**Building PicTest project**

Start with building the Bitstream

1. Clone or download: <https://github.com/zahike/PicTest.git> from github to <PicTest-Dir> directory.
2. If you cloned, sometimes you'll get a <trunk> directory.

you should have 6 directories:

* constraints
* fpga\_sw
* rtl
* script
* simulation
* work

1. Open Vivado 2018.2 . In tcl command line go to:

cd <PicTest-Dir>/work

1. Run in tcl command line:

source ./build.tcl

1. After the build finished run:

source ./PicTesst\_BD.tcl

1. After the block design was created run:

source ../script/FreqChanges.tcl

1. Save and generate the block design.
2. After the block design was generated, generate Bitstream.

When the Bitstream is ready, build the software.

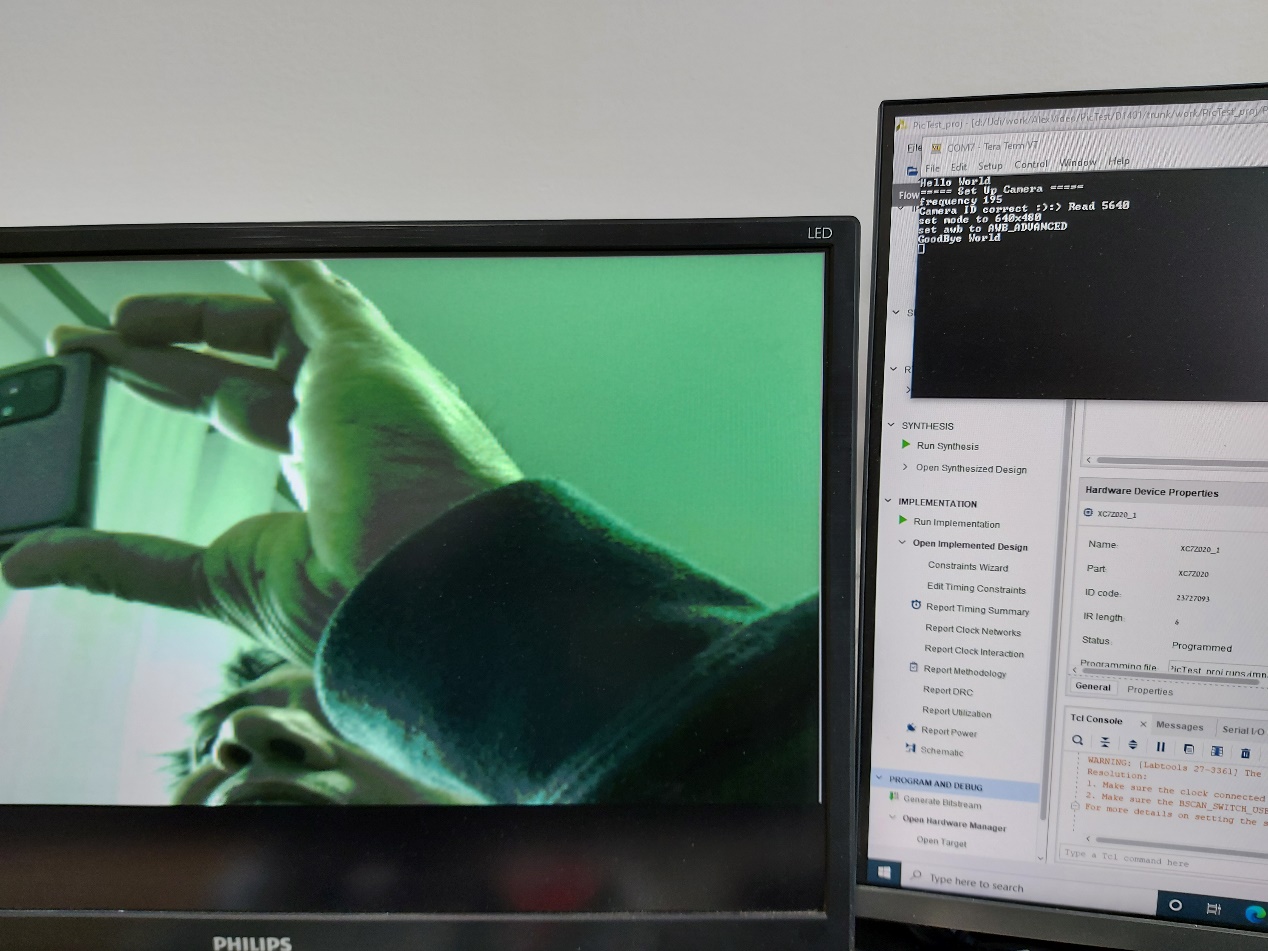
1. Export Hardware to: (include bitstream)

<PicTest-Dir>/fpga\_sw

1. Lunch SDK to the same location.
2. Create a new application project with 'Hello World' template.
3. Copy all the \*.c & \*.h files into the new project /src/ directory. (Replace the helloworld.c template file with /fpga\_sw/helloworld.c file)
4. Clean & build the new project.

With the Bitstream & Software ready, start the system.

1. Connect the ZyboZ7 board:
   1. Connect the Camera to the board.
   2. Connect the HDMI out port to a screen.
   3. Connect the usb port to the computer.
2. Turn on the board & screen.
3. Load the Bitstream (ether from the Vivado or the SDK)
4. Open UART console (like Tera Term or Putty) and connect it to the COM with speed of 115200.
5. Run the new Application project.



The Design has 4 control switches:

* 1. SW[3] : '1' – use RGB555;

'0' – use RGB444;

* 1. SW[2] : '1' – use Checkmate mode;

'0' – view original picture; (there is not enough BRAM for full RGB555 picture so the picture is 640X400 pixels).

* 1. SW[1] : '1' – use Auto Checkmate mode.

'0' – show individual frame selected by SW[0].

* 1. SW[0] : Select frame0 or frame1 – in individual Checkmate mode.