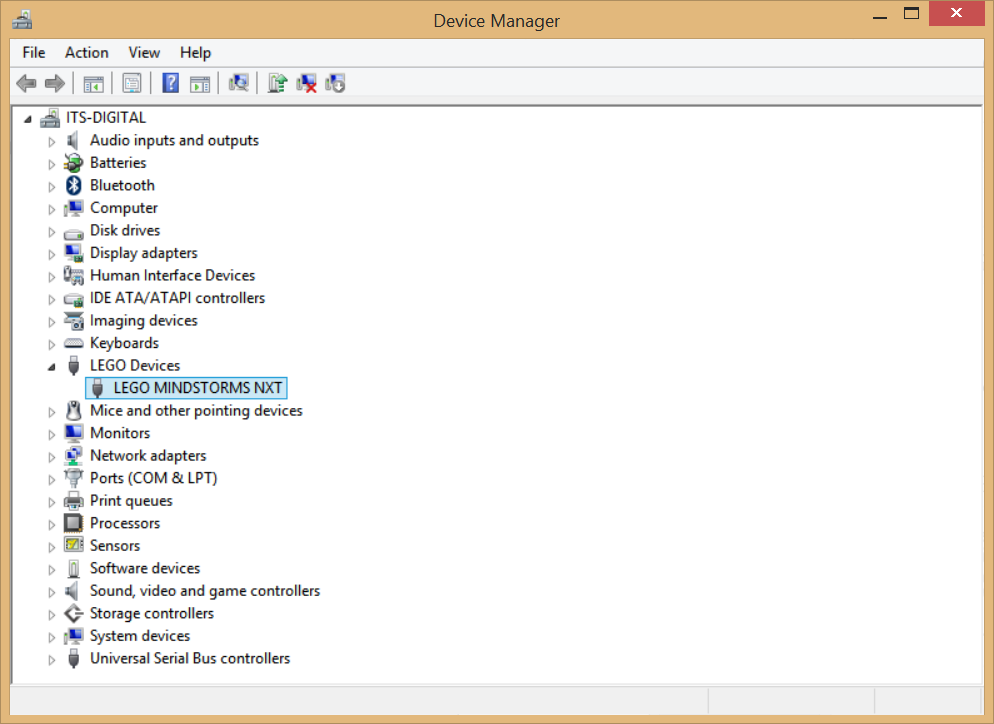
## Acquire Software/Firmware

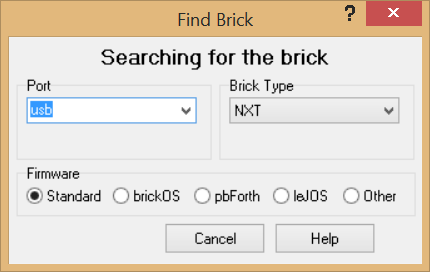
Download the following software:

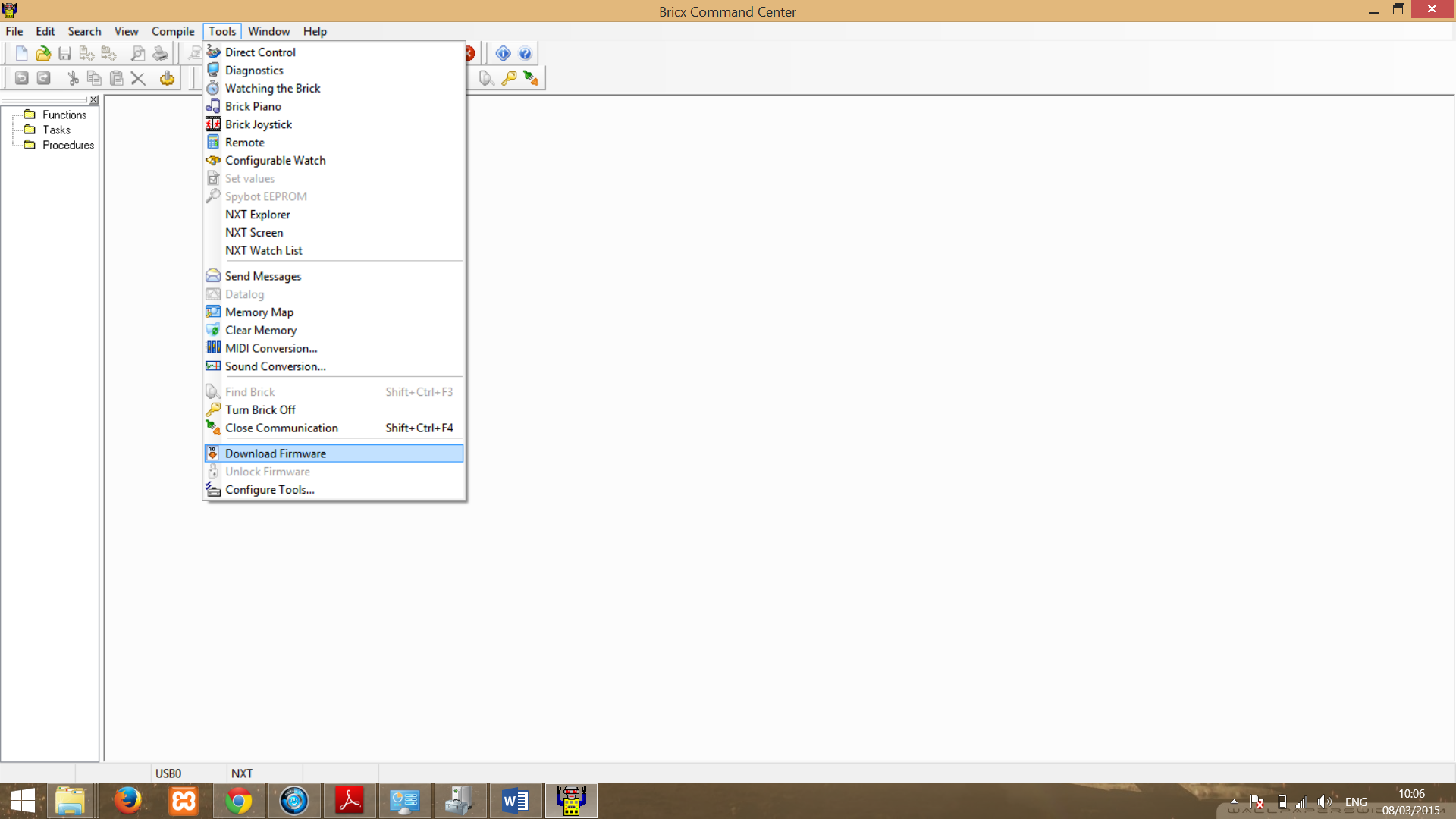
* **Firmware for the Lego® NXT® Brick**The Official Lego® Mindstorms® NXT Brick firmware; the *latest* official firmware is made available on the official Lego® Mindstorms® website. Extract the \*.RFW file from the ZIP archive and place in a new folder labelled “Lego NXT Firmware”.
* ***Original* Driver for the Lego® NXT® Brick**The driver files are collectively referred to as the “NXT Fantom Driver” and are made available on the official Lego® Mindstorms® website. Download the *latest* version and extract the *Windows* driver files from the ZIP archive, place them in a folder labelled “Lego NXT Original Driver”.
* **RWTH - Mindstorms NXT Toolbox for MATLAB**This is the toolbox/library that will allow us to communicate with (and hence, control) the Brick from the MATLAB environment.  
  Download the latest version from the official website and extract the contents to a folder named “RWTH”.
* **‘libusb’ Library and Driver Package**‘libusb’ is an open-source C library that allows applications to easily access USB devices on various operating systems. The *RWTH - Mindstorms NXT Toolbox for MATLAB* utilizes libusb to communicate with the Brick on the MATLAB® environment.  
  Hence, download the latest version of the ‘libusb’ package from the official website, and extract the contents of the ZIP archive to a new folder labelled “libusb”.
* **Bricx Command Center Application**Download and *install* the *latest* *version* of the Bricx Command Center application.  
    
  Bricx Command Center will allow us to do the following, but *only* with the *original* Lego® Mindstorms® NXT driver:
  + **Test Motors and Sensors** – allows direct control of motors via a GUI.
  + **Act as screen replacement** – Data sent to screen is sampled at chosen frequency and displayed on an LCD *emulator* – useful if screen is damaged. **See Appendix.**
  + **Upload new firmware to the Brick.**
  + **Add/Remove/Explore programs installed on the Brick.**

We will mainly use the Bricx Command Center to upload new firmware to the Brick, and install the RWTH MotorControl *NXC* program on the Brick.

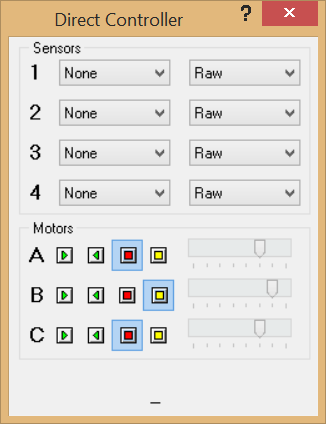
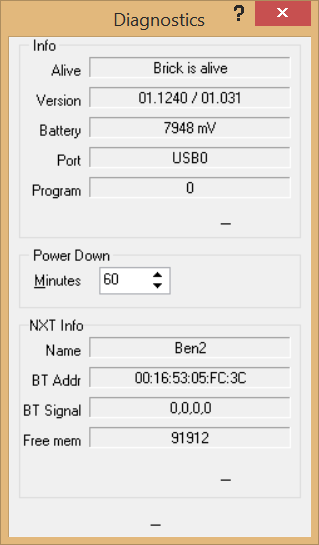
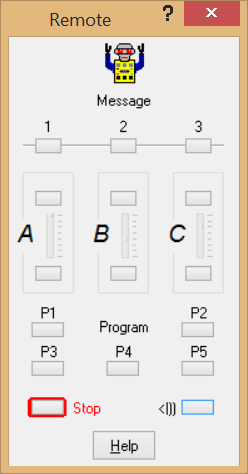
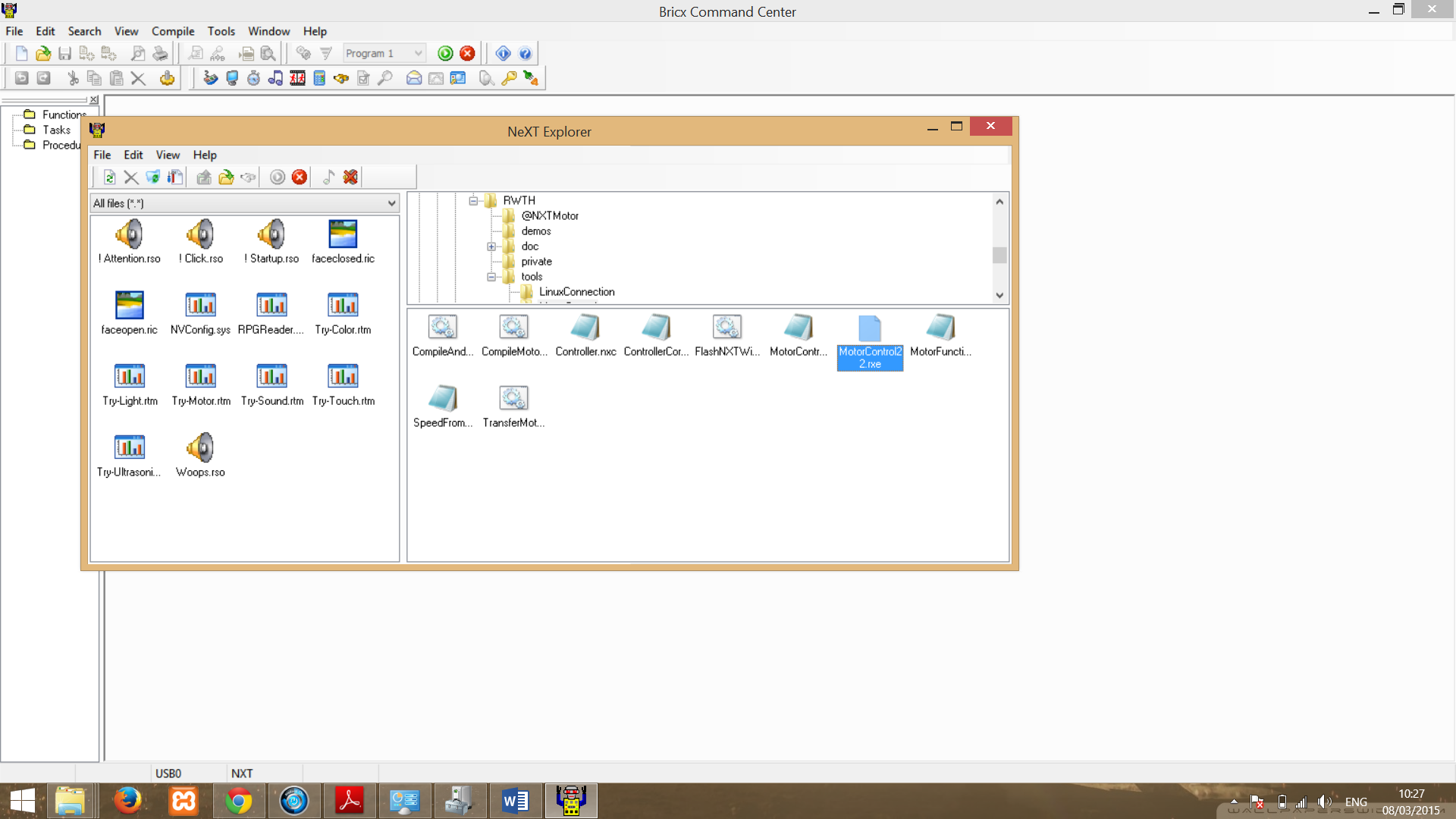
## Set up the Brick for MATLAB

1. In the “Lego NXT Original Driver” directory, execute “Setup.exe” *as administrator*.   
   There may a chance that “Setup.exe” returns an error message upon execution due to an error in the settings file packaged with the installation executable.  
     
   In this case, find “LEGO\_NXT\_Driver\_64” folder under the “Products” directory and execute “LegoMindstormsNXTdriver64.msi” instead.
2. After the installation is successful, connect the Brick to the machine using a USB cable. Using ‘Device Manager’, you should find that the Brick has been detected and associated with the Lego® Mindstorms® NXT driver under “LEGO devices”. If this is *not* the case, use ‘Device Manager’ to update the driver, uninstalling and repeating the process if necessary.  
   
3. Run Bricx Command Center as administrator.



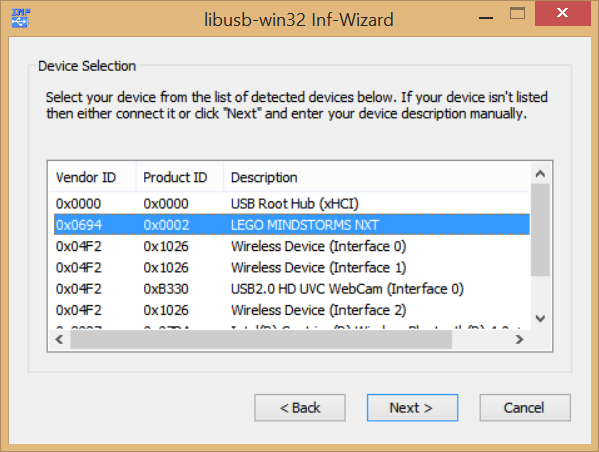
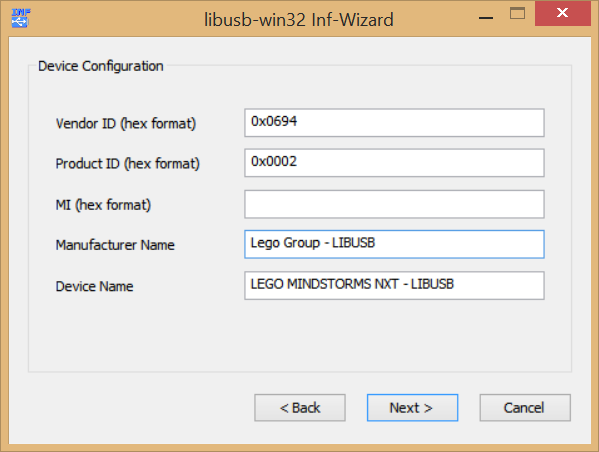
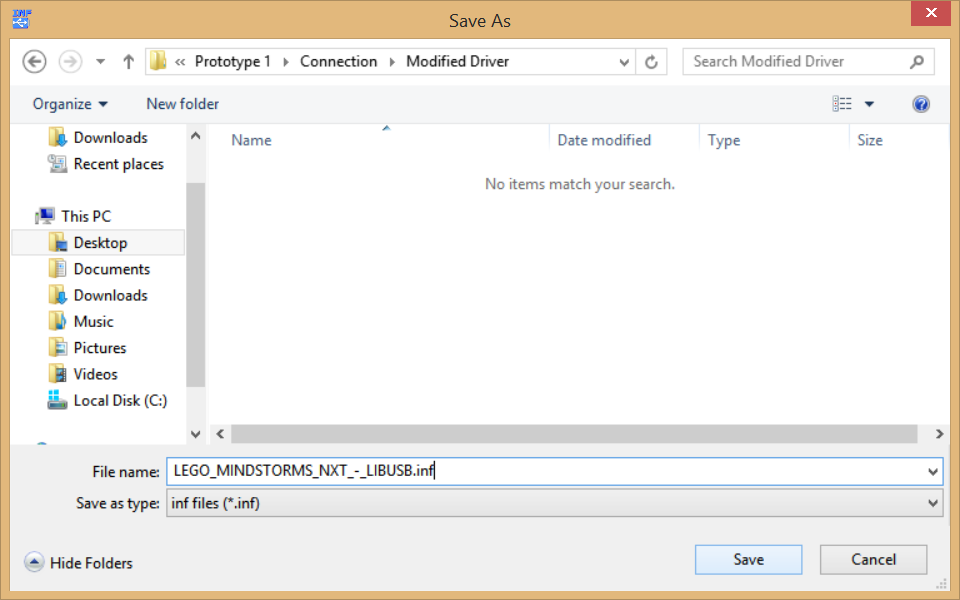
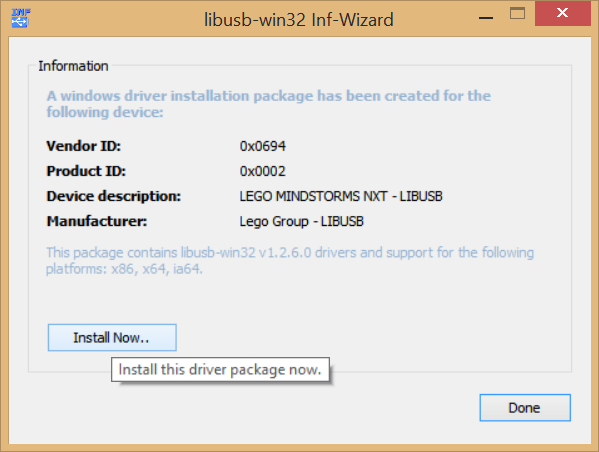
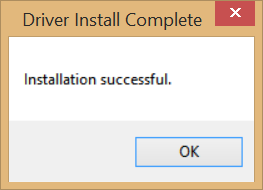


In the menu bar, under “Tools”, click “Download Firmware”, and point to the location of the latest firmware. Firmware update should begin immediately, and upon completion, the Brick will reboot.

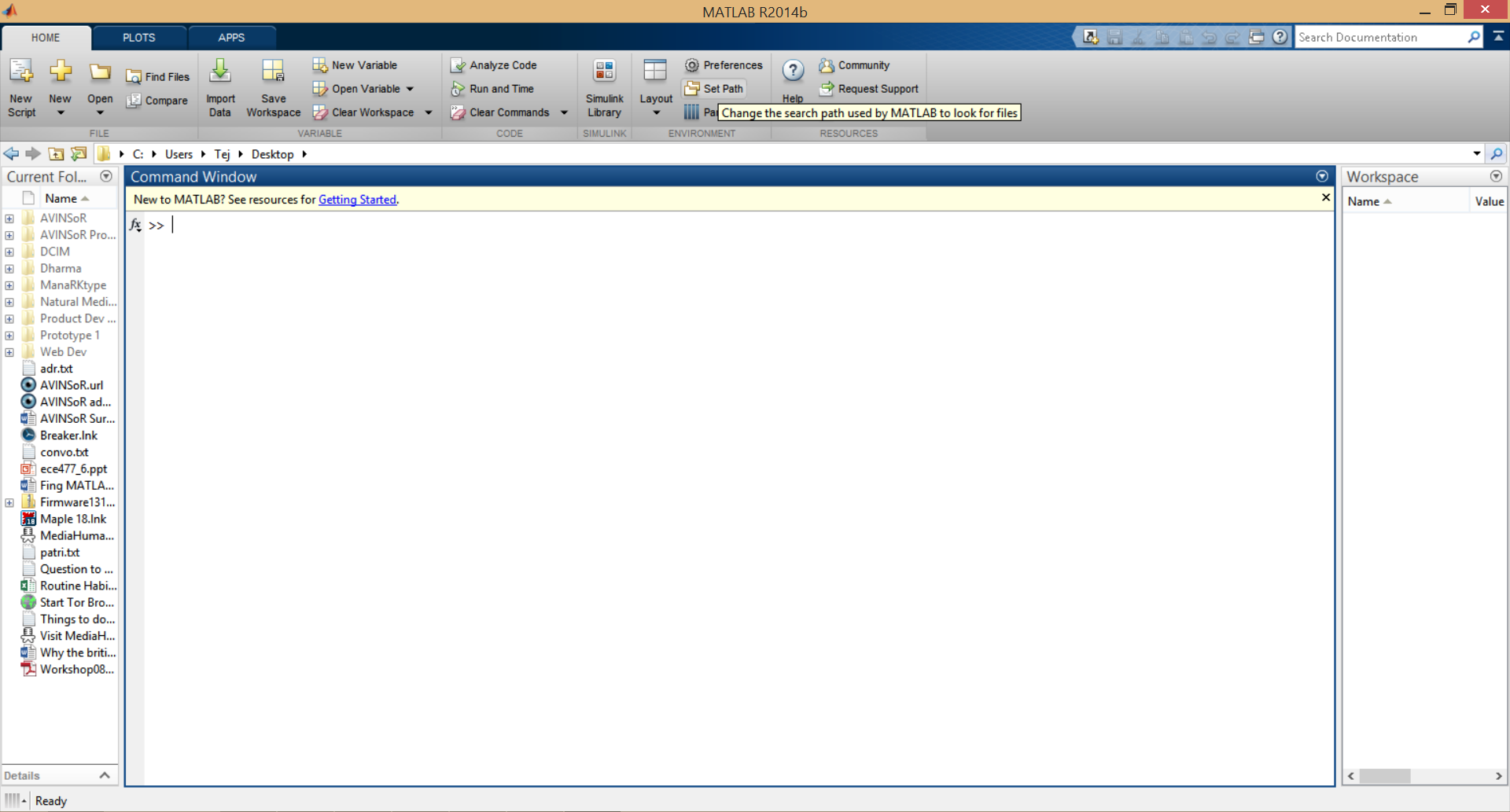
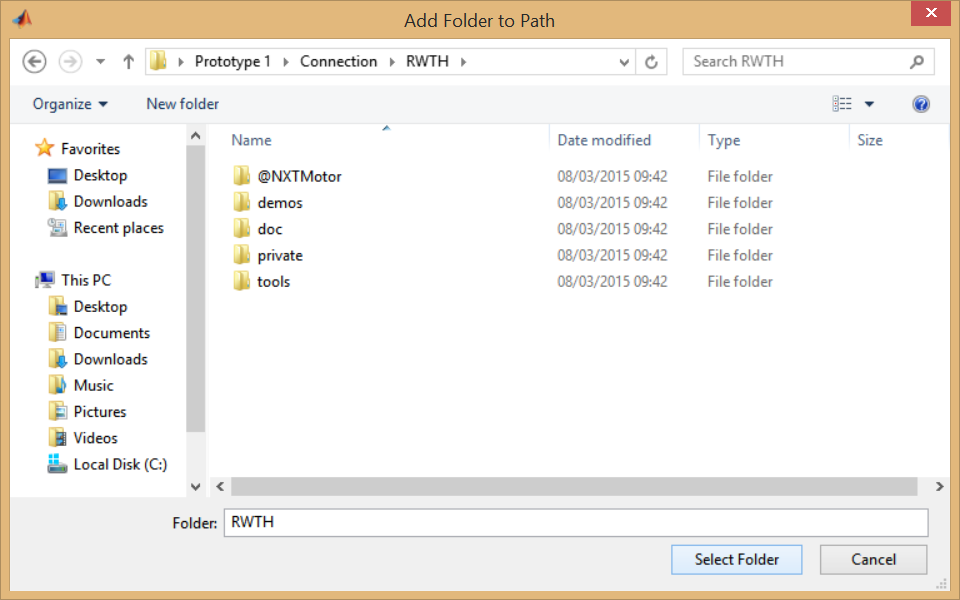
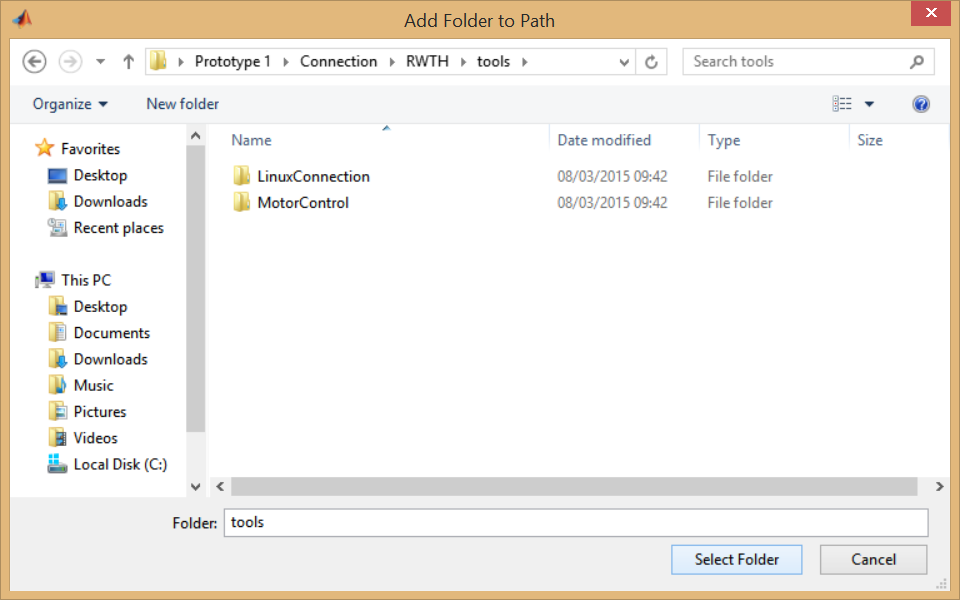
1. Use “Diagnostics” to view information about the Brick, and connectivity data. Use “Remote” and “Direct Control” under tools to test your assembly (i.e. the motor and sensor connections to the Brick).   
     
2. Use “NeXT Explorer” under tools to delete all existing NXT executables (\*.RXE files) installed by the firmware. Install the Motor Control RXE executable provided by RWTH toolbox (“MotorControl22.RXE”) – find “MotorControl\*\*.RXE” using the file explorer and click the ‘download’ icon. Defragment the Brick to optimize performance; click “Files” under the menu bar and find “degragment”.  
     
   The Motor Control RXE executable is executed on the Brick to allow it read and act upon the motor control commands sent from MATLAB via the RWTH toolbox methods.  
   
3. **If the screen is not working:** this is a good time to use the virtual LCD screen (“NXT Screen”, under “tools” in the main menu) to:
   1. Switch Bluetooth on and make the device *visible*. Rename the device and reset the PIN if necessary.
   2. Execute the Motor Control app.

## Modify Original Driver to allow MATLAB connectivity

Use **inf-wizard** provided by ‘libusb’ to create a *wrapper* *driver* around the original Lego® Mindstorms® NXT driver in order for RWTH to communicate with the Brick on the 64-bit Windows platform

1. Find inf-wizard.exe in the “bin” folder under the main “libusb” directory and execute *as administrator*. Make sure the Brick is still active and connected to the PC (use device manager to check, if necessary). Select the active driver and click ‘Next’.   
   
2. Do not keep the original driver name, as this will not allow us to differentiate between which driver (original *or* ‘libusb’-modified) is active/installed, if ever required to do so in the future.  
     
     
     
     
   Note that the installation *will* fail if ‘libusb’ is not executed *as administrator*, or if ‘driver signature enforcement’ has not been disabled in Windows settings. **See Appendix.**

## Install the Toolbox

1. Run MATLAB and click the “Set Path” icon.  
     
     
   
2. Add the “RWTH” folder, and the “RWTH\tools” folder, to the list of paths and click “Save” before closing.
3. Now that RWTH Toolbox has been *installed*.

## Test the Toolbox Connectivity.

Create a new folder labelled “CODE” (or whatever name you prefer) and set as the working MATLAB path. Either copy TestRoutine.m directly, or use the source code below to create the file. The routine will allow you to easily test MATLAB-Brick connectivity. Make sure you answer with the “n” key when prompted to connect using Bluetooth.

% \*\* RESET \*\*

COM\_CloseNXT**(**'all'**);** % Close all previous COM-port/NXT connections (if any)

close all**;** % Close all figures/plots.

clear all**;** % Clear all variables

clc**;** % Clear the command screen.

% \*\* Establish a connections \*\*

a **=** input**(**'Is this a Bluetooth connection? (y/n)'**,**'s'**);**

**if** strcmpi**(**a**,**'n'**)**

% establish a WIRED connection

hNXT **=** COM\_OpenNXT**;**

**else**

% establish a wireless BLUETOOTH connection

hNXT **=** COM\_OpenNXT**(**'bluetooth.ini'**);**

**end**

COM\_SetDefaultNXT**(**hNXT**);**

% Set the global default handle, so that later on, whenever we're calling functions, we don't have to pass the handle every time.

handle **=** COM\_GetDefaultNXT**;**

% \*\* Test ALL Motors -- GO NUTS!! \*\*

mA **=** NXTMotor**(**'A'**);**

mB **=** NXTMotor**(**'B'**);**

mC **=** NXTMotor**(**'C'**);**

mA**.**Power **=** 100**;** % let's start with the maximum power!

mB**.**Power **=** 100**;**

mC**.**Power **=** 100**;**

mA**.**SendToNXT**();**

mB**.**SendToNXT**();**

mC**.**SendToNXT**();**

fprintf**(**'Press any key to stop.\n'**);**

pause**;** % wait for KeyPress before stopping!

mA**.**Stop**();**

mB**.**Stop**();**

mC**.**Stop**();**

% \*\* Test Sensors \*\*

% Ultrasonic Sensor plugged in Sensor Port 1

% Sound Sensor plugged in Sensor Port 2

% Light Sensor plugged in Sensor Port 3

OpenUltrasonic**(**SENSOR\_1**);**

OpenSound**(**SENSOR\_2**,** 'DB'**);**

OpenLight**(**SENSOR\_3**,** 'ACTIVE'**);**

**for** i **=** 1**:**100

clc**;**

fprintf**(**'Distance (centimetres): %d\n'**,** GetUltrasonic**(**SENSOR\_1**));**

fprintf**(**'Sound Level (decibels): %d\n'**,** GetSound**(**SENSOR\_2**));**

fprintf**(**'Light Intensity (# from 1 to 1023): %d\n'**,** GetLight**(**SENSOR\_3**));**

% prompt user for another reading!

a **=** input**(**'Get another set of sensor readings (y/n)? '**,**'s'**);**

**if** strcmpi**(**a**,**'n'**)**

**break;**

**end**

**end;**

CloseSensor**(**SENSOR\_1**);**

CloseSensor**(**SENSOR\_2**);**

CloseSensor**(**SENSOR\_3**);**

%% \*\* END OF TEST notification \*\* %%

clc**;**

fprintf**(**'END OF TEST ROUTINE\n'**);**

fprintf**(**'Press any key...\n'**);**

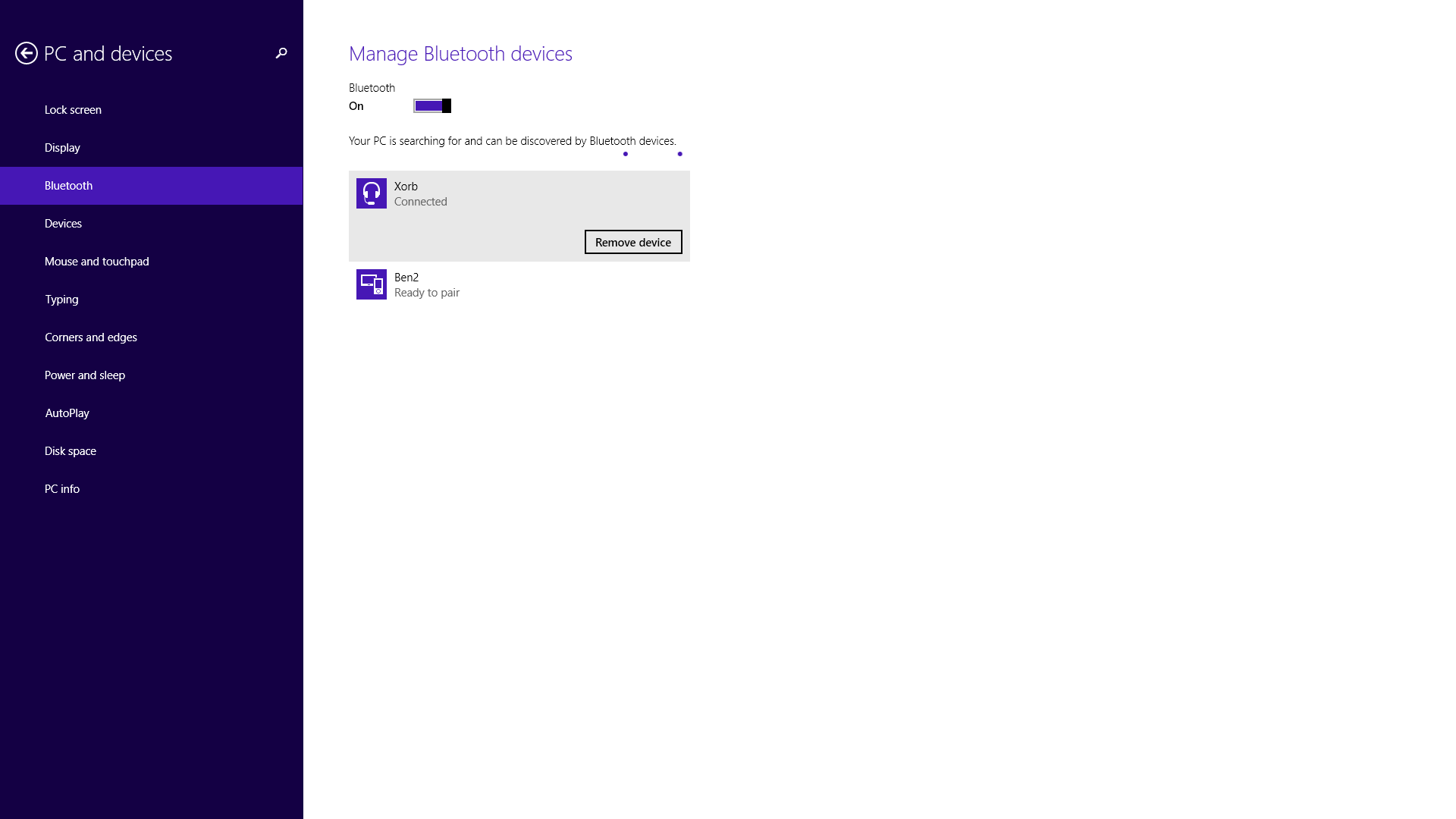
pause**;**

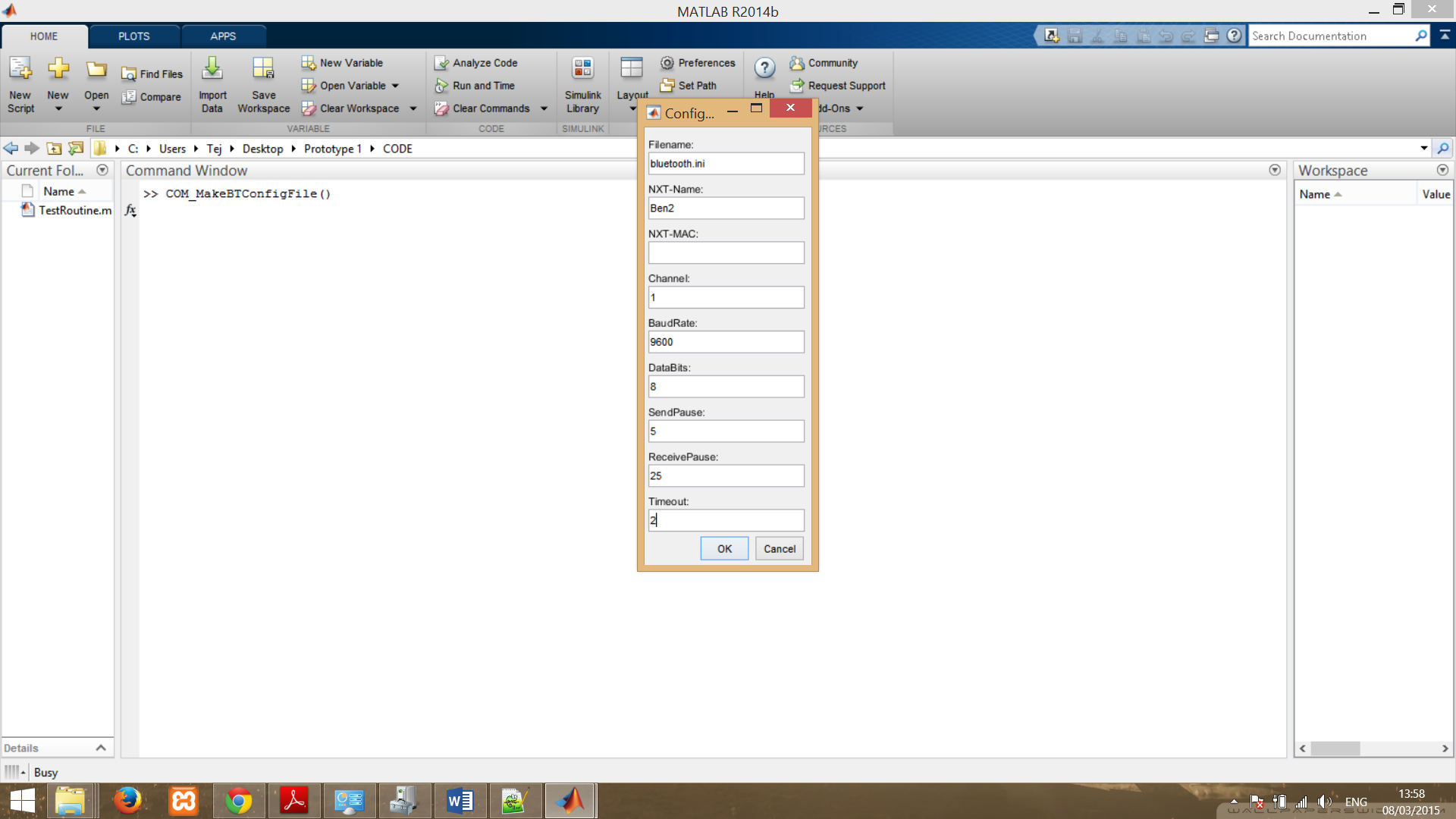
clc**;**

% \*\* Close Connection \*\*

COM\_CloseNXT**(**hNXT**);**

# Establishing a Wireless (Bluetooth) Connection

1. Using your operating system’s Bluetooth connections tool, find and connect to the Brick. The Brick should have the name given earlier. In this case, “Ben2”. The default PIN (if not reset) is usually “0000” or “1234”.  
     
   **If the screen is not working:** When you attempt to connect to the device, the Brick will bleep. At this stage, you should press the **Orange** button. Immediately, the operating system of your machine will ask for a PIN – allowing you to pair device.  
   
2. Once paired, go back into MATLAB. Use the COM\_MakeBTConfigFile()command to create a Bluetooth.ini file in the *working directory* created earlier. On a 64-bit operating system, all that is required in the Bluetooth.ini file is the NXT-Name (in the example given, “Ben2”), everything else can be left as default.



Upon clicking “OK”, the Bluetooth.ini file should appear in the *working directory*.

1. Use the TestRoutine.m script to test the wireless connection. Answer “y” when prompted to connect using Bluetooth.