DDiscovery-Write-Data User Guide

This project is designed to write custom digital data sequences to devices using the Digilent Digital Discovery. All configuration parameters and data bits are managed through the `config.csv` file.

Dependencies

- Python 3.7 or higher
- dwfpy

Installation:

pip install dwfpy

File Structure

- `main.py`: The main program, responsible for reading configuration and data, then writing them to the device.
- `config.csv`: The configuration and data bits file (must be edited by the user).

---

config.csv Format Description

Example:

```
key,value
frequency,100
num_cycles_to_reset,2
length_of_data,16
repeats,2
clock_channel,24
data_channel,25
resetn_channel,26
splitter,========================
bitA,1
```

bitB,0
bitC,1
bitD,0
...

Parameter Descriptions

- **frequency**: Clock signal frequency in Hz (e.g., 100 means 100Hz).
- **num_cycles_to_reset**: Number of clock cycles the reset signal stays low before writing data.
- **length_of_data**: Number of data bits to write in one cycle (must match the number of data bit entries below).
- **repeats**: Number of times to repeat the data sequence; the same data will be written consecutively.
- **clock_channel**: Output channel number for the clock signal (Digital Discovery channels, usually 24~39).
- **data_channel**: Output channel number for the data signal (Digital Discovery channels, usually 24~39).
- **resetn_channel**: Output channel number for the reset signal (Digital Discovery channels, usually 24~39).
- **splitter**: Separator row, content can be anything, just to separate the parameters from the data bits.

Data Bit Instructions

- Every row after the splitter is treated as a data bit. The key can be named arbitrarily (does not have to start with `data_`), and the value must be either `0` or `1`.
- The number of data bits must match the value of `length_of_data`; otherwise, the program will throw an error.

How to Use

1. **Edit config.csv**
   Fill in the parameters and data bits as shown above. The keys for data bits can be named freely as long as the values are `0` or `1`.

2. **Connect the Digital Discovery device**

Run the main program
In the terminal, run:

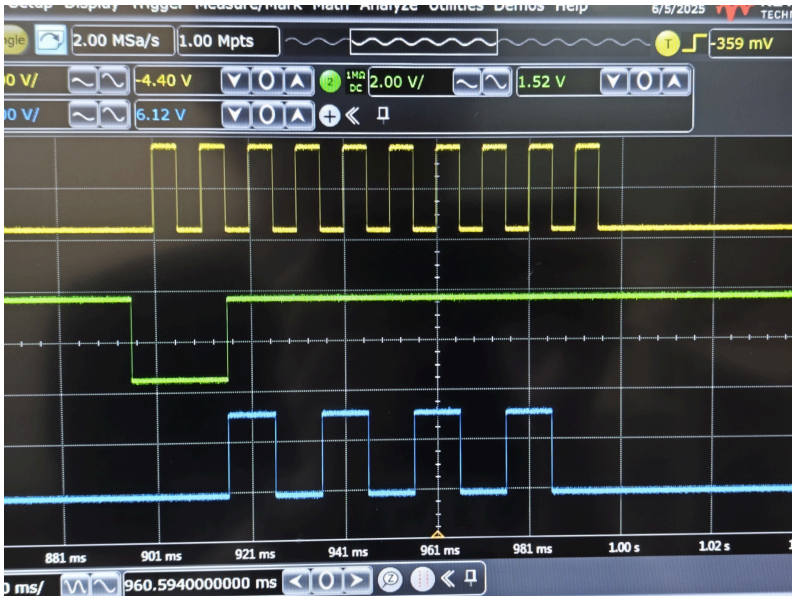
python main.py

3.
4. **Follow the prompts**
   Each time you press the Enter key, the data will be written to the device.

Example Behavior

If the configuration file is as shown above, the output waveform will look like the diagram.

```
key, value
frequency, 100
num_cycles_to_reset, 2
length_of_data, 4
repeats, 2
clock_channel, 24
data_channel, 25
resetn_channel, 26
splitter, ========================
data_0, 1
data_1, 0
data_2, 1
data_3, 0
```



Explanation: Since `repeats` is set to 2, the `1010` pattern is repeated twice.

Notes

- All parameters must be set in `config.csv`; the `config` dictionary in the code is just a placeholder.
- The number of data bits must match `length_of_data`.
- To change frequency, channels, or other parameters, simply modify `config.csv`.
- Data bit keys can be named freely; they do **not** have to start with `data_`.