**Jenkins Build LPC based Code**

# Setup – watch the install locations so you don’t have to type long command lines

* + 1. Install MCUxpresso
    2. Install Jenkins
    3. Install GIT
    4. Install Python
    5. Git build instructions and patches from: https://github.com/RussellBauer/LPC\_Automation.git

1. Configure Jenkins – steps are the same for each project type
2. Click **New Item**
   1. Name project (G5\_IM, G5\_BC or G55\_BC)
   2. Click Freestyle Project then OK at bottom
3. **General**
   1. Click on newly created project
   2. Click Configure
      1. Description – Whatever you want
   3. Click GITHub Project
      1. Add url: https://github.com/DellESI/dcsg5-im.git/ or https://github.com/DellESI/dcsg5-bc.git/
4. **Source Code Management**
   1. Click GIT
   2. Add url: https://github.com/DellESI/dcsg5-im.git/ or https://github.com/DellESI/dcsg5-bc.git/
   3. Add Credentials: I used mine, need to work on a machine level credentials
5. **Build Triggers**
   1. Add as needed
6. **Build Environment**
   1. Click Delete workspace before build starts
7. **Build**
   1. Click Windows Batch File
   2. Add: (Common for all builds. Each section should be a single line, modify paths to match your install)

rem Batch file to build an Eclipse project from the command line on Windows

rem Example for using MCUXpresso IDE

rem path to GNU tools and compiler: arm-none-eabi-gcc.exe, ....

SET TOOLCHAIN\_PATH=C:\nxp\MCUXpressoIDE\_10.0.0\_344\ide\tools\bin;C:\Users\Russell\_Bauer\AppData\Local\Programs\Python\Python36\

rem variable to the command line Eclipse IDE executable

SET IDE=C:\nxp\MCUXpressoIDE\_10.0.0\_344\ide\mcuxpressoidec.exe

ECHO Extending PATH if not already present

ECHO %PATH%|findstr /i /c:"%TOOLCHAIN\_PATH:"=%">nul || set PATH=%PATH%;%TOOLCHAIN\_PATH%

* 1. **Add the appropriate build instructions…**
     1. **FOR IM BUILD:**

"%IDE%" -nosplash --launcher.suppressErrors -application org.eclipse.cdt.managedbuilder.core.headlessbuild -importAll "%WORKSPACE%"\software\embedded\nxp\_lpc\_18xx\source\fs\_op\_image\ -data \g5WS\G5\_IM\ -cleanBuild OpImage\_IM/Release

python \sls\_source\utils\utility\_1.0.py "%WORKSPACE%"\software\embedded\nxp\_lpc\_18xx\source\fs\_op\_image\opimage\_im\release\OpImage\_im.bin A3 00 00 %BUILD\_NUMBER%

* + 1. **FOR G5 BC:**

ECHO Patch the files...

git apply --reject --ignore-space-change --ignore-whitespace c:\jenkins\workspace\g5\_bc.patch

"%IDE%" -nosplash --launcher.suppressErrors -application org.eclipse.cdt.managedbuilder.core.headlessbuild -importAll "%WORKSPACE%"\software\embedded\nxp\_lpc\_18xx\source\fs\_op\_image\ -data \g5WS\G5\_BC\ -cleanBuild OpImage\_BC/Release

python \sls\_source\utils\utility\_1.0.py "%WORKSPACE%"\software\embedded\nxp\_lpc\_18xx\source\fs\_op\_image\opimage\_bc\release\OpImage\_BC.bin A3 00 00 %BUILD\_NUMBER% G5

* + 1. **FOR G55 BC:**

ECHO Patch the files...

git apply --reject --ignore-space-change --ignore-whitespace c:\jenkins\workspace\g55\_bc.patch

"%IDE%" -nosplash --launcher.suppressErrors -application org.eclipse.cdt.managedbuilder.core.headlessbuild -importAll "%WORKSPACE%"\software\embedded\nxp\_lpc\_18xx\source\fs\_op\_image\ -data \g5WS\G55\_BC\ -cleanBuild OpImage\_BC/Release

python \sls\_source\utils\utility\_1.0.py "%WORKSPACE%"\software\embedded\nxp\_lpc\_18xx\source\fs\_op\_image\opimage\_bc\release\OpImage\_bc.bin A3 00 00 %BUILD\_NUMBER% G55