Wrocław University of Science and Technology



Faculty of Electronics, Photonics and Microsystems

Theme of class: Analog Discovery

Student: Hayrettin Aycetin (276807)

Date of class: 04.12.2023 15:15-16:55

Group No:3

Submition Date:18.12.2023

Lab assistant: Aleksander Kubeczek, Alicja Kwaśny GRADE:

Task 1

Make sure that Digilent Waveforms and Digilent Adept packages are installed (available on eportal).

A screenshot of a computer

Description automatically generated

Task 2

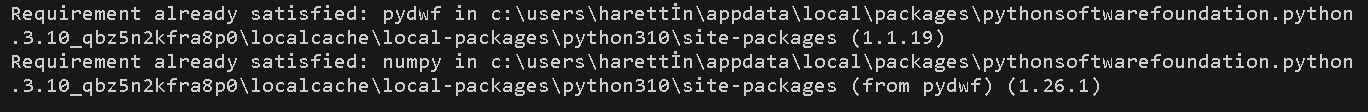
Install pydwf package [pydwf documentation] (it is recommended to use pip – see website for documentation).

A screenshot of a computer

Description automatically generated

I just wrote these commands to my terminal in Visual Studio Code.

Output:



A black background with white text

Description automatically generated



Comment:

I already installed pydwf so it said already satisfied.

Second code let me check the version.

Third code gives me an output of no digilent waweforms found because I tried on my computer without digilent discovery at home but in University Lab it worked without any problems.

Task 3

Extract example scripts provided by the pydwf authors.

A screenshot of a computer

Description automatically generated

I just wrote these commands to my terminal in Visual Studio Code.

A screenshot of a computer program

Description automatically generated

It automatically downloaded some files from the web to my directory.

Comments:

Everything was made only using two commands so I think not much to comment.

Task 4

Run AnalogInRecordMode.py example script on your Analog Discovery board (remember to connect the wires properly).

A screen shot of a computer program

Description automatically generated

Output:

A graph of a function

Description automatically generated with medium confidence

A diagram of a computer network

Description automatically generated

This is a guide to how to use Analog Discovery.

Comments:

This script operates by producing signals on two AnalogOut channels and recording them on corresponding AnalogIn channels. Ensure that you connect analog-out channel #1 to analog-in channel #1 and analog-out channel #2 to analog-in channel #2 before running the script. After setup and execution, the displayed graph will show the resulting waveforms.

Task 5

Run AnalogOutPlayCustomWaveformModified.py. (available on eportal) Generate some custom waveform and feed it to the instrument.

A white text with black text

Description automatically generated

Above is the code which is given on eportal.

Output:

A graph with blue lines

Description automatically generated

Comments:

This codes shows the analog output of a custom waveform.

Task 6

Run DigitalIO.py. Try to modify the script so that the digital input states are stored in some data structure (e.g. numpy.ndarray). Experiment with the interval that the digital input is read.

A screenshot of a computer program

Description automatically generated

Comments: I run this code successfully and saw some binary numbers in terminal but I couldn’t store them in a data structure.My code gave errors which I couldn’t solve with DigitalIO.py but I uploaded normally how would I store input states in a data structure below.

A screen shot of a computer program

Description automatically generated

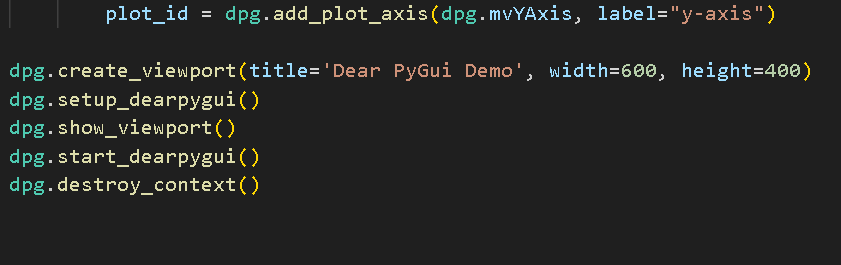
In this code I assumed I had five inputs and I assigned them one by one manually using np.zeros function.

Task 7

Write a very simple GUI for plotting data from either task 4 or 6. It is recommended to use Dear PyGui [Dear PyGui documentation], but you can choose to use other GUI library/framework.

A screen shot of a computer program

Description automatically generated



Output:

A screenshot of a computer

Description automatically generated

Comments:

This script uses the Dear PyGui library to create a simple graphical user interface (GUI) application for plotting random data. The GUI includes a window with a button labeled "Plot Random Data" and a plot area with x and y axes. When the button is clicked, a callback function generates a set of random x and y data points, updates the plot, and displays the result. The overall application operates within the Dear PyGui context, and the graphical interface is presented in a viewport. The script sets up the Dear PyGui event loop to handle user interactions, and upon closing the application, the Dear PyGui context is appropriately destroyed.

Conclusion

The Analog Discovery 2 stands out as an entry-level "lab-on-a-board," offering versatile functionalities such as a basic oscilloscope, logic analyzer, and function generator. Remarkably cost-effective compared to alternative tools, it consolidates multiple functions into one compact device. While it proves to be a valuable and feature-rich tool, one potential drawback lies in its software interface. Users may find the graphical user interface (GUI) somewhat lacking in certain aspects. Nevertheless, the hardware performs admirably, showcasing the convenience of having diverse tools integrated into a single device. Despite any GUI limitations, the Analog Discovery 2 effectively fulfills its intended tasks, making it a practical choice for various applications..