

# UPduino v3.0 Lattice Radiant and Reveal v2.2 Setup With Debug Cable

The following document is a quick set-up guide for the UPduino v3.0 FPGA programming board. The purpose of this document is to show how to set up the RGB blinky verilog file with the Lattice v2.2 toolchain plus running an example debug session using the Reveal v2.2 tools.

All files and example project files have been provided – it could be necessary to change some of the files to fit the directory structure the examples are downloaded/cloned into – this is OS and tool dependent to some extent. The demo programs here were all run under Win10-64.

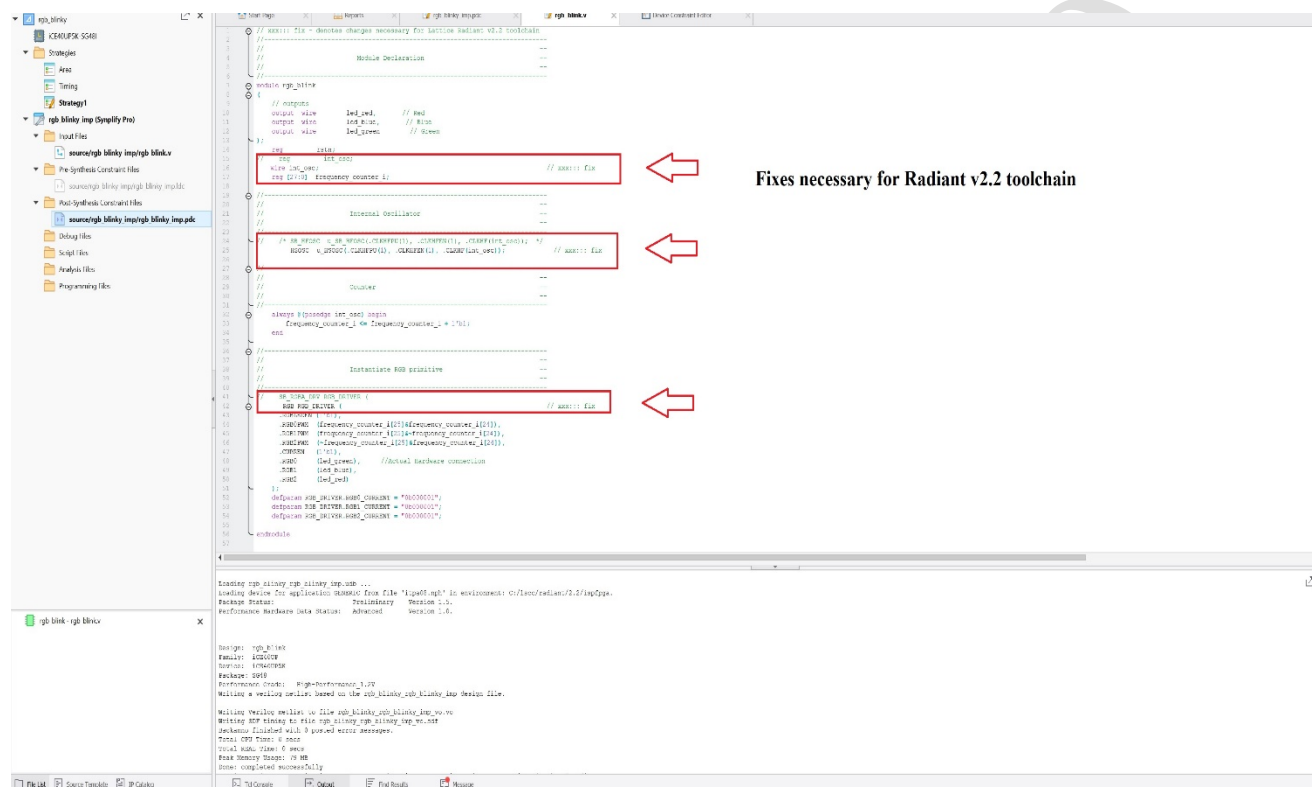
It is assumed the UPduino v3.0 git repository has been downloaded to a local directory as well as what is contained in this demo document and the corresponding directories. To get started under Radiant – it is necessary to set up a project and to make some small changes to the `rgb_blink.v` file – if you have downloaded/cloned the files – but if you are following along and want to make your own local changes – diagrams/pictures have been provided that will hopefully make it easy. If you haven't gone through the Lattice tutorial document – it will more than likely be helpful to take a look at that – plus it will be referenced in this document a few times - **Lattice Radiant Software Tutorial for iCE40 UltraPlus<sup>1</sup>** – the download link is here (if this can't be found – then 'google' the title):  
[https://www.latticesemi.com/media/LatticeSemi/Documents/InstallationGuides/Radiant20\\_Tutorial\\_ice40.ashx?document\\_id=52758](https://www.latticesemi.com/media/LatticeSemi/Documents/InstallationGuides/Radiant20_Tutorial_ice40.ashx?document_id=52758)

Please refer to the above tutorial for the finer details on creating a new project – it would be good to review even if you have cloned the project and are trying to run

it for the first time. If you have cloned the project and the assumption is that you have – the project you want to open is:

<your dir>/UPduino-v3.0/RTL/Lattice-Radiant-Project-v2p2/rgb\_blinky/rgb\_blinky.rdf

The first thing to note are a few changes are necessary to run the RGB demo code:



**Figure 1 – Necessary verilog updates to run demo program in Lattice Radiant.**

Note the changes outlined in Figure 1 are suggestions – but for the purposes of this demo please make sure that your verilog file has those updates or it will not build (under the Lattice Radiant tools).

Please make sure the project builds OK without error. There is an example constraints file included in the example project directory.

We want to be able to run the Reveal debugger – for purposes of this demo, the debug cable used is the **FTDI C232HM USB 2.0 HI-SPEED TO MPSSE CABLE<sup>2</sup>**.

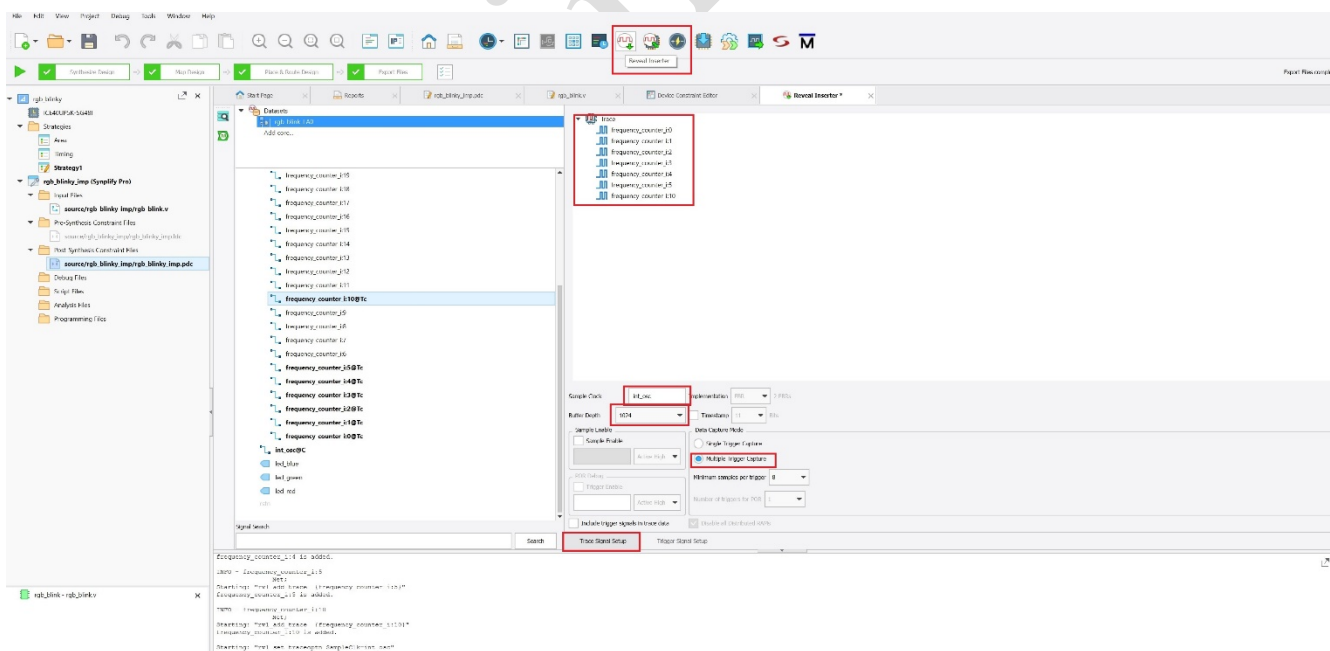
This cable should be able to be used as received from FTDI – the only real difference I have seen thus far is setting the drive current to 8 milli-amps for the I/O pins. The xml file for the FTDI cable is included in the git repository just in case. The FTDI cable can be viewed [here](#):

<https://www.ftdichip.com/Products/Cables/USBMPSSE.htm> and PDF:  
[https://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS\\_C232HM\\_MPSSE\\_CABLE.pdf](https://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_C232HM_MPSSE_CABLE.pdf)

Using this cable is cheaper than the **HW-USBN-2B<sup>3</sup>**:

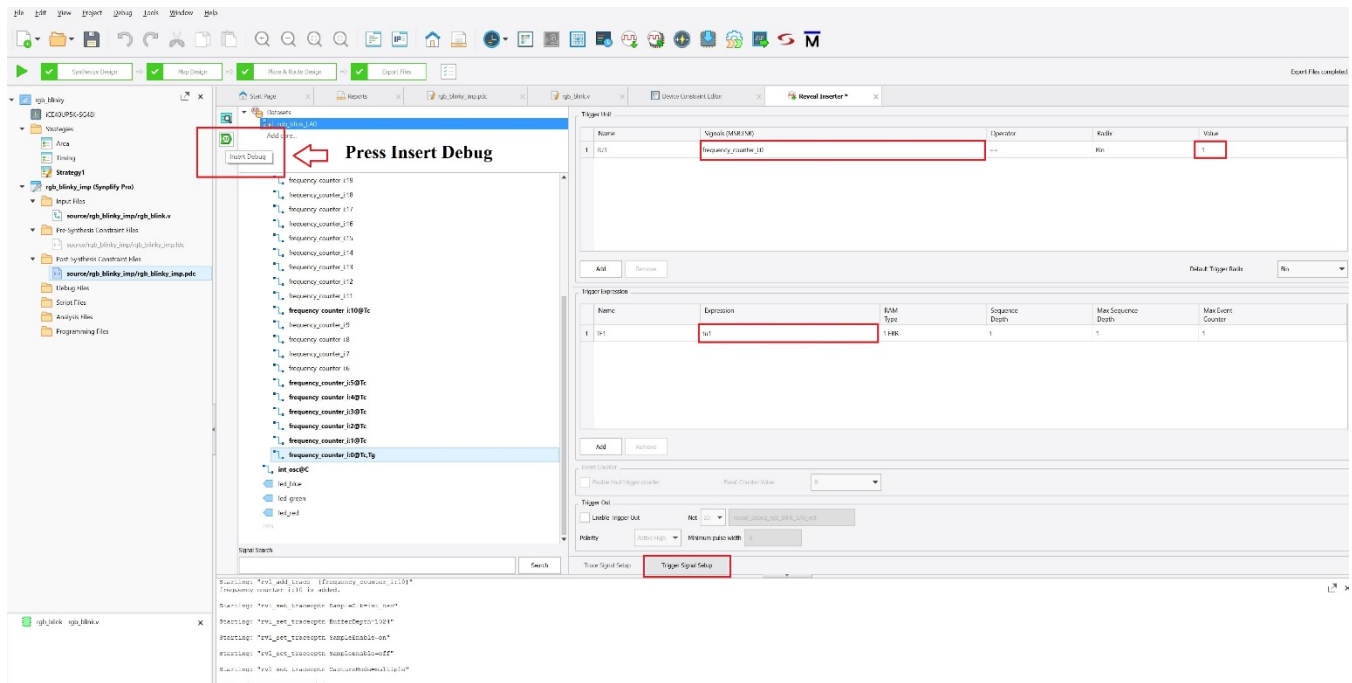
<https://www.latticestore.com/searchresults/tabid/463/searchid/1/searchvalue/hw-usbn-2b/default.aspx> - plus it is available – Mouser P/N: 895-C232HM-DDHSL-0 and current cost is \$35.00 (USD) (as an example of a cheaper option).

Please refer to pages 38 - ~ 44 of the tutorial for any questions regarding the following – the first step is to add debug logic to our design using the Reveal Inserter:



**Figure 2 – Using Reveal Inserter to add on-chip Debug Logic.**

Once the desired trace signals have been added – trigger signals need to be added:



**Figure 3 – Adding trigger signals to the Reveal Inserter set up for on-chip Debug Logic.**

To add to the project – press the Insert Debug Icon as shown in Fig. 3.

Once that is added – make sure the project can build. Once the project has built with no errors – then the JTAG connections need to be checked. The Lattice tools will set the JTAG pins even if they are not in the constraints file – but subsequent builds could reassign the pin locations – so it is a good idea to try and lock those down – otherwise you will be moving the JTAG pins every time you make a change and rebuild.

In the Device Constraints Editor – you can type in the pin number to override what the tools have assigned if it does not match the constraints file or what you have used for your JTAG pins. Make sure and connect GND first when connecting the FTDI cable to the UPduino board.

If you must update the JTAG pin assignments – you will have to rebuild to make sure the JTAG pins are assigned to the pin numbers used for the JTAG FTDI cable. Otherwise, a cable server error will appear (when making the connections – we’re not quite ready for this now – but as this is a potential issue – it’s better to bring it up sooner than later) and it’ll tell you to check items 9-1 (Failure Points) in the following document:

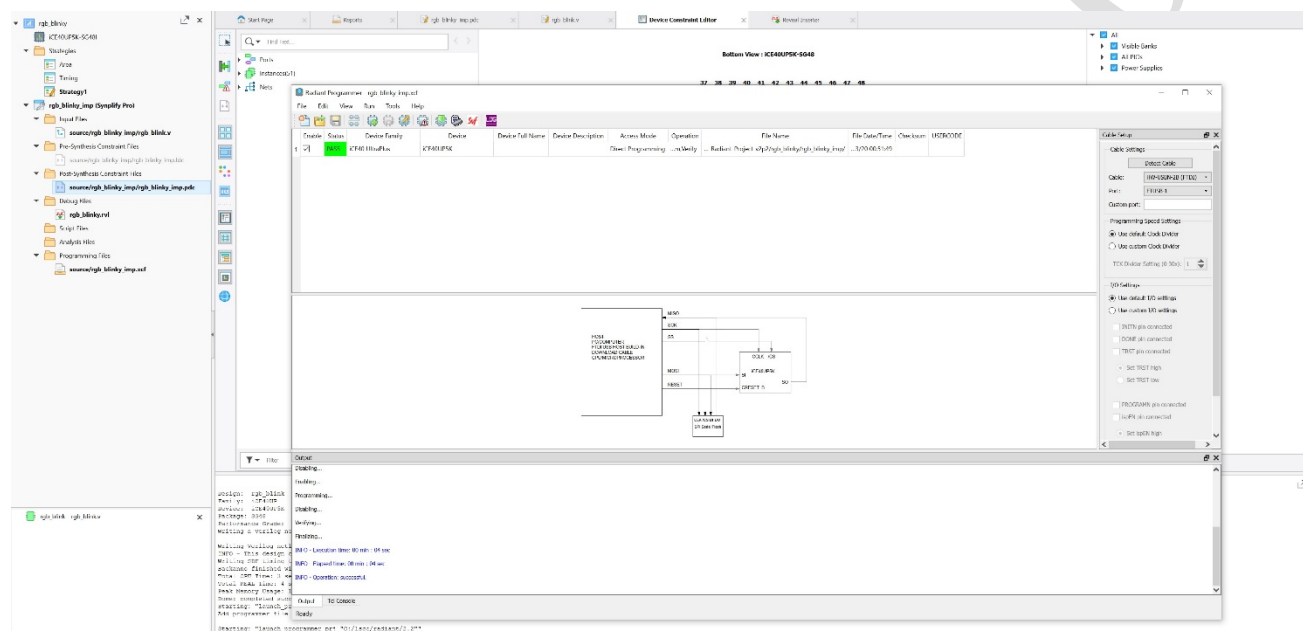
[https://www.latticesemi.com/view\\_document?document\\_id=52931](https://www.latticesemi.com/view_document?document_id=52931)

whathisface@protonmail.com

Once the project has been built and the pin assignments have been verified and you have connected your debugger –

Note you can refer to P. 43 of the tutorial document for a quick explanation on how to connect the debug cable to a target board – here we are connecting an FTDI cable to the UPduino v3.0 – but the procedure is essentially the same.

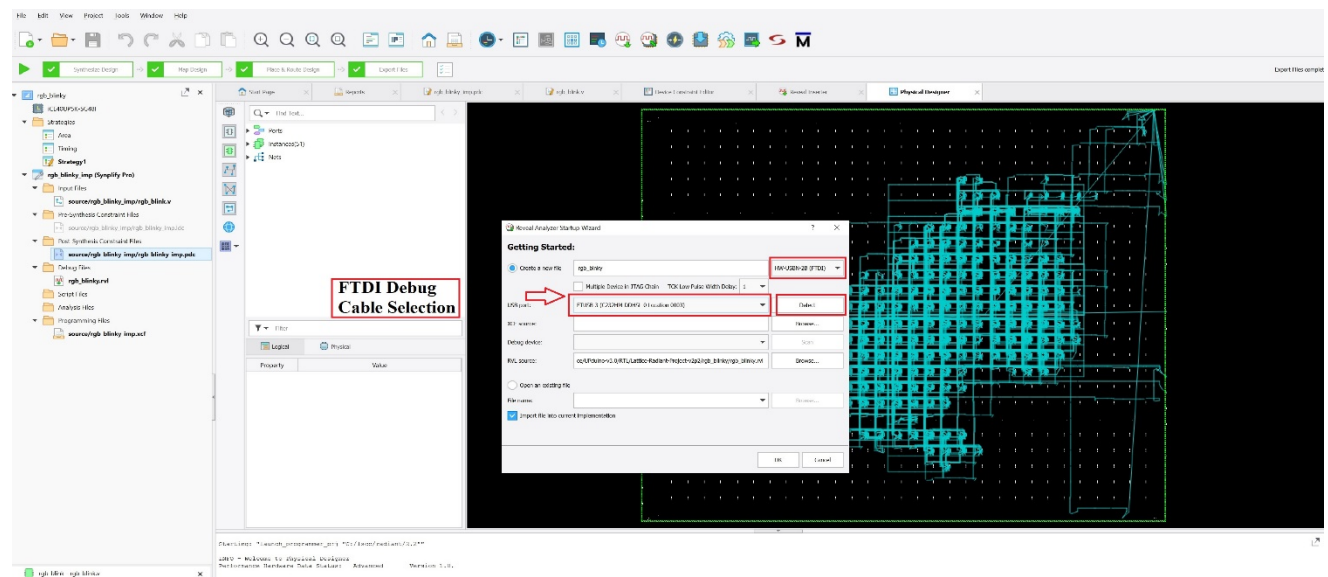
The build needs to be downloaded to the target:



**Figure 5 – Program the target board with the rgb\_blink.v build plus on-chip debug logic.**

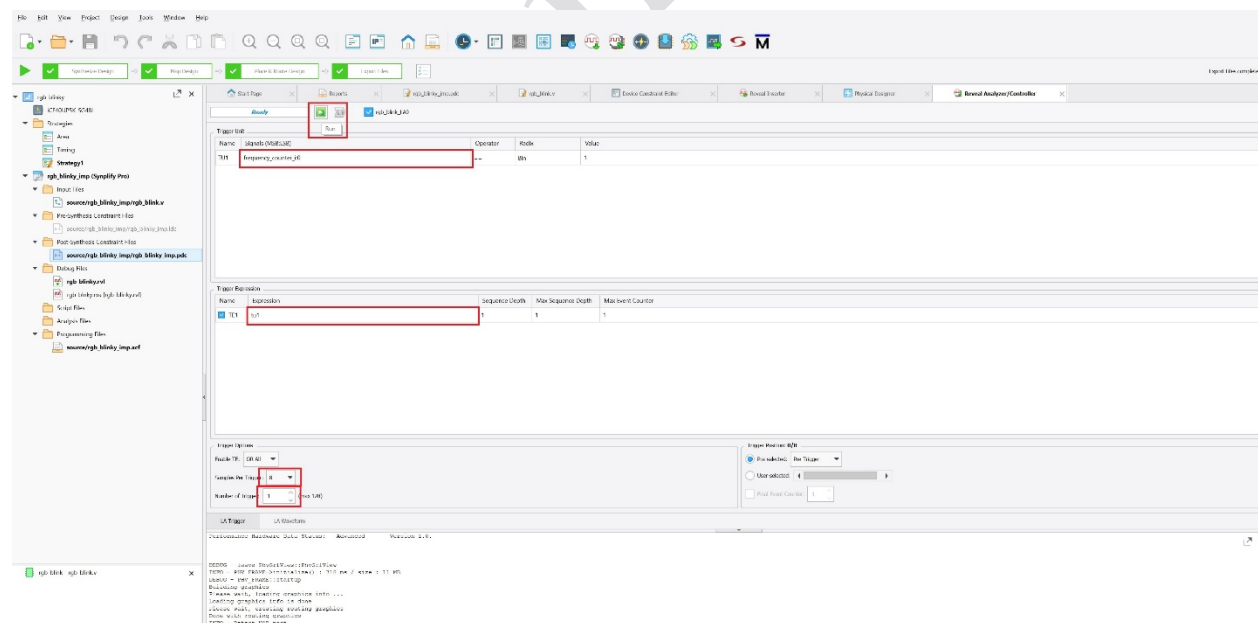
For the next steps you can refer to pages 44 - ~48 of the tutorial document.

Once the target has been programmed – open the Reveal analyzer to create the debug analyzer session:



### Figure 6 – Creating Reveal Analyzer Session

Refer to Figure 6 regarding creating the Reveal Analyzer session – make sure the correct FTDI cable is selected for the debug session.



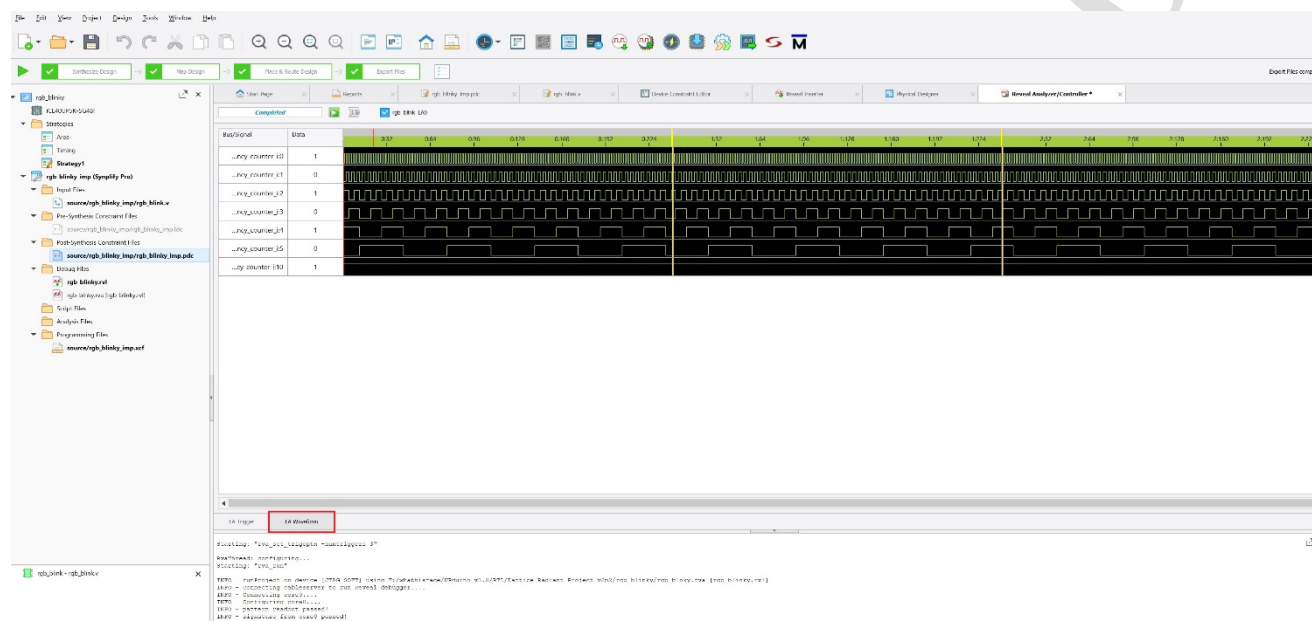
**Figure 7 – Reveal Analyzer Window**



Refer to Figure 7 regarding the Reveal Analyzer Window – the green box with the arrow can be pressed to run the analyzer session – note ‘Ready’ should be visible to the left of that window.

If an error pops up – please refer to the troubleshooter link mentioned previously – but errors are more than likely due to incorrect JTAG connections.

Once the Reveal Analyzer runs – you should see something like:



**Figure 8 – Reveal Analyzer Session Results**

Hopefully, this brief guide has been helpful in running a JTAG debug session on the UPduino v3.0 FPGA board and will serve as a handy reference.



## References:

### **(1) Lattice Radiant Software Tutorial for iCE40 UltraPlus**

[https://www.latticesemi.com/media/LatticeSemi/Documents/InstallationGuides/Radiant20\\_Tutorial\\_ice40.ashx?document\\_id=52758](https://www.latticesemi.com/media/LatticeSemi/Documents/InstallationGuides/Radiant20_Tutorial_ice40.ashx?document_id=52758)

### **(2) FTDI C232HM USB 2.0 HI-SPEED TO MPSSE CABLE**

<https://www.ftdichip.com/Products/Cables/USBMPSSE.htm> and PDF:  
[https://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS\\_C232HM\\_MPSSE\\_CABLE.pdf](https://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_C232HM_MPSSE_CABLE.pdf)

### **(3) LATTICE HW-USBN-2B Programmer**

<https://www.latticestore.com/searchresults/tabid/463/searchid/1/searchvalue/hw-usbn-2b/default.aspx>

### **(4) Reveal Troubleshooting for Lattice Radiant Software**

[https://www.latticesemi.com/view\\_document?document\\_id=52931](https://www.latticesemi.com/view_document?document_id=52931)