

# DDiscovery-Write-Data 使用说明

本项目用于通过 Digilent Digital Discovery 向数字设备写入自定义数据序列。所有配置参数和数据位均通过 `config.csv` 文件进行管理。

## 依赖环境

- Python 3.7 及以上
- [dwfpy](#)

安装方式:

```
pip install dwfpy
```

## 文件结构

- `main.py`: 主程序, 负责读取配置和数据并写入设备
- `config.csv`: 配置参数和数据位文件(需用户自行编辑)

## config.csv 格式说明

示例:

```
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
...
```

## 配置参数说明

- `frequency`: 时钟信号频率, 单位为 Hz(如 100 表示 100Hz)
- `num_cycles_to_reset`: 复位信号持续的时钟周期数(写入数据前复位信号为低的周期数)
- `length_of_data`: 单次写入的数据位数(数据位总数, 需与下方数据位数量一致)

- **repeats**: 数据序列重复写入的次数, 将连续写入相同的数据序列多次
- **clock\_channel**: 时钟信号输出通道编号 (Digital Discovery 上的通道号, 通常为 24~39)
- **data\_channel**: 数据位信号输出通道编号 (Digital Discovery 上的通道号, 通常为 24~39)
- **resetsn\_channel**: 复位信号输出通道编号 (Digital Discovery 上的通道号, 通常为 24~39)
- **splitter**: 分隔符行, 内容可任意, 作用是分隔参数和数据位

## 数据位说明

- **splitter** 行之后的每一行都视为一个数据位, key 可自定义 (不必以 **data\_** 开头), value 只能为 0 或 1
- 数据位数量需与 **length\_of\_data** 参数一致, 否则程序会报错

## 使用方法

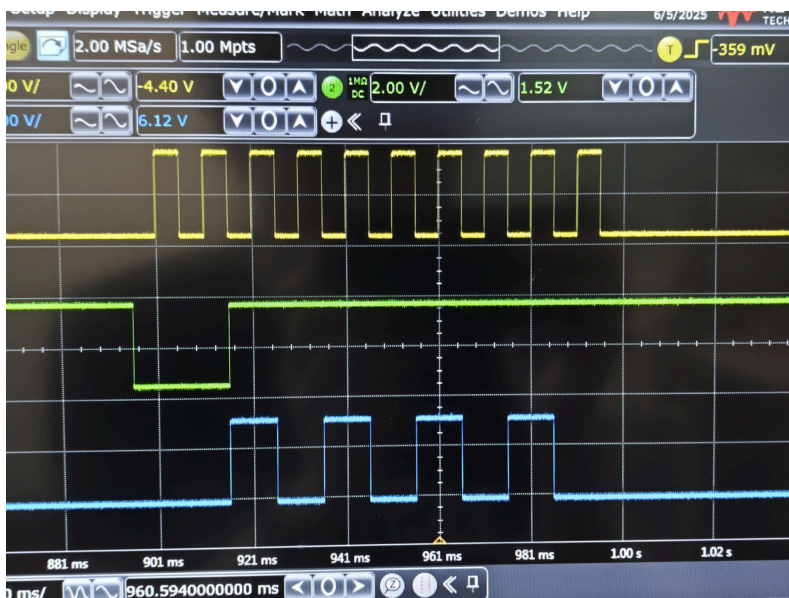
1. 编辑 **config.csv**  
按上述格式填写参数和数据位。数据位的 key 可任意命名, 只要 value 为 0 或 1。
2. 连接 **Digital Discovery** 设备
3. 运行主程序  
在终端中执行:  
  

```
python main.py
```
4. 按提示操作  
每次按下回车键, 数据会被写入设备。

## 行为举例

- 当配置文件如下时, 输出的波形如图

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



解释: 因为repeats是2, 所以1010这个patten重复了两次

## 注意事项

- 所有参数均需在 `config.csv` 中设置, 代码中的 `config` 字典仅作占位。
- 数据位数量需与 `length_of_data` 参数一致。
- 若需更改写入频率、通道等参数, 请直接修改 `config.csv`。
- 数据位 key 可自定义, 不必以 `data_` 开头。