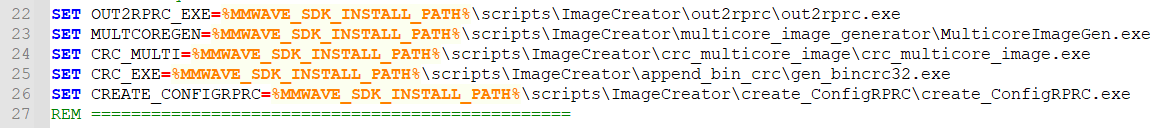
How to Build the TI SDK Binary (3.5.0.4)

1. Set up the build environment in Windows. (See SDK manual pg. 37)
   1. Create command prompt at <mmwave\_sdk\_<ver>installpath>\packages\scripts\windows\ folder.
   2. You should see a file “setenv.bat” in this directory.
   3. If needed, change the device type inside “setenv.bat”. We will refer to this variable as <device\_type>.
      1. Typical values for <device\_type> are “awr14xx”, “iwr14xx”, “awr16xx”, etc.
   4. From command prompt, run “setenv.bat”
   5. Extra steps for a secure development in SDK manual pg. 37.
2. Building the demo
   1. Make sure the environment is setup from the steps before.
   2. Create command prompt at <mmwave\_sdk\_<ver>installpath>\ti\demo\<device\_name2>\mmw folder.
      1. Typical values for <device\_name2> are “xwr68xx”, “xwr16xx”, “xwr18xx”.
   3. Call “gmake clean” to clean the current project.
      1. On first attempt, must run gmake clean. Then, incremental builds only need to call “gmake all”.
   4. Call “gmake all” to build the project.
3. Advanced build
   1. In case modification has been made to the driver, algorithms, or control component libraries, they can be built with the following instructions.
   2. First, make sure the environment is setup from the steps before.
   3. Navigate to the correct path and create a command prompt.
      1. adcbuf library: %MMWAVE\_SDK\_INSTALL\_PATH%\ti\drivers\adcbuf
      2. mmwavelink library: %MMWAVE\_SDK\_INSTALL\_PATH%\ti\control\mmwavelink
      3. aoaproc dpu library: %MMWAVE\_SDK\_INSTALL\_PATH%\ti\datapath\dpc\dpu\aoaproc
   4. Call “gmake clean” to clean the current project within the correct path.
   5. Call “gmake all” to build the current project.

Attempt to enable hardware trigger for xwr68xx demo

1. Goal: enable LVDS streaming, CLI interface, and hardware trigger for xwr68xx sensor
2. Dependencies:
   1. Under directory %MMWAVE\_SDK\_INSTALL\_PATH%\ti\demo\xwr68xx\mmw
      1. Inside makefile
      2. include $(MMWAVE\_SDK\_INSTALL\_PATH)/ti/common/mmwave\_sdk.mak
      3. Inside mmwave\_sdk.mak
      4. GENERATE\_METAIMAGE = @$(MMWAVE\_SDK\_INSTALL\_PATH)/scripts/windows/generateMetaImage.bat
      5. Inside generateMetaImage.bat
      6. Calls several .exe files
      7. Stuck here.
   2. Under directory %MMWAVE\_SDK\_INSTALL\_PATH%\ti\demo\xwr68xx\mmw\include
      1. Inside mmwave\_config.h
      2. #include <ti/control/mmwave/mmwave.h>
      3. Inside mmwave.h
      4. #include <ti/control/mmwavelink/mmwavelink.h>
      5. Inside mmwavelink.h
      6. #include <ti/control/mmwavelink/include/rl\_sensor.h>
         1. Helpful: w.r.t the SYNC\_IN pulse, the actual transmission has ~160 ns delay and 5 ns uncertainty in singlechip and only 300 ps uncertainty (due to tight inter-chip synchronization needed) in MULTICHIP sensor applications as defined in rlChanCfg\_t. For more details, please refer to device datasheet. (Line 1006)
         2. Variable triggerSelect, type rlUInt16\_t, under struct rlFrameCfg and also under struct rlAdvFrameSeqCfg
      7. Inside ti/control/mmwavelink/src/rl\_sensor.c
         1. SYNC\_IN pulse needs to be between 25 ns and 1 us. (Line 976)
         2. Recommended that time between pulses <300 us. (Line 989)
         3. In non-periodic HW triggered mode, if frame stop command is issued when HW pulses are paused (i.e. after completing previous frame), then HW pulse is required to trigger next frame/bursts and frame will be stopped at the end of this triggered frame. In HW triggered mode, the forced frame stop is not supported. The frame end command is honored only if frames are actively running. (Line 1483)

Inside CLI Platform

* Look into mmw\_spi\_comm.c
* Possibly: rangeprochwa.c
* Mmwave\_json\_parser.c