Using ST-Link Utility inside Atollic TrueSTUDIO

2014.09.11

Using ST-Link Utility inside Atollic TrueSTUDIO

Requirements

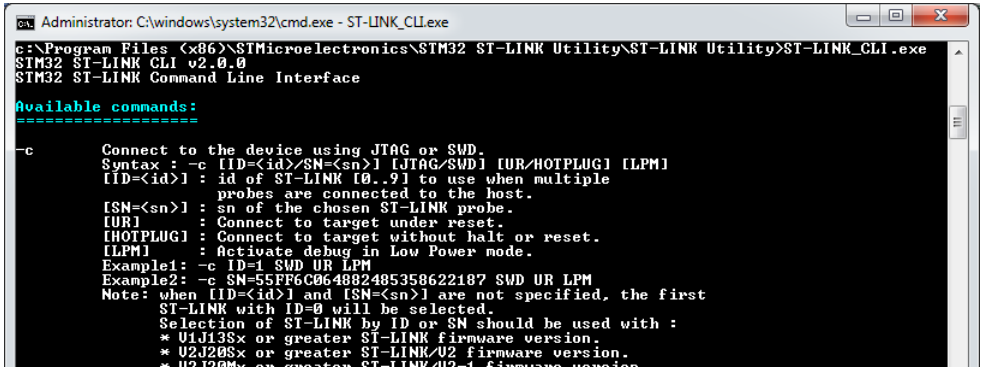
- Atollic TrueSTUDIO
- St-Link Utility (Download from http://www.st.com)
- ST-Link
- ST-Link utility does not support elf-files. Use Intel Hex.

Why use ST-Link Utility inside Atollic TrueSTUDIO

The ST-Link GDB-server used for debugging STM32 devices does not implement all functionality available int the ST-Link utility. It is however possible to call ST-Link Utility from inside the IDE, this can save a lot of time when performing various debugging related tasks.

Typical use cases when this is beneficial:

- When certain parts of the flash need to be erased before loading binary
- When you want to compare the binary file in target with the one just built with TrueSTUDIO.
- For setting option bytes such as read out protection.
- For faster loading into flash than offered by the ST-Link GDB-server
- …



An example..

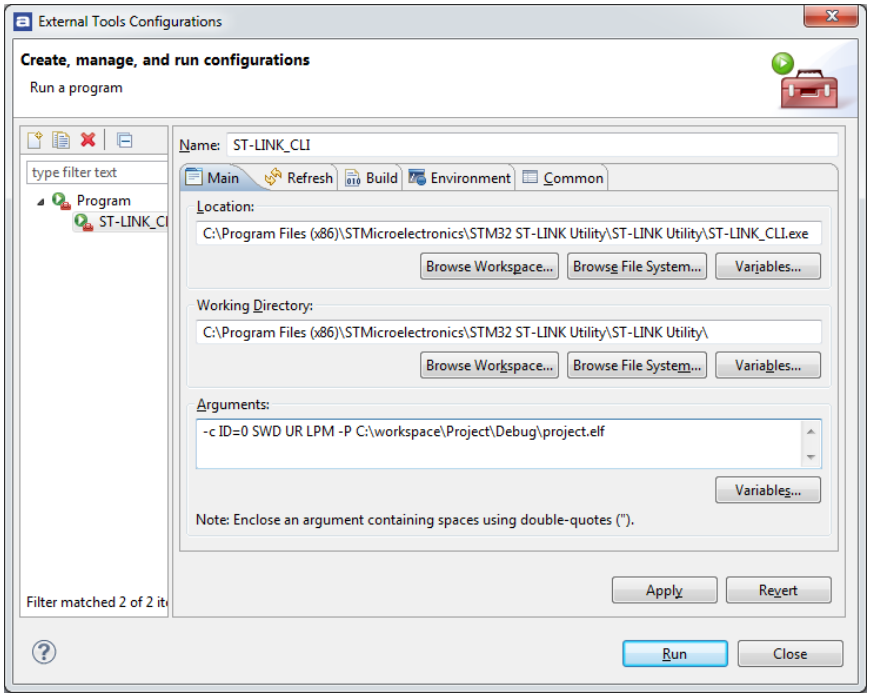
…on how to setup ST-Link utility to flash a binary into target before TrueSTUDIO launches a debug session…

Steps that needs to be performed:

- Setup ST-Link Utility with suitable input parameters as an external tool
- Convert your build output to Intel Hex
- Create / modify a debug configuration so that the flash operation is ONLY performed by ST-Link Utility
- Create a Launch Group to perform the ST-Link Utility operations before the TrueSTUDIO debugger starts

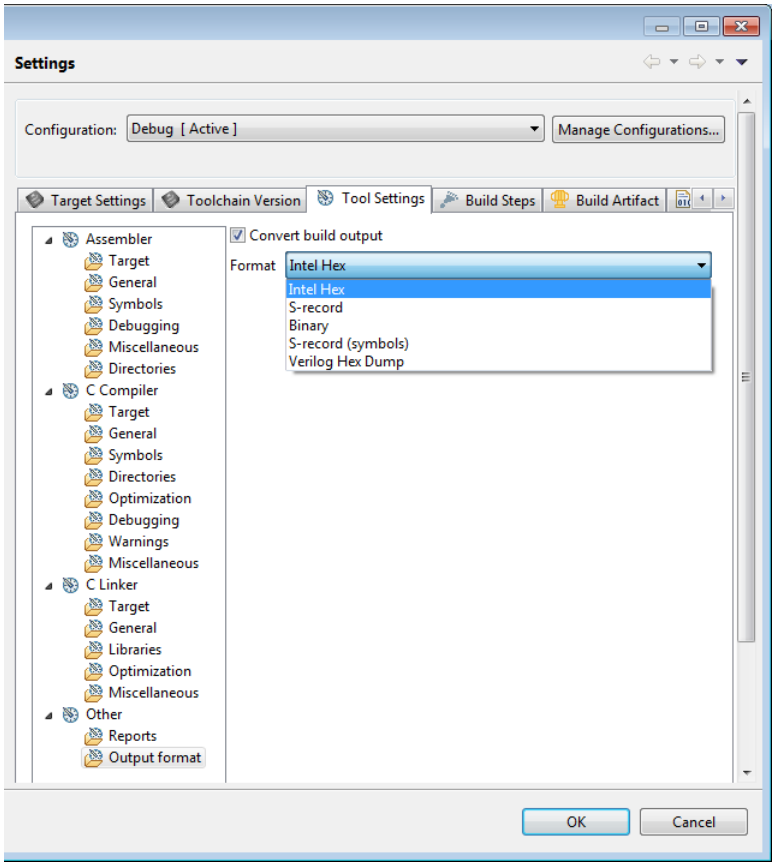
Setup ST-Link Utility as an external tool

- Run → External Tools… → External Tools Configurations…
  - Create a new "Launch configuration"
    - Name i.e. "ST-LINK_CLI"
    - Location i.e. *C:\Program Files (x86)\STMicroelectronics\STM32 ST-LINK Utility\ST-LINK Utility\ST-LINK_CLI.exe*
    - Arguments i.e. *-c ID=0 SWD UR LPM -P C:\workspace\Project\Debug\Project.hex*
    - Working directiory i.e. *C:\Program Files (x86)\STMicroelectronics\STM32 ST-LINK Utility\ST-LINK Utility*
    - *Apply.*



- Test that the external tool just setup is working by clicking *Run* or Run → External Tools… → ST-LINK_CLI

Convert your build output to Intel Hex

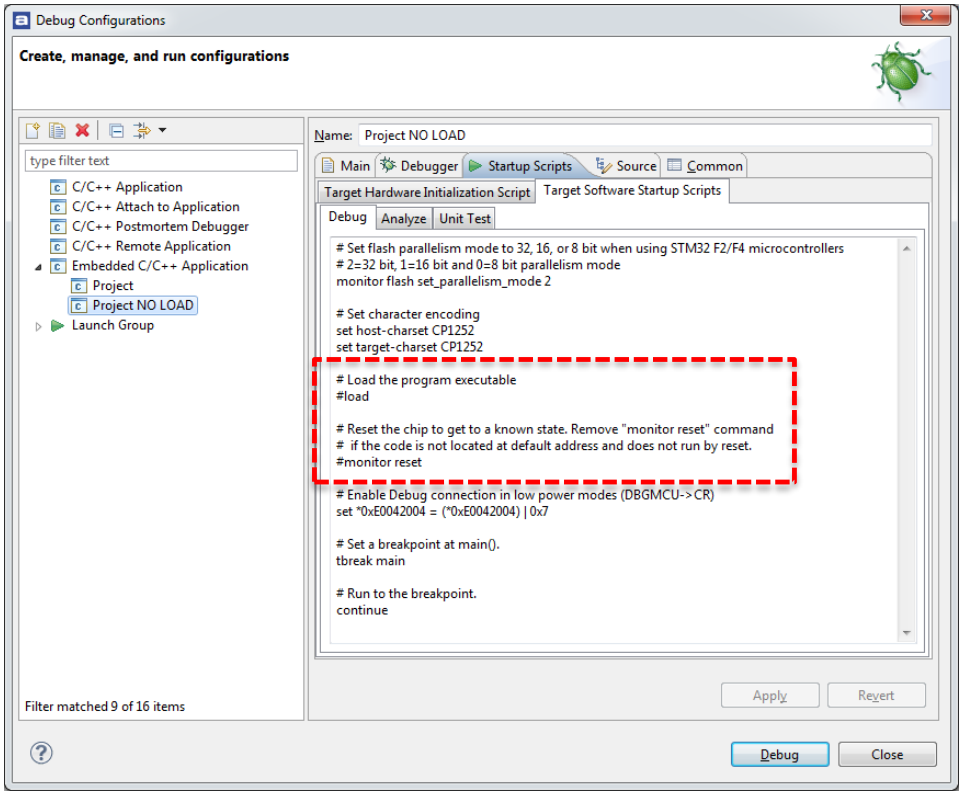


Project → Build settings… → C/C++ Settings → Tool Settings → other → Output format.

- Be cautious about which Configuration that is selected!
- Check the *Convert build output* checkbox
- Select *Intel Hex*
- *OK*
- Build your project!
- The output name will be *%PROJECT%.elf.hex*. Make sure that this binary is selected when creating the debug configuration. This will not work with an *.elf*-file

Modify the Debug configuration

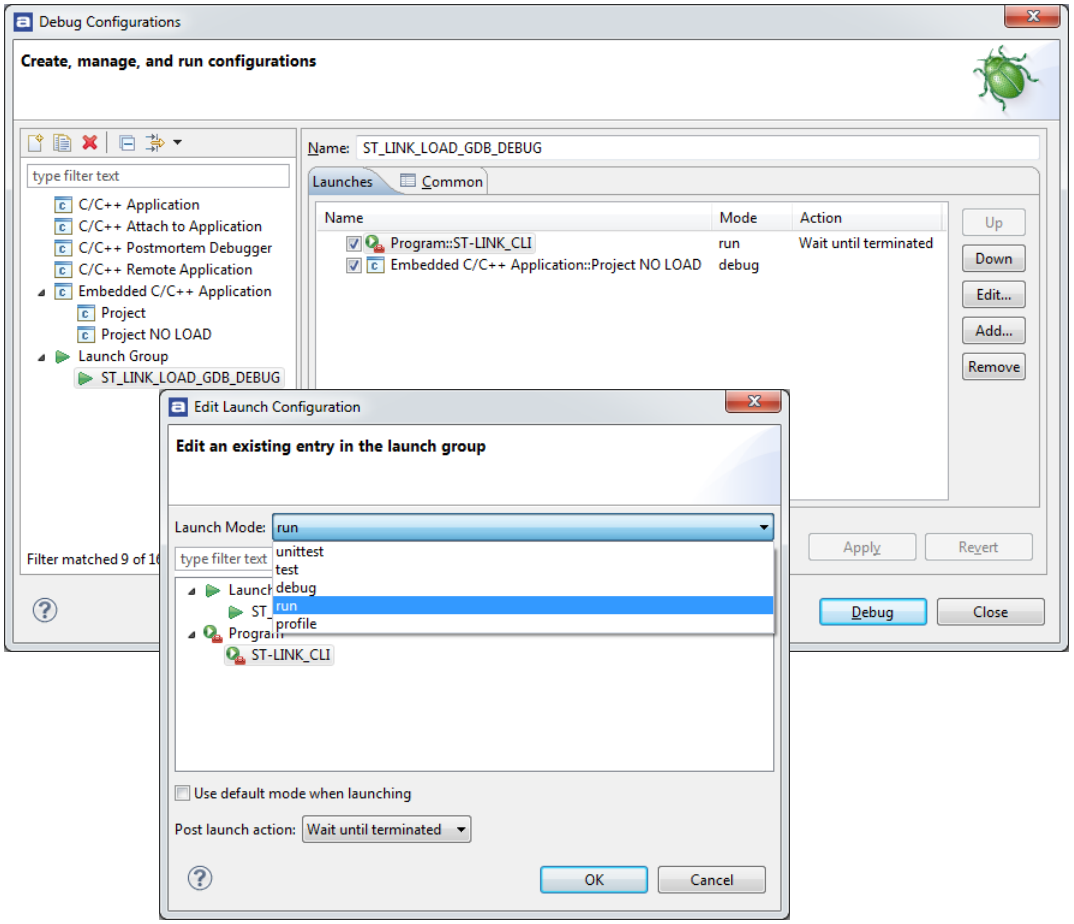
It is recommended that you make a copy of your current debug configuration as we will need to modify the debug script slightly.



- Run → Debug Configurations… → Righ-click on your debug configuration → "duplicate"
- Change the name of this configuration to "… NO LOAD", this is since GDB will not be used to load the hex.
- Open the "Startup Scripts" tab, comment out the "load" command load → #load. You may also want to comment out the "monitor reset" command.
- Click Apply.

Create a Launch Group 1/3

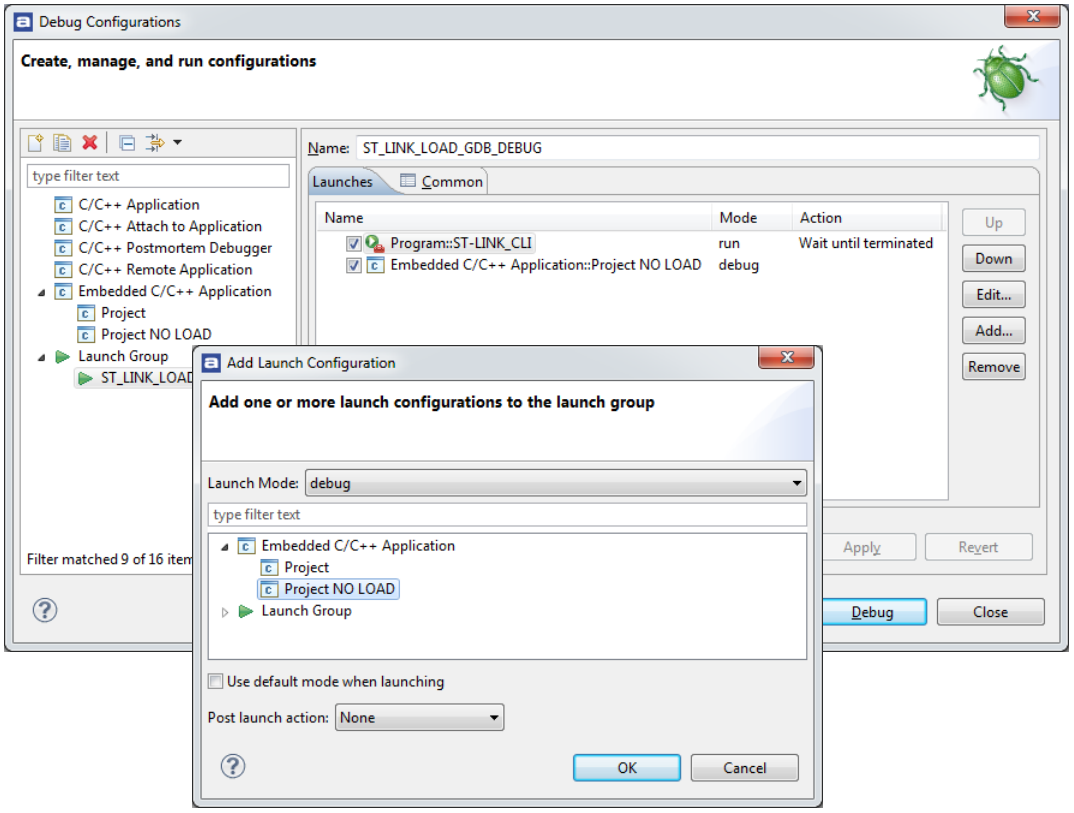
The Launch Group will allow you to launch several applications (configurations) by just clicking one button



- Double-click on the "*Launch Group*" node to create a Launch group and give it a name.
- Click *Add..*
  - Select *Launch Mode: run*
  - Expand *Programs* and select your *external tool configuration*, i.e. *ST-LINK_CLI.*
  - Set *Post launch action = Wait until terminated.*
  - *Click OK*

Create a Launch Group 2/3

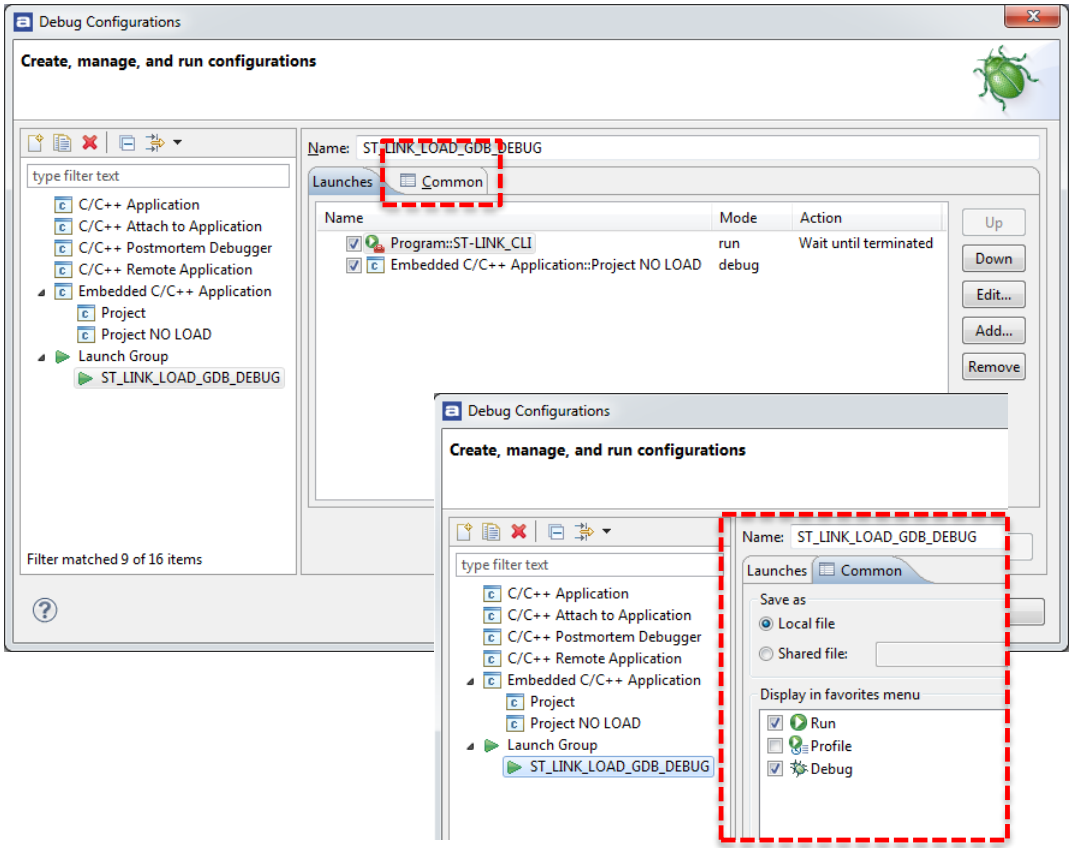
The Launch Group will allow you to launch several applications (configurations) by just clicking one button



- Click *Add..*
  - Select *Launch Mode: debug*
  - Expand *Embedded C/C++ Applications* and select your debug configuration, i.e. *Project NO LOAD.*
  - Set *Post launch action = None.*
  - Click *OK*

Create a Launch Group 3/3

The Launch Group will allow you to launch several applications (configurations) by just clicking one button



- Click *Common* tab
  - *Display in favorites menu = Run*
  - *Display in favorites menu = Debug*
- Click *Apply*

This will make the launch group available in TrueSTUDIO from the *Run → Run* menu and later the *Run → Debug History → …*

Finished!

ST-Link Utility is now flashing the binary into the target memory and the debugger is started as soon as the ST-Link Utility has finished.

You can create several configurations performing different tasks!