CSCE 4114/5114 Embedded Systems: Board Flashing Instructions

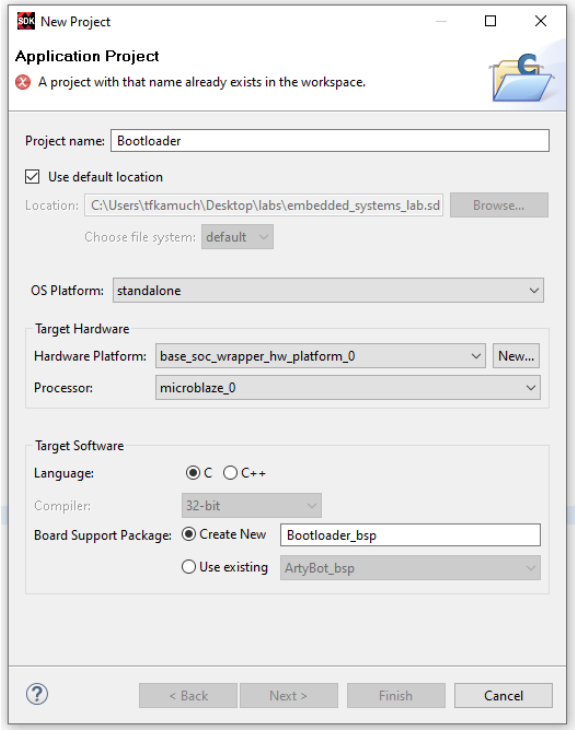
Tendayi Kamucheka

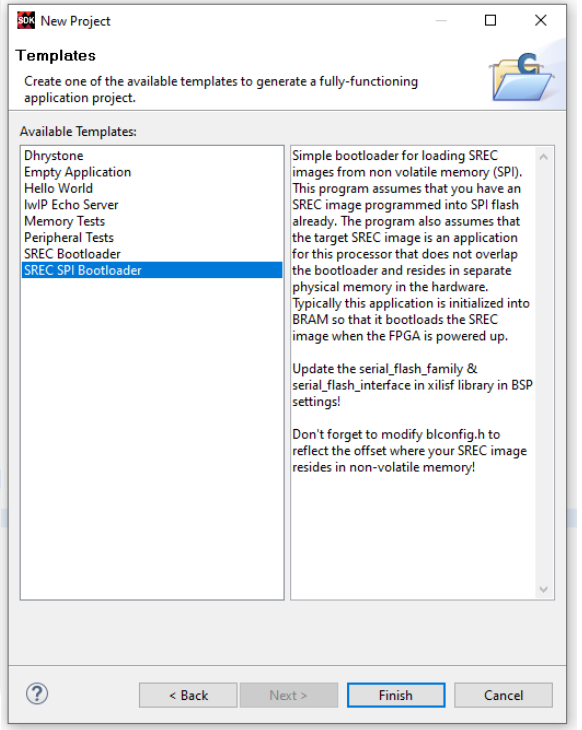
Introduction

- To get your application flashed on your board you will need two things:
  - A bootloader application to launch your application once it is stored on the flash memory
  - Your project application

Bootloader: Creating the Application

- Start off by creating a new "**Application Project**" in the SDK
- Name the project "*Bootloader*" or something along those lines
  - Make the OS Platform is "**standalone**"
  - For Board Support Package, you'll want to create a new package for the Bootloader.
- On the next page:
  - Select the "**SREC SPI Bootloader**" template
  - Then click on "**Finish**" to create the project

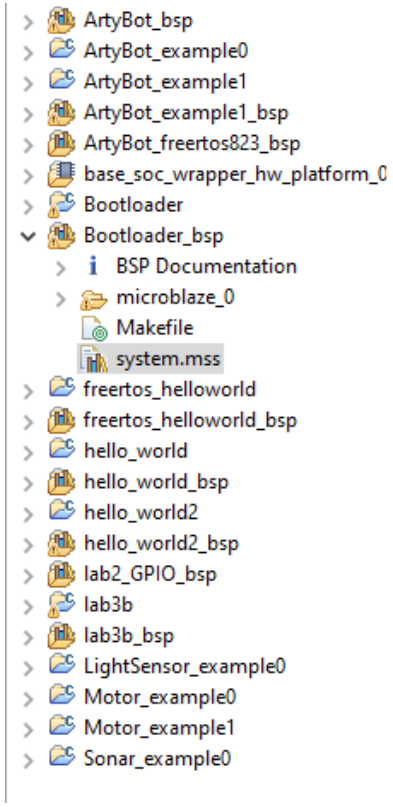


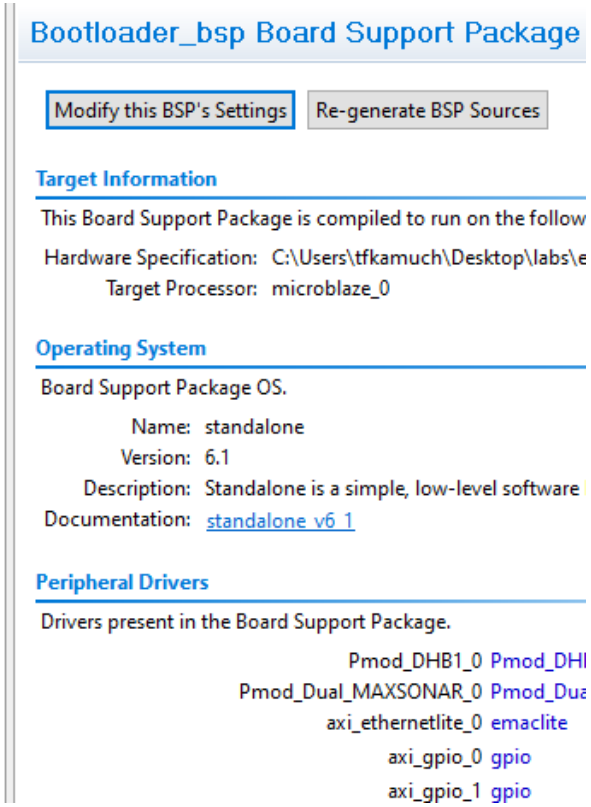


The examples repository has a bootloader ready to go!

Bootloader: Board Support Package

- We need to make a few changes to the board support package before the Bootloader is ready:
  - Namely, we want to add support for flash memory
  - And change the flash memory family to match what's on the board
- Under "**Bootloader_bsp**":
  - Open the "*system.mss*" file
  - Then select "Modify this BSP's Settings"

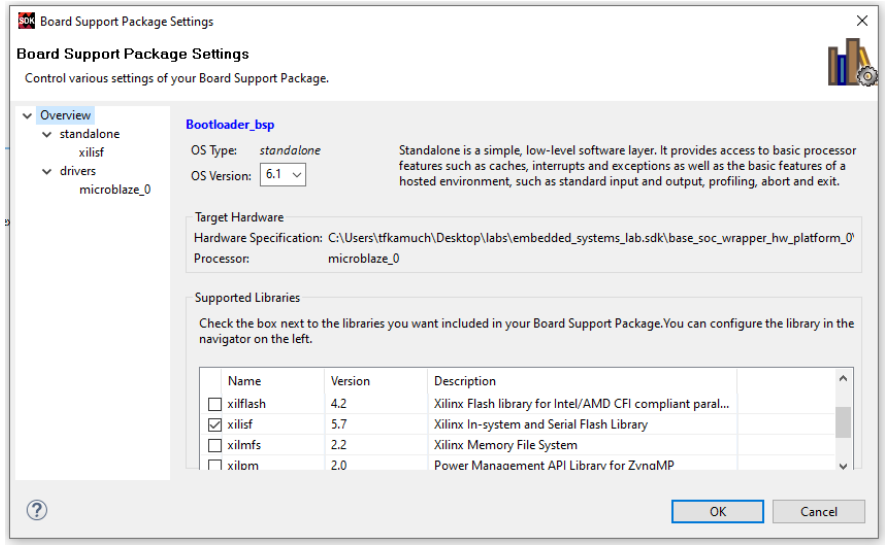


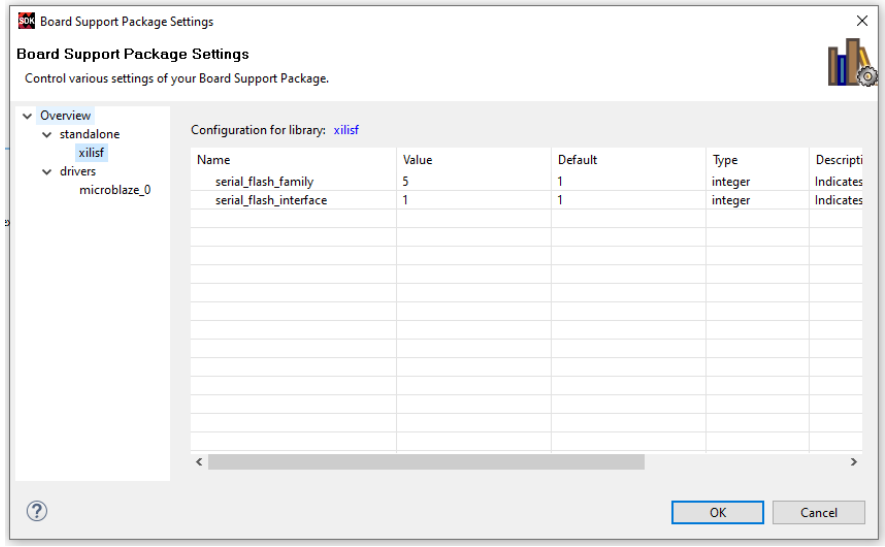


The examples repository has a bootloader ready to go!

Bootloader: Board Support Package

- In the Board Support Package Settings window:
  - Enable the "**xilisf**" option under supported libraries
  - A new "**xilisf**" option should appear on the left side under "*Overview > standalone > xilisf*"
  - Select the "**xilisf**" option and modify "serial flash family" value to **5**.
  - **5**, matches the Spanion flash chip on the Arty boards
- We're done with the Bootloader's BSP!

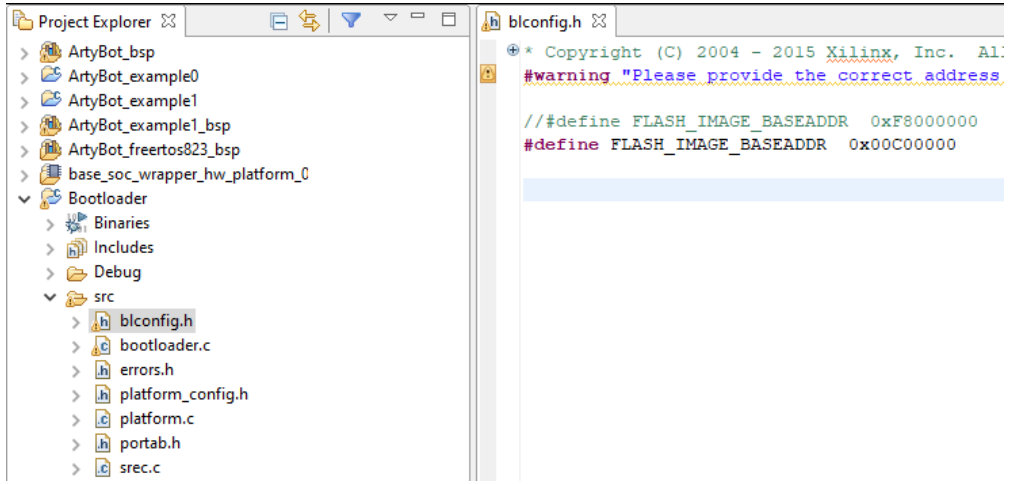


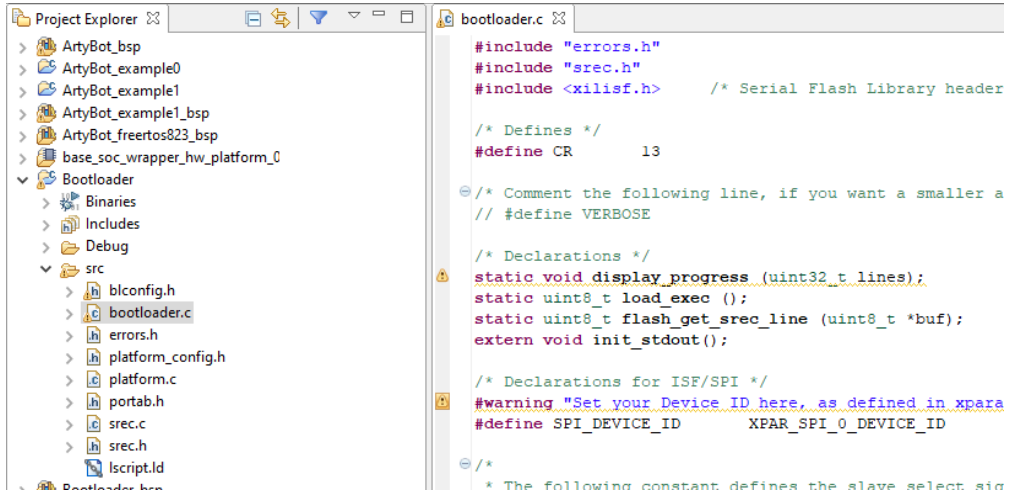


The examples repository has a bootloader ready to go!

Bootloader: Memory Configuration

- In the "Bootloader" project, open the "*blconfig.h*" file.
- Next, modify the "**FLASH_IMAGE_BASEADDR**" value to "**0x00C00000**"
- This will be the starting address of your project in the flash memory
- Optionally, if you want to see debug info in Putty:
  - Uncomment the "#define VERBOSE" line in "*bootloader.c*"
- The Bootloader is ready!

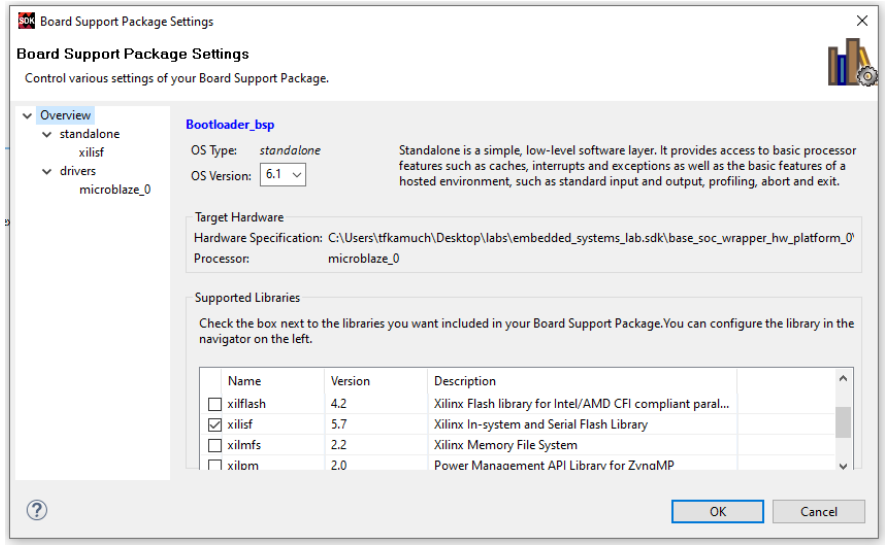


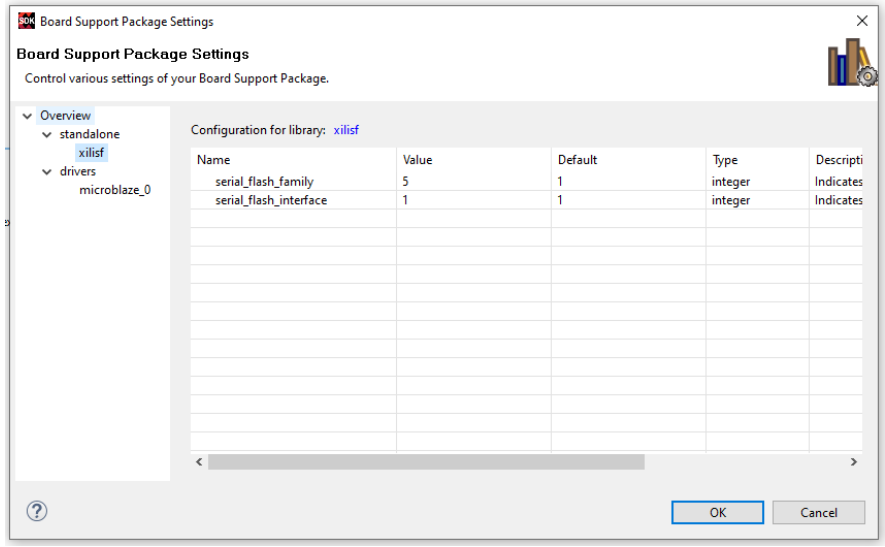


The examples repository has a bootloader ready to go!

Creating Your Project: BSP

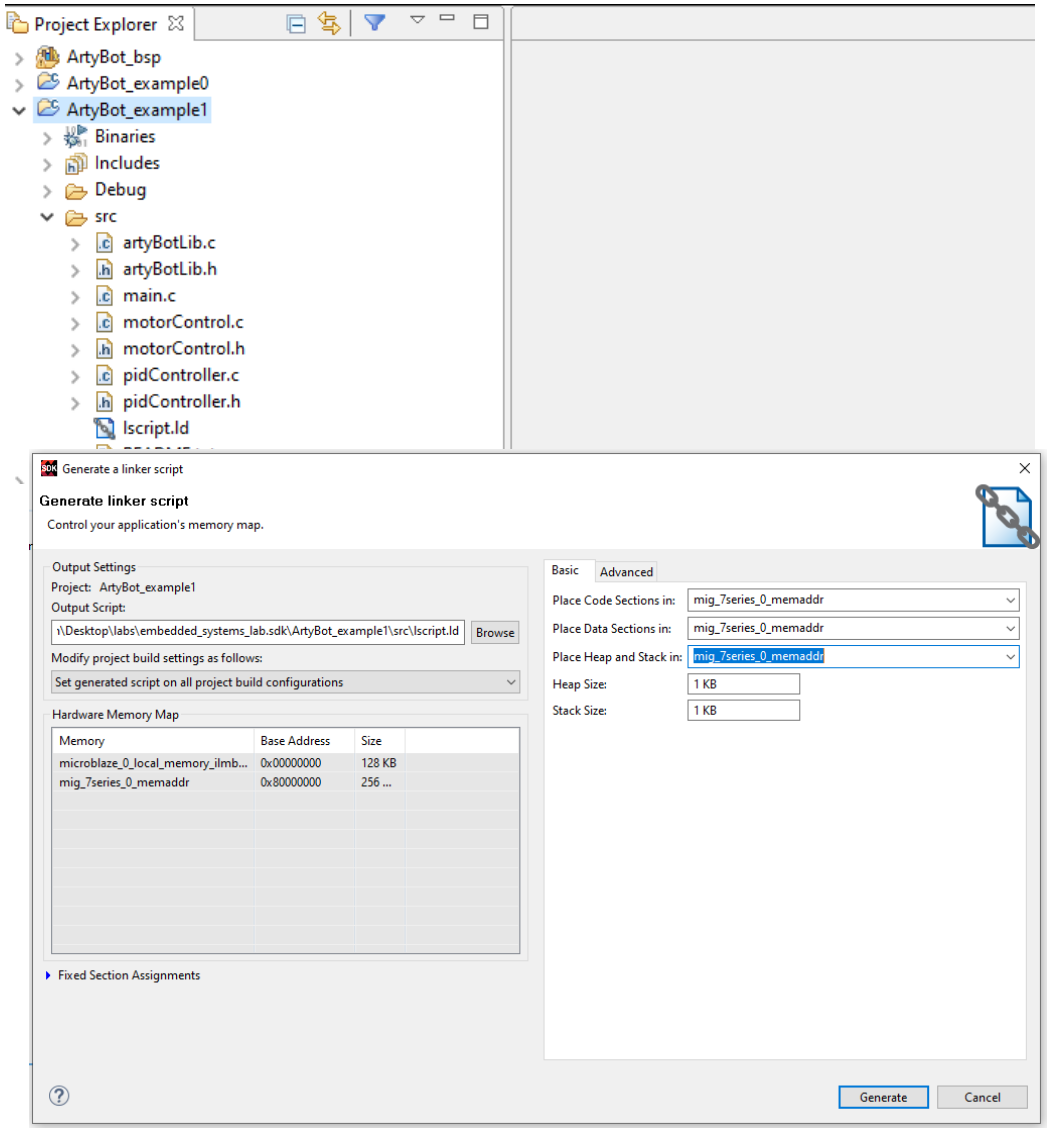
- Start by creating your "**New Application Project**" for your design:
  - You can create a separate board support package for your application
- Next, make the same changes to your application's board support package to support the flash chip





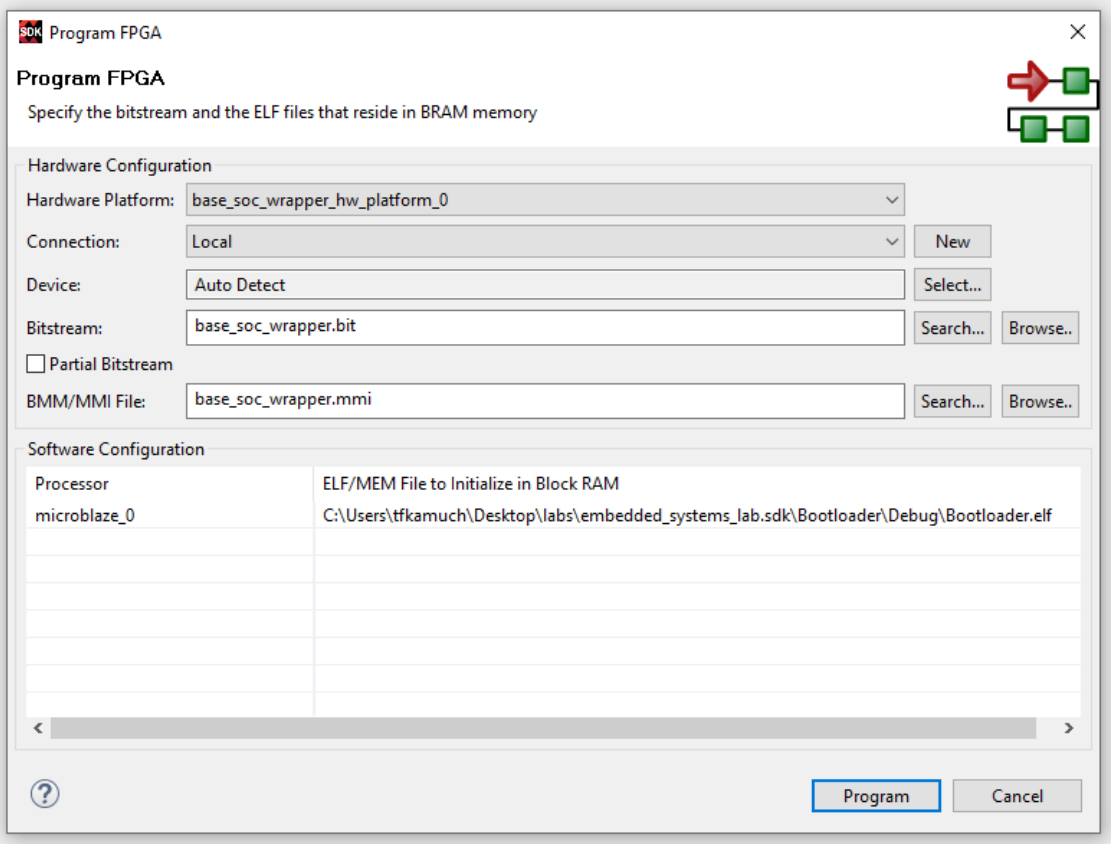
Creating Your Project: Linker Script

- We want to make sure our compiled application are run from the DDR3 ram, so we need to change the linker script
- Start by right-clicking your project's folder and selecting "**Generate Linker Script**"
- On the right side of the generate window, change all the memory locations to "*mig_7series_0_memaddr*"
- Now, move on to developing your amazing application



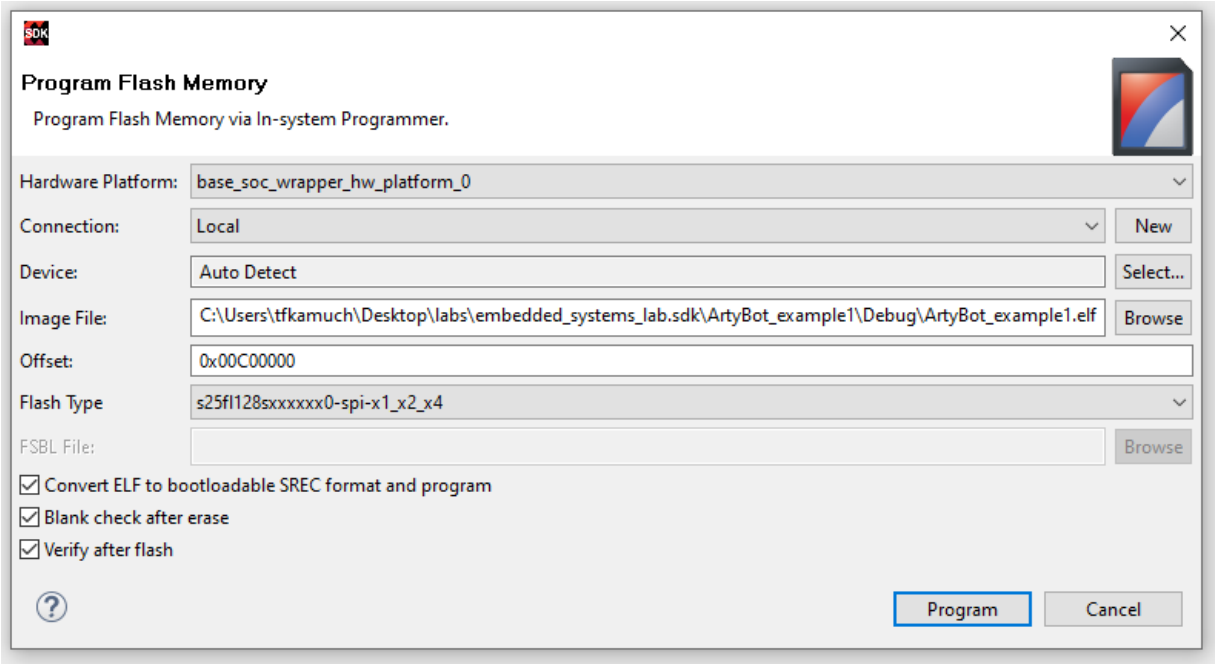
SPI Flashing: Generating Bitstream

- First, we generate a bitstream that we'll later use to flash the SPI memory.
- Open the "**Program FPGA**" utility
- Then change the "**ELF/MEM File to Initialize Block RAM**" option to "*Bootloader.elf*"
- Refer to example image for path to elf file
- Now, click "**Program**"



SPI Flashing: Your project

- Now let's get your project on the board
- Open the "**Program Flash Memory**" utility
- Set the "**Image File**" to your project's compiled elf file
- Set the "**Offset**" to "**0x00C00000**"
- For flash type, refer to example image
- Check all the checkboxes in the bottom section
- Now, click "**Program**" to begin flashing the board



SPI Flashing: Bootloader

- Almost there! Now we need to flash Bootloader
- Open the "**Program Flash Memory**" utility again
- This time, change "**Image File**" to "*base_soc_wrapper_hw_platform_0\download.bit*"
- Set the "**Offset**" to "**0x0**"
- Finally, click "**Program**"
- We're done! Your bot should be ready to drive solo.