Nordic Semiconductor Toolchain Setup

Helpful Nordic tutorial: [devzone](https://devzone.nordicsemi.com/tutorials/b/getting-started/posts/development-with-gcc-and-eclipse)

*Windows command line toolchain with GCC and Make*

*Installers: \\MISSOURI\SharedDocs\applications\NordicToolChain*

For this example, the Nordic Semiconductor tools were obtained off the downloads tab of the main page for the nRF52840 board, located at “https://www.nordicsemi.com/eng/Products/nRF52840”

1. Install appropriate SDK from Nordic Semiconductors – Current version released week 27 of 2017 (first week of July) is nRF5 SDK v13.1.0, this version is necessary to use the S140 V5.0.0-2.alpha soft device which is the first “soft device” available with Bluetooth 5 support.
   1. Download zip file from Nordic Semiconductors website
   2. Extract to SDK folder – Currently using C:\Nordic
      1. Maybe split into versions in case it is desirable to have multiple versions of the SDK installed in the future?
2. Install Nordic Semi command line tools via install executable (version used was, “nRF5x-Command-Line-Tools\_9\_6\_0\_Installer.exe”)
   1. Make sure location of embedded tools—which was “C:\Program Files\Nordic Semiconductor\nrf5x\bin\” for this computer—is added to the Path variable
3. Install GNU Arm Embedded Toolchain (Standard version of GCC does not include support for embedded processors), this can currently be found at “https://launchpad.net/gcc-arm-embedded”
   1. Make sure the system or user path variable is updated with the location of the \bin folder of the GNU ARM embedded toolchain, for this install this was “C:\Program Files\GNU Tools ARM Embedded\5.4 2016q3\bin”
   2. Update Makefile.windows found in “<nRF52\_SDK\_INSTALL>\components\toolchain\gcc” with path and version of GNU ARM embedded toolchain
4. Install GNU Make if it is not already installed
5. To program and run a project that does not require a soft device
   1. Compile an example in the SDK by opening a terminal, navigating to “<SDK>/examples/peripheral/<example name>/<board name>/blank/armgcc/” and entering “make” without quotes. Board name used currently is pca10056.

* 1. Use the nrfjprog command line tool to program the compiled project onto the target board. This should be run from within the same folder that “make” was run from in the previous step. Note: replace nRF52 in all the following commands with the family of your target device, and change the name of the hex file in step b to the hex file output from running make in step 5. For this example, the target board was the nRF52840 development kit board, which is part of the nRF52 family. Note, if commonly only using one family of device, a default value can be set in the nrfjprog.ini file, which will prevent it from being necessary to type “—family nRF5x” for every command. This file resides in the directory where the command line tools were installed, which is “C:\Program Files\Nordic Semiconductor\nrf5x\bin” on this (32-bit) machine.
     1. Erase target:\> nrfjprog --family nRF52 –e
     2. Program target:\> nrfjprog --family nRF52 --program \_build\nrf52840\_xxaa.hex
     3. Reset target and run application:\> nrfjprog --family nRF52 –r

1. To program and run a project that does require a soft device (Which is any of the BLE projects)
   1. Compile and program the soft device onto the target
      1. Erase target as in step 5.b
      2. Run “make flash\_softdevice” in the armgcc folder referenced above, this will compile the soft device, erase the target, and load the soft device onto the target.
   2. Compile and program the progect
      1. Run make, as in step 5.
      2. Use the nrfjprog tool as in step six, omitting the erase step
         1. Program target:\> nrfjprog --family nRF52 --program \_build\nrf52840\_xxaa.hex
         2. Reset target and run application:\> nrfjprog --family nRF52 –r
   3. Alternative method that was discovered after viewing the makefile
      1. Build project:\> make
      2. Erase target:\> make erase
      3. program softdevice:\> make flash\_softdevice
      4. Program and reset target and run application:\> make flash
2. The application should now be running on the target device

Here is my setup document. A couple notes:

* The nrf52832 is pca10040 not pca10056 (which is the nrf52840), so everywhere this says 56 replace with 40 for programming the puck
* The more recent version of the arm embedded gcc compiler may not work, version 4.9 does for sure
  + I put installers for the version of gnu make and arm gcc that I am using in Shareddocs\applications as we discussed
* The note about nordic sdk versions is old, my system is setup up like C:\Nordic\<sdk version>
* I am not sure if the nrfjprog needed python. I don't think so, but it should give a helpful error if it does. If you do need to install python, install the latest version of python 2 (should be 2.7.something), python 3 does not work with the applications for the dfu stuff that we will probably be working with later

Another note. By reading the makefiles I discovered that Nordic added shortcuts so you don't have to type out all of the nrfjprog commands, and I have been using the makefile shortcuts mostly.

How I program the puck now:

* Peripheral application
  + Navigate to <example directory>\pca10040\blank\armgcc
  + Run "make"
  + Run "make erase"
  + Run "make flash"
* Bluetooth application
  + Navigate to <example directory>\pca10040\s132\armgcc
    - Run "make"
    - Run "make erase"
    - Run "make flash\_softdevice"
    - Run "make flash"