

Changes in version 1.6.0

1. The configuration file is more human readable and has a future-proof dictionary-type format. In the past, the key strokes for the interactive inputs were stored in the configuration file. It's hard to read. Information in the new configuration file is organized in **key:value** pairs separated by commas. Here is what it looks like (only the most important parts):

```
{"ser": ["SDI12DIO01"], "sdi_12_address": "1", "sdi_12_command": ["0"], "delay_between_pts": 120, "total_data_count": 20000}
```

2. The adapter or serial port is referred to by its unique ID, such as SDI12DIO01, instead of serial port device name such as COM4 or /dev/ttyUSB0. Once the configuration file is set up, subsequent runs of the script will always connect to the correct adapter by ID instead of device names that can change. You can find or change the adapter ID. Here is how:
<https://liudr.wordpress.com/2018/07/03/open-ftdi-usb-serial-uart-port-by-id/>
3. The data file name now includes the configuration file's name, such as Liudr.cfg_20180709.csv
4. You can specify a different configuration file name in command line instead of the default.
5. There are also stability updates that prevent the script from quitting if, say, the sensor fails to return any valid response. Instead, the logger waits until the next data point and try again.
6. The interactive questions are slightly adjusted to support multiple sensors per adapter, as well as multiple sensing commands per sensor. You will be asked to enter all sensor addresses in one shot, such as '12z' for sensor addresses 1, 2, and z (the adapter's own address if you have analog or GPS features on your adapter). Later, for each sensor address, you will be asked what sensing commands you want to use. For simple sensors, enter '0'. For sensors that support multiple sensing commands, such as M! (aka M0!) for temperature and M1! for moisture for a multi-sensor probe, you enter '01'. You can also elect to not execute the script so you return to command line or Python shell and can run the code again to set up a different configuration.

The purpose of this update is:

- A. You can set up different configuration files for different purposes and run any configuration file you like without having to manually enter the parameter values again.
- B. If you plan to run multiple SDI-12 USB adapters on the same computer, you can create one configuration file for each adapter. The configuration file includes the unique ID of the adapter. You just execute the script simultaneously several times, with each execution using its own configuration file and only read the adapter it's supposed to read. You can tell which data file comes from which adapter because the configuration file's name is included in the data file's name.
- C. Your logger script will be less prone to quit on certain errors.
- D. Your logger can handle multiple sensing commands per sensor.

How to apply this update?

1. Rename your old configuration file and keep your logging script. If you made changes to my stock script, note the changes so you can apply them to the new script.
2. Start the new script, you will enter an interactive session and set up the script like before.

- Run the script like before. You can even copy the configuration file from PC to raspberry pi and expect it to work without reconfiguring.

How to take full advantage of this update?

- Assume we want to set up a configuration called backyard.cfg.

On Windows, enter "python sdi_12_logger_v1_6_0.py cfg:backyard.cfg" in command prompt.

On GNU/Linux/RPI, enter "python3 sdi_12_logger_v1_6_0.py cfg: backyard.cfg" in terminal.

If you are setting up multiple adapters on one computer, make sure you select the correct adapter when creating each configuration file.

- Now that they are all set up, start the script the same way you did in the previous step.

- If you are starting multiple adapters, each with a different configuration:

On Windows, enter "start python sdi_12_logger_v1_6_0.py cfg:config_name.cfg" in command prompt. A new command prompt window is started. You can then return to your original command prompt and start another one in the with a different configuration file.

On GNU/Linux/RPI, enter "python3 sdi_12_logger_v1_6_0.py cfg:config_name.cfg &" in terminal. You are returned to the terminal despite that the script you just started is still printing to the screen. You can then start another one the same way with a different configuration file. If you don't wish to see the printouts on screen, use '>>/dev/null &' instead of '&'.

The following demonstrates eight sensors deployed in two zones, A and B. Each adapter connects to four sensors in one zone. The sensor addresses need not be unique between zones, just unique within each zone. Up to 4 sensor/zone and 4 zones/rpi is recommended for easy setup and maintenance.

Multiple sensor and adapter deployment

- Each adapter runs a unique config file that contains the adapter's unique ID and controls one zone with up to 4 sensors
- Each sensor on the same adapter has a unique address. Sensors on different adapters can have the same address.
- Each raspberry pi (rpi) controls up to 4 zones.
- More sensor/zone or zones/rpi makes the system harder to set up or maintain

