Steps to install pc-ble-driver-js (32-bit)

This is for the user interface

Sources:

<https://github.com/NordicSemiconductor/pc-ble-driver-js>

<https://github.com/NordicSemiconductor/pc-ble-driver/blob/master/README.md>

Note: All steps starting with $ are commands for Command Prompt (CMD), so no need to copy “$”

1. Download Visual Studio 2017

<https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&rel=15>

1. In Visual Studio Installer, select and install
2. Universal Windows Platform development
3. Desktop development with C++
4. Install Chocolatey. Install with cmd.exe (Run as administrator and copy everything below)

@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"

1. Add to Chocolatey to Environment Variables

$ SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"

1. Install Git

$ choco install -y git

1. Install CMake

$ choco install -y cmake

1. Install vcpkg

$ git clone https://github.com/Microsoft/vcpkg.git

$ cd vcpkg

$ .\bootstrap-vcpkg.bat

1. Intall Ninja

$ choco install -y ninja

1. Install GNU Embedded Toolchain for Arm

$ choco install gcc-arm-embedded

1. Uninstall nodejs if you have it
2. Install nodejs v8.11.2 for 32-bit [node-v8.11.2-x86.msi](https://nodejs.org/download/release/v8.11.2/node-v8.11.2-x86.msi)
3. Download BOOST [boost\_1\_69\_0.7z](https://dl.bintray.com/boostorg/release/1.69.0/source/boost_1_69_0.7z) and unzip under C:\boost\boost\_1\_69\_0
4. Build Boost (step iv will take a while)
5. $ cd C:\boost\boost\_1\_69\_0
6. $ bootstrap.bat
7. Edit project-config.jam file:

import option ;

using msvc : 14.0 : "C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\VC\Tools\MSVC\14.16.27023\bin\Hostx86\x86\cl.exe";

option.set keep-going : false ;

1. $ b2 toolset=msvc-14.1 --build-dir=.x86 --stagedir=stage\_x86
2. $ cd C:\Users\Admin\Desktop\puck capstone 2019
3. $ git clone <https://github.com/NordicSemiconductor/pc-ble-driver-js.git>
4. $ cd pc-ble-driver-js
5. Delete pc-ble-driver folder
6. $ git clone <https://github.com/NordicSemiconductor/pc-ble-driver.git>
7. $ cd pc-ble-driver-js\pc-ble-driver
8. $ mkdir build && cd build
9. $ vcpkg install asio
10. $ vcpkg install catch2
11. Switch from CMD to x86 Native tools Command prompt for VS 2017 and change into \pc-ble-driver
12. Open C:\Users\Admin\Desktop\puck capstone 2019\pc-ble-driver-js\pc-ble-driver\build\CMakeCache.txt

Change

1. ASIO\_INCLUDE\_DIR:PATH=C:\Users\Admin\Desktop\puck capstone 2019\vcpkg\installed\x86-windows\include
2. Catch2\_DIR:PATH=C:\Users\Admin\Desktop\puck capstone 2019\vcpkg\installed\x86-windows\share\Catch2
3. $ cmake -G Ninja ..
4. $ cmake --build .
5. Should build without errors
6. Cd back to \pc-ble-driver-js
7. $ npm config set cmake\_CMAKE\_GENERATOR:INTERNAL="Visual Studio 15 2017"
8. $ npm config set cmake\_BOOST\_ROOT=c:\boost\boost\_1\_69\_0
9. Add .npmrc file in \pc-ble-driver-js folder(builds dependencies for running driver on Electron)

runtime = electron

target = 2.0.17

disturl = <https://atom.io/download/electron>

1. $ npm install
2. $ npm test
3. Still in \pc-ble-driver-js, edit package.json file:

{

"name": "your-app",

"version": "0.1.0",

"main": "main.js",

"scripts": {

"start": "electron .",

"package-mac": "electron-packager . --overwrite --platform=darwin --arch=x64 --icon=assets/icons/mac/icon.icns --prune=true --out=release-builds",

"package-win": "electron-packager . electron-tutorial-app --overwrite --asar=true --platform=win32 --arch=ia32 --icon=assets/icons/win/icon.ico --prune=true --out=release-builds --version-string.CompanyName=CE --version-string.FileDescription=CE --version-string.ProductName=\"Electron Tutorial App\"",

"package-linux": "electron-packager . electron-tutorial-app --overwrite --asar=true --platform=linux --arch=x64 --icon=assets/icons/png/1024x1024.png --prune=true --out=release-builds"

}

}

1. Install fast-csv

$ npm install fast-csv

1. Install Electron

$ npm install --arch=ia32 --save-dev electron@2.0.17

1. Start program

$ npm start

At this point the environment is ready for development. The main.js file is where the electron browser is defined and can call other js files.

1. To publish the application as an executable, on the command line enter

npm run package-win