

Revision	Date	Author	Comments
1A	2020-09-11	Tim S. timothystotts08@gmail.com	First publishable draft of the HYGRO Sensor Readings Tester

<https://github.com/timothystotts/fpga-iic-hygro-tester-1>

Copyright 2020 Timothy Stotts
MIT License

HYGRO Sensor Readings-Tester Experiment

HYGRO Sensor Readings-Tester Experiment: Folder Structure

HYGRO Sensor Readings equivalent function of polling a HDC1080 sensor via IIC bus; displaying reading in multiple formats on a 16x2 character LCD; and displaying sensor reading display selection on a two-digit seven-segment display.

Project Folder	Project Description
HYGRO-Tester-Design-Verilog (Vivado 2019.1)	A utility designed for polling a temperature and relative humidity sensor once per second and displaying the readings in Celsius, Fahrenheit, and RH Percent. Two clock domains exist; 20. MHz and 7.37 MHz, together controlled by a MMCM; and each slower domain is controlled by a clock enable divider RTL dividing a MMCM clock by clock by clock enable pulse, or a clock divider, generating slower clocks that can be in a process' sensitivity list or drive an output clock port.
HYGRO-Tester-Design-AXI (Vivado 2019.1 and SDK 2019.1)	A utility designed for polling a temperature and relative humidity sensor once per second and displaying the readings in Celsius, Fahrenheit, and RH Percent. The design is completely in Microblaze AXI subsystem with standard Xilinx IP Integrator components, vendor Xilinx IP Integrator components, and Standalone C language program executing on the Microblaze soft processor.

For the HYGRO-Tester-Design-AXI project, the BSP of a bootloader (if added) require that the flash family parameter be set in the xilisf configuration on the BSP configure page. The default value is 1, but must be changed to 5 as the Arty-A7-100T board has a Spansion 128Mbit serial flash for booting the FPGA.

To successfully open the project, it is necessary to add the directory arty-a7-100 from the directory board_files/ to the installation directory of Vivado 2019.1 and also to the installation directory of SDK 2019.1. For example:

```
$ which vivado
/opt/Xilinx/Vivado/2019.1/bin/vivado
$ cd ./board_files
$ sudo cp -R ./arty-a7-100 /opt/Xilinx/Vivado/2019.1/data/boards/board_files/
$ sudo cp -R ./arty-a7-100 /opt/Xilinx/SDK/2019.1/data/boards/board_files/
```

Note that the Digilent Guide at <https://reference.digilentinc.com/vivado/installing-vivado/2018.2> indicates that an initialization script can be executed in the user's profile to set up a path to additional board files. It is the experience of this author that the TCL initialization script provides an intermittent or non-functional

detection of the board files in the user's home folder. By copying to the install directory of the tool, the board files are always found. Otherwise, the following TCL command is supposed to instruct Vivado to locate the board files:

```
set_param board.repoPaths [list "<extracted path>/Vivado/board_files"]
```

HYGRO Sensor Readings-Tester Experiment: Methods of Operation

The purpose of the design is to boot a Digilent Inc. Arty-A7-100T (Artix-7) development board with Pmod CLS, Pmod HYGRO, and Pmod SSD peripheral boards, which are a 16x2 Character dot-matrix LCD display, a IIC bus temperature and relative humidity sensor, and a two-digit seven-segment display, respectively. The PMOD CLS connects to the FPGA with its own dedicated SPI bus via a higher-speed plug and jack. The Pmod CLS connects to board Pmod port JB. The Pmod HYGRO connects to board Pmod port JC. The use of an extension cable makes the Pmod CLS able to connect to only one 2x6 Pmod port and the Pmod SSD able to connect to only one 2x6 Pmod port. See Figure 1: Arty-A7-100T Assembled with Pmod CLS, Pmod HYGRO, Pmod SSD.

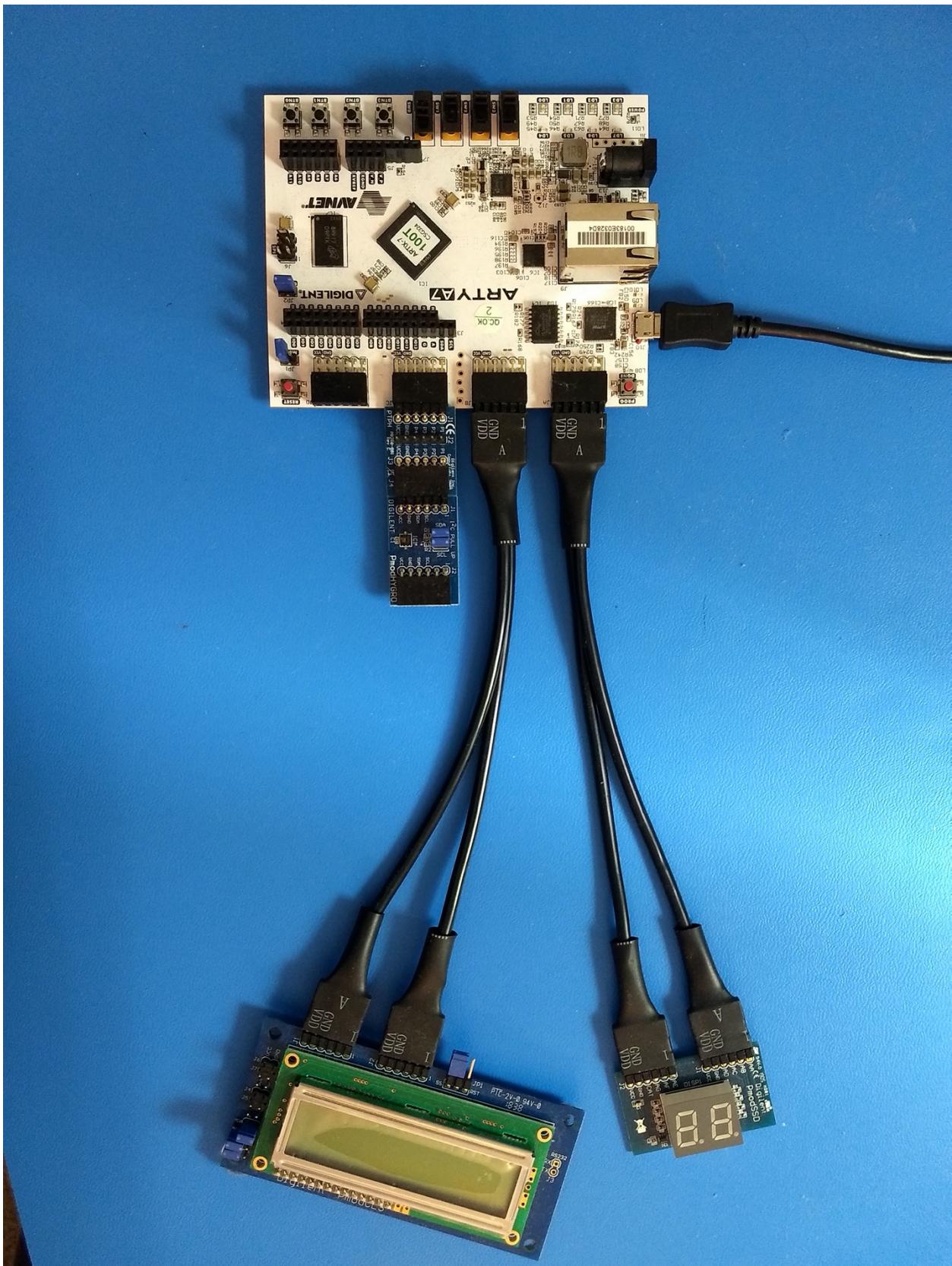


Figure 1: Arty-A7-100T Assembled with Pmod CLS, Pmod HYGRO, Pmod SSD

HYGRO Sensor Readings-Tester Experiment: Method of Operation: iterative poll of sensor readings and display on LCD

HYGRO Sensor Readings-Tester Experiment: Design Operation

In the Verilog implementation, the four switches are debounced as mutually exclusive inputs, as are the four buttons. To operate the design, the LCD displays an instruction to the operator to put Switch Zero in the ON position. After that, the LCD displays an instruction to the operator to push on of the four buttons.

While Switch Zero is in the ON position, the design polls the Pmod HYGRO HDC1080 sensor approximately once per second. Each time a button is depressed and then released while switch zero is positioned as ON, the incoming Pmod HYGRO sensor readings are converted to human readable content which in turn is displayed on the Pmod CLS LCD display. Pressing a different button (or the same button if button zero) changes the text and/or number format units on the LCD display.

Note that in the AXI design, drivers downloaded from Digilent Inc. for the Pmod HYGRO and Pmod CLS are used in the block design with some minimal modification and update to target the Arty-A7-100 board instead of the original Arty board, as well as configure the IIC bus for the AXI speed of 83 MHz. The AXI implementation integrates vendor components plus adding additional C code.

HYGRO Sensor Readings -Tester Experiment: Design Theory

Conceptual-only FSM diagrams are also included in this design repository in the PDF document HYGRO-Tester-Design-Documents/HYGRO-Tester-Design-Diagrams.pdf. The final FSM diagrams show the original FSM design prior to and during coding of the first draft; and the block descriptions assist with understanding the code architecture. More complete FSM diagrams will also be included in the document, following the simpler FSM diagram page, for each major FSM designed in the HDL designs.

Note that in the IPI-BD (called AXI) Design, drivers downloaded from Digilent Inc. for the Pmod SF3 and Pmod CLS are used in the block design with some minimal modification to target the Arty-A7-100T. Both drivers target the Arty-A7 instead of the Arty. The Git repository contains a submodule that pulls from a branch of the author's fork of the Digilent Vivado-library repository on GitHub.

Coding style and choices of block design

Software design practices were used to author the Verilog-2001 sources. After the sources were drafted with a large top-level module and cohesive modules for drivers, a large self-instruction homework experiment was converted into a standalone design.

HYGRO Sensor Readings-Tester Experiment: 3rd-party references:

Digilent Inc. References

Arty – Getting Started with Microblaze Servers

<https://reference.digilentinc.com/learn/programmable-logic/tutorials/arty-getting-started-with-microblaze-servers/start>

How To Store Your SDK Project in SPI Flash

<https://reference.digilentinc.com/learn/programmable-logic/tutorials/htsspif/start>

Vivado Board Files

<https://github.com/digilent/vivado-boards>

Master XDC files for all Digilent Inc. boards, including Arty-A7-100T

<https://github.com/Digilent/digilent-xdc>

Digilent Inc IP library for Xilinx Vivado

<https://github.com/Digilent/vivado-library/>

Textbook References

In the HDL sources and design diagrams document:

- Pulse Stretcher Synchronous, Textbook Figure 8.28a. quoted from,
- FSM design theory and methodology adapted by Tim S. and extended from,

Volnei A. Pedroni, *Finite State Machines in Hardware: Theory and Design (with VHDL and SystemVerilog)*. London: The MIT Press, 2013.

Use of IP Integrator to create the Microblaze AXI block diagram and synthesis:

- Tutorials followed from text to understand IPI block design,

L. H. Crockett, R. A Elliot, M. A. Enderwitz, and R. W. Stewart, *The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC*, First Edition, Strathclyde Academic Media, 2014.

Study of Verilog HDL IEEE 1364-2001:

- FPGA-relevant homework studied and applied for coding Verilog-2001 designs,

Samir Palnitkar, *Verilog HDL: A Guide to Digital Design and Synthesis*. 2nd ed., USA: SunSoft Press, 2003.

Suggestions for best practices when coding VHDL:

- Suggestion to code RTL design with as few MMCM/PLL generated clock domains as possible,
- Suggestion to exercise software design practices when coding VHDL,

Ricardo Jasinski, *Effective Coding with VHDL: Principles and Best Practice*. London: The MIT Press, 2016.