



README:

Guide to the Project Folder

Last modified 2021-01-26

Executive Summary

Welcome to the O.H. Hinsdale Wave Research Laboratory and to your project. This guide is intended for use as an introduction and reference to the project folder that holds everything for your project, from planning and designing the project all the way through operations and data collection to post-processing and recovery. The guide is general in nature, so it might not cover some project-specific information.

Table of Contents

Executive Summary	1
Table of Contents	1
1. Introduction	2
2. Project Folder Access and Directory Structure	2
2.1 Project Folder Directory: <code>data</code>	2
2.2 Project Folder Directory: <code>docs</code>	3
2.3 Project Folder Directory: <code>images</code>	3
2.4 Project Folder Directory: <code>sw</code>	3
2.5 Project Folder Directory: <code>video</code>	3
2.6 Project Folder Directory: <code>wavemaker</code>	3
3. Data Files	4
3.1 Raw Data	4
3.2 Intermediate Data	6
4. Data Acquisition System (DAQ)	7
5. Data Acquisition Synchronization and Timing	8
6. Instrumentation and Wiring Details	9
7. Surveys and Location Plans	10
8. Calibration	10
9. Wave Generation	11
10. Post-processing	12

1. Introduction

This is a guide to the project folder. The guide is broken into sections, starting with an overview of access to the project folder and its directory structure. Following this project folder overview, the guide describes the structure of O.H. Hinsdale Wave Research Laboratory (HWRL) data files. The next section describes the HWRL data acquisition system (DAQ), including timing and synchronization. The connection of instrumentation to the DAQ and location of instrumentation in the facility is described, as well as methods used to calibrate instruments and the tracing of these calibrations. Wave generation is described next, and finally the guide finishes with an overview of the standard post-processing steps in use at the HWRL.

2. Project Folder Access and Directory Structure

Any online access to the project folder requires an OSU College of Engineering account. Dumping the folder contents to another location, such as a shared ftp site or DesignSafe or a portable drive, can provide offline access. For those who have an OSU College of Engineering account, online read access is always available. Write access for active projects is always available to HWRL staff. Guest access for writing to the project folder can be granted on a case-by-case basis to those with an OSU College of Engineering account. Once the project has ceased activity, the project folder will be locked down to read-only and any guest access for writing will be removed.

Accompanying this file in the project folder is a set of directories as follows: `data`, `docs`, `images`, `sw`, `video`, and `wavemaker`. Not all directories may be present; for example if no video was collected as part of the project then the empty `video` directory will eventually be removed when the project ceases activity. The project folder directories are described in the following sections.

2.1 Project Folder Directory: `data`

All observations from the project are stored in the `data` directory. Subdirectories such as `raw`, `inter`, and `final` are used to separate data processing stages. Each experiment for the project is a separate subdirectory with its own name, with trials contained in directories below those. Some experiment names such as `Calibration`, `DAQTest`, `SelfCalibration`, and `Survey`, will contain files or trials that are purely for testing or calibration or instrument location purposes.

The `raw` subdirectory includes data as recorded by the DAQ without any modification. Analog signals, such as wave gauges (*wg*) and acoustic range finders (*uswg*), are recorded in voltages. Digital signals are recorded with their own custom units; for example, VECTRINO signals are recorded in channels including *velocity* (m/s), *signal level* (counts), and *correlation* (%). To maintain data integrity, raw data cannot be deleted or modified. The structure of raw data files is described in section 3 on page 4.

The `inter` subdirectory contains data that has been post-processed from its originally recorded state. The originally recorded data is always retained in the `raw` subdirectory and never modified. The `inter` subdirectory will always contain a `Calibration` subdirectory that holds all the calibration summary sheets for the project, along with any applicable wave gauge calibration sheets. This `Calibration` subdirectory should not be changed or modified; doing so will break automatic scripts used for post-processing. Calibration is described in section 8 below. Post-processing is described in section 10 below.

2.2 Project Folder Directory: docs

All documentation and metadata from the project are stored in the `docs` directory. Various subdirectories may be used to separate different types of documentation, when appropriate. The `setup` subdirectory will always be present, with two subdirectories (`instm_locations` and `wiring_details`) always used to store spreadsheets that log instrument locations and the setup of instruments and their connectivity to the data acquisition system (DAQ). This `setup` subdirectory should not be changed or modified; doing so will break automatic scripts used for post-processing.

2.3 Project Folder Directory: images

All images from the project are stored in the `images` directory. Various subdirectories may be used to separate different types of images, when appropriate.

2.4 Project Folder Directory: sw

All source code used during the project is stored in the `sw` directory. Various subdirectories may be used to separate different types of source code, when appropriate. The `data_processing` subdirectory holds project-specific MATLAB routines for post-processing of data. The `hwr1_daq` subdirectory holds the LabVIEW source code used to run the data acquisition system (DAQ). The `wavemaker` subdirectory holds project-specific MATLAB routines for generating waves; the output from these routines is stored in the `wavemaker` project folder directory, described in section 2.6 below. Finally, a `toolbox` subdirectory may be present which holds a reference copy of the HWRL MATLAB toolboxes used for the project.

2.5 Project Folder Directory: video

All videos from the project are stored in the `video` directory. Various subdirectories may be used to separate different types of videos, when appropriate.

2.6 Project Folder Directory: wavemaker

All wavemaker input files from the project are stored in the `wavemaker` directory. Various subdirectories may be used to separate different types of wavemaker inputs, when appropriate. The `disp_bin` subdirectory holds the converted binary inputs used to run the wavemaker in displacement control mode; these are the actual files loaded by the wavemaker control computer after conversion from ASCII text. The `disp_txt` subdirectory holds ASCII text files specifying the board displacement; these are the files taken to the wavemaker control computer and subsequently converted to binary files for running waves. Finally, the `waves_txt` subdirectory holds ASCII text files specifying the free surface height at the face of the wavemaker piston(s).

In some cases, wavemaker input files will not be used, and instead built-in commands for the wavemaker control computer will be used. The distinction is typically noted in metadata headers of the data file, described in section 3 below.

MATLAB routines for wave generation are used to create all ASCII text files, whether displacement or free surface. Those routines, as well as built-in wavemaker operations, are described in section 9 below.

3. Data Files

Observations collected by the HWRL data acquisition system (DAQ) as part of the project are stored in data files. This section describes those files, for both raw data and intermediate data.

All HWRL data files are recorded in ASCII text. Headers are delimited by % symbols and contain metadata. Data are stored in tab-delimited columns. End-of-line characters are system-specific due to ftp transfers from the DAQ to the project folder. The use of ASCII text data with column tab delimiters and % header delimiters allows MATLAB to easily load data files using the built-in load function.

HWRL data file headers all follow the convention of a descriptive name using uppercase and lowercase letters with no spaces followed by a colon containing the value for that metadata field. Headers are also divided into sections for readability, with section names denoted by square brackets. Headers are preserved throughout processing steps and added to; never taken away.

All data files have the following section header, after which the data record begins:

```
% [Data]
```

The following two sections describe raw data and intermediate (post-processed) data file and filename conventions, including examples of headers.

3.1 Raw Data

Raw data files are those recorded by the DAQ without any modification. They are stored in the `data/raw` subdirectory, as described in section 2.1 above.

All raw data files have the following naming convention used to generate unique filenames:

```
[xxxxx.xxxx]_[daqname]_[daqfiletype][port/trigger id].[extension]
```

Every raw data file has a prefix with a file creation timestamp. The timestamp is in Coordinated Universal Time (UTC), with units of days, where 0.0000 corresponds to midnight UTC on January 1st of 1904. Note that this is a file creation timestamp and the data acquisition start time corresponding to the timestamp of the first sample is recorded in the file when appropriate. HWRL DAQ timestamps are described in section 5 below.

The next section of the file name is a name for the DAQ computer that recorded this file. This field exists to prevent overwriting of the same data files recorded at the same time by multiple computers for an experimental trial.

The next section of the file name identifies the type of the DAQ file. Typically this is `analog` or `adv` for time series data. Other possibilities include: `mta` or `lrf` for bathymetric survey data, and `pressure` or `uswg` for calibration data for some instruments, such as pressure gauges or ultrasonic range finders.

The next section of the file name identifies the port being used, or the triggering mode in use by that system. Ports are a numerical identifier (1, 2, 3...) that immediately follow the `daqfiletype` field. Triggering mode can be: `_leader` for systems that are sending out trigger pulses for synchronization purposes, or `_follower` for systems that are listening for trigger pulses for synchronization purposes, or `_off` for systems that are not being sampled synchronously.

Finally, the file extension is usually `txt` to denote an ASCII text file. Other possibilities are `log` for logs from data conversion of Vectrino binary data to ASCII, `raw.bin` for those Vectrino binary data files, and `raw.hdr.txt` for the accompanying text header for those Vectrino binary data files. You should not be concerned with anything regarding the binary data files or logs, unless you are either conducting testing of the DAQprocessor or debugging Vectrino operations.

The rest of this section describes those raw data files that are recorded in ASCII text. Raw metadata headers are formatted as described in section 3 above. Following are some examples of useful metadata that you might find in different raw data files, including section names, headers, and values. Note that the sections are deliberately separated in this example by blank lines to illustrate that the examples are only pieces of actual headers. Empty lines between sections are marked in real headers with the `%` delimiter.

The following headers are for an example of a raw analog data file:

```
% [O.H. Hinsdale WRL Metadata]
% FileType: DAQFile
% ProjectName: YourProjectName
% ExperimentName: OneOfYourExperimentNames

% [NEES Metadata]
% ProjectName: YourProjectName
% ExperimentName: OneOfYourExperimentNames
% TrialName: Trial91
% TrialTitle: Trial 91
% TrialDescription: Regular waves; 10 waves + 40s ramp up/down, DAQ for 4min
% TrialConditions: H=0.45m, T=5s, h=2.448m, REG_H0.45_T5_h2.448_t90_disp.bin

% [Analog Input Metadata]
% AnalogInputDeviceType: PXI-6259
% ChannelCount: 1
% DAQHardwareChannels: SC1Mod1/ai0
% MinVoltage: -5.000000 V
% MaxVoltage: 5.000000 V
% LowpassCutoffFrequency: 25.000000 Hz
% SampleRate: 50.000000 Hz

% [Timing Metadata]
% StartDateTimeUTC: Wed 2015/12/16 17:36:49.808 UTC
% StartYearDayUTC: 40892.7339098172

% [Channel Metadata]
% ChannelNames: wmstart
% ChannelUnits: Volts
```

The following headers are for an example of a Vectrino data file:

```
% [Serial Port Metadata]
% SerialPortName: ASRL7 (COM7 - PXI-8420/16)

% [Vectrino Hardware Configuration Metadata]
% HardwareSerialNumber: VNO 0099

% [Vectrino Probe Head Configuration Metadata]
% HeadSerialNumber: VCN 8173
```

```

% [Vectrino ADV Header]
% Temperature:          9.570 degC
% SpeedOfSound:        1445.500 m/s

% [Channel Metadata]
% ChannelNames: error, u, v, w1, w2, amplitude1, amplitude2, amplitude3, amplitude4,
correlation1, correlation2, correlation3, correlation4
% ChannelUnits: n/a, m/s, m/s, m/s, m/s, counts, counts, counts, counts, %, %, %, %

```

Raw data is stored in tab-delimited columns. Each column is a channel of data, and each row is a time step. The sample rate of the data in the file is stored in the headers, and so is the start time. More details on synchronization and timing are provided in section 5 below.

3.2 Intermediate Data

Intermediate data files are data files that have been post-processed from their originally recorded (raw) states. They are stored in the `data/inter` directory, as described in section 2.1 above. Post-processing is described in section 10 below.

All intermediate data files have the following naming convention. Note that all intermediate data files are ASCII text:

```
[channelname].txt
```

The `channelname` prefix can correspond to an analog data channel name from an analog data file. If the channel name is `wmstart` from the previous examples, the resulting filename will be `wmstart.txt`. The prefix can also correspond to a Vectrino channel name (such as `u` from the previous examples) paired with the port of that Vectrino (such as `1`). In this case, the channel name prefix will be `u1` and the filename will be `u1.txt`.

Intermediate metadata headers are formatted as described in section 3 above. Following are some examples of useful metadata that you might find in different intermediate data files, including section names, headers, and values. Note that the sections are deliberately separated in this example by blank lines to illustrate that the examples are only pieces of actual headers. Empty lines between sections are marked in real headers with the `%` delimiter.

The following headers are for the example of an intermediate analog data file (`wmstart.txt`) described previously. Note that some of the headers don't have valid values, because the `wmstart` channel doesn't have a location in the HWRL coordinate system:

```

% [Calibrated Metadata]
% StillWaterDepth: 2.4479e+00
% ChannelName: wmstart
% CalibrationSlope: 1.000
% CalibrationOffset: 0.000
% CalibrationUnits: V
% X:   NaN
% Y:   NaN
% Z:   NaN

```

The following headers are for the example of an intermediate Vectrino data channel (u1.txt) described previously:

```
% [Calibrated Metadata]
% StillWaterDepth: 2.4479e+00
% MeanSNR: 27.069
% MeanCorrelation: 98.203
% TemperatureOK: YES
% ChannelName: u1
% CalibrationSlope: 1
% CalibrationOffset: 0
% CalibrationUnits: m/s
% X: 45.018
% Y: 0.568
% Z: 1.562
```

Intermediate data files are recorded as single columns, so that there is one file for each channel, with its own channel name as a file name as described previously. The time coordinate for intermediate data files is stored in a separate file (time.txt) and the post-processing steps to create that file and all other intermediate files are described in section 10 below on post-processing.

4. Data Acquisition System (DAQ)

The data acquisition system (DAQ) used at the HWRL runs on National Instruments PXI architecture computers. These computers use a real-time version of the LabVIEW programming environment. Computers are upgraded approximately every three years. The LabVIEW programming environment is updated annually, typically on the spring release cycle of the campus-wide site license in place at Oregon State University. This is done so that the LabVIEW programming environment in use is the presumably more stable “Service Pack 1” programming environment instead of the first release of a new environment. Each computer is installed in a PXI chassis along with accompanying PXI and or SCXI modules.

Analog data acquisition for each DAQ is controlled by a NI PXI-6259 M-series 16-bit multifunction DAQ module. The DAQ module communicates via the PXI/SCXI backplane to SCXI-1143 Butterworth anti-aliasing filter modules in its containing PXI-1052 chassis. Each filter module is fronted with SCXI-1305 terminal blocks that take $\pm 5V$ differential inputs from analog channels via 50Ω coaxial cable with BNC connectors. The SCXI-1143 Butterworth anti-aliasing filters are set with cutoff frequency at $\frac{1}{4}$ the sampling rate. There are eight filter modules per PXI-1052 chassis and 8 channels per filter module, for a total of 64 channels of analog inputs per chassis. Synchronization of up to 3 independent chassis permits up to 192 synchronized analog input channels. The DAQ sampling rate is typically 100 Hz; the DAQ is capable of up to 5 KHz when sampling all 64 channels. The DAQ assumes that all input sources are ground-referenced; therefore, it has no resistor to ground on the low (negative) side of each differential input. Testing at the HWRL has shown that the DAQ is accurate to within $\pm 0.3mV$ when observing known voltages on NIST-traceably certified voltage standards.

All observed analog input signals are subject to some unwanted signals (noise). Noise is minimized through careful controls at all stages of the DAQ, including but not limited to the use of shielded cabling, isolation of instrumentation power supplies from non-instrumentation outlets, elimination of ground loops, high-quality equipment from trustworthy vendors, and regular traceable calibrations. All in-situ instruments and equipment are calibrated using methods traceable to NIST standards, and the HWRL routinely maintains calibration sheets and documentation of these traces, as described in section 8 below.

Digital communication with Vectrino systems is done over RS-232 serial ports. A 16-port PXI-8420 module controls serial data acquisition for each DAQ. Vectrino systems are triggered by TTL synchronization pulses as described in the next section.

All observed data are written continuously to a solid-state drive on the DAQ. Data is copied via ftp to an independent data processor (Mac Mini running macOS) that keeps a locally backed-up archive of each data file before sending a copy onward to the HWRL share. The HWRL share is located on-campus in a separate building from the HWRL, and it is maintained and backed up by the College of Engineering information technology support team. This process ensures that at least three multiply-backed-up copies are made of each data file before it is made available for post-processing and quality control.

5. Data Acquisition Synchronization and Timing

Accurate synchronization and timing of observations are important concerns. The HWRL provides several options for synchronizing its own systems with each other and with those of visitors. Options for sample synchronization and absolute timing are reviewed in this section.

Each NI PXI-6259 M-series DAQ module has an 80 MHz sample clock that is accurate to within ± 50 ppm ($\pm 0.005\%$) and a resolution of 50ns. The DAQ modules are kept in spec through routine on-site self-calibration and traceable calibration every 2 years, per the recommendations of National Instruments. It is most preferable to connect instruments directly to the HWRL DAQ analog input channels, which guarantees sample-for-sample timing to within the DAQ sample clock resolution.

If it is not possible to connect an instrument to the analog inputs of the HWRL DAQ, or if multiple HWRL DAQ systems are required due to the size of a project deployment plan, then sample coherency is maintained through the use of TTL pulses generated by one leader DAQ and listened to by the other follower DAQs. This method is also used to ensure sample coherency between analog observations and Vectrino systems. All Vectrinos at the HWRL are set up in sample-on-sync mode and triggered by a leader DAQ. TTL pulses generated by the leader DAQ are buffered and repeated by Pulse Research Labs PRL-414B 1:4 50 Ω TTL line drivers. TTL pulses generated by the leader DAQ are optimized for Vectrino triggering with +5V highs at 40 μ s widths, which corresponds to a 0.004 duty cycle when operating at 100Hz.

If connection to the HWRL DAQ and triggering will not work, then simultaneous recording of common signals can be used along with post-processing realignment to achieve synchronization. For example, the wavemaker outputs a 5V high at the start of operations and holds it high until after the board is stopped and current span has wound down to 0. This is the *wmstart* signal described previously in examples of data files in section 3 above. It is an analog output transmitted over 50 Ω coaxial cable and recorded as an analog input channel on the HWRL DAQ, and it can be buffered and sent to other systems for simultaneous recording and then post-processed to align signals afterwards. The accuracy of this procedure is to within ± 1 sample of the slowest-sampling system. This *wmstart* signal can also be used with mobile instruments over rigid beds to reconstruct a synoptic data set from multiple trials of the same wave conditions. Another example is an LED that turns on and off with random delays between 0.25s and 5s. The driver also outputs to the DAQ on a separate channel named “led”, with 0V corresponding to the LEDs being off and 5V corresponding to the LEDs being on. This gives an unambiguous way to synchronize multiple DAQs recording the voltages as well as cameras recording video, to within ± 1 sample of the slowest-sampling system.

The remaining options for synchronization rely on absolute timing: the use of timestamps. All HWRL DAQ timestamps use Coordinated Universal Time (UTC). Timestamps from HWRL DAQ system clocks are provided for the first sample of analog DAQ systems in two headers, shown below in an example:

```
% [Timing Metadata]
% StartDateTimeUTC: Wed 2015/12/16 17:36:49.808 UTC
% StartYearDayUTC: 40892.7339098172
```

The first header (`StartDateTimeUTC:`) is a human-readable UTC date (YYYY/mm/DD format) and time (HH/MM/SS.uuu format) with millisecond resolution. The second header (`StartYearDayUTC:`) has units of days with resolution of under $10\mu\text{s}$, where 0.0000 corresponds to midnight UTC on January 1st of 1904.

HWRL DAQ system clocks are synchronized to a grandmaster clock using the National Instruments NI-TimeSync implementation of the IEEE 1588-2008 Precision Time Protocol. The grandmaster clock is a dedicated PXI-based HWRL DAQ system with an NI-PXI-6682 timing and synchronization module connected to a Trimble Bullet III GPS antenna mounted on the roof of the HWRL. The GPS signal provides $\pm 100\text{ns}$ accuracy relative to UTC. The grandmaster clock sends synchronization messages over the local firewalled HWRL DAQ network every 500ms. Testing shows that HWRL DAQ timestamps agree to within $\pm 7\text{ms}$ when observing triggered coherent samples of the same analog signal.

Systems that lack their own GPS antenna and receiver can use NTP to set clocks to UTC. Available NTP servers are the stratum 2 server at OSU: `time.oregonstate.edu` and the United States NTP pool: `0.us.pool.ntp.org` through `3.us.pool.ntp.org`. Testing shows that this yields absolute clock accuracy to within $\pm 50\text{ms}$.

6. Instrumentation and Wiring Details

All HWRL project instrumentation deployments are tracked using wiring details spreadsheets. The purpose of the wiring details spreadsheets is to enable debugging and diagnostics of experimental setups. The spreadsheets are stored in the `docs/setup/wiring_details/` subdirectory in the project folder. The file names include the project name and a local timestamp. Each modification to the project instrumentation deployment or drivers or DAQ wiring or cabling is noted in an updated sheet. Wiring details sheets and their filenames and parent directories should not be changed or modified; doing so will break automatic scripts used for post-processing. An example of the top two rows of a wiring details sheet is shown below:

data column	sensor name	cable to driver	driver	driver channel	cable to DAQ	DAQName	DAQHardwareChannels
standard	VSTD-8241	NA	NA	1	CX-05-188	pxi2	SC1Mod1/ai0

The *data column* corresponds to the channel name recorded by the DAQ. The *sensor name* is a unique identifier for each sensor that is a combination of an abbreviated text field, such as VSTD for a voltage standard, and an abbreviation of the corresponding equipment serial number. If the sensor has a separate driver, such as a pressure gauge that consists of the submerged gauge itself and the signal conditioning module outside the tank, then that driver, driver channel, and cable between the driver and the sensor are logged in the wiring details. Finally, the rightmost columns describe the cabling between the sensor or driver to the DAQ, the name of the DAQ to which the sensor is cabled, and the input channel for that sensor. In this way, the wiring details sheet is a complete description of the project setup.

7. Surveys and Location Plans

Locations of instruments are typically found using a Nikon pulse laser station model NPL 332. This total station is precise to within $\pm 0.5\text{mm}$ and accurate to within $\pm 5\text{mm}$ when running in precise mode without a prism. Prior to survey, the station is leveled to within ± 0.05 arc-second (or $1/7200^{\text{th}}$ of a degree) and targeted at up to 8 known points around the wave tank with typical standard deviation of the resection at less than $\pm 1\text{mm}$.

All HWRL project instrumentation locations are tracked using location plan spreadsheets. The spreadsheets are stored in the `docs/setup/instm_locations/` subdirectory in the project folder. The file names include the project name and a local timestamp. Every change to instrument locations is noted in an updated spreadsheet. Examples of changes to instrument locations are: addition of sensors, removal of sensors, or moving of existing sensors. Location spreadsheets and their filenames and parent directories should not be changed or modified; doing so will break automatic scripts used for post-processing. An example of the top of a location plan spreadsheet is shown below:

data column	sensor name	coord space	placement (X)	placement (Y)	placement (Z)	placement (I)	placement (J)	placement (K)	comment
standard	VSTD-8241	Large Wave Flume							
wmstart	wmstart	Large Wave Flume							
wmdisp	TMPO-LWM	Large Wave Flume		0.000					
wmwg	RWG-LWM	Large Wave Flume		0.000					
level	PRES-9959	Large Wave Flume	13.961	-1.527	0.638				
wg1	RWG-2260-01	Large Wave Flume	17.728	-1.269					

The *data column* and *sensor name* entries must match those in the corresponding wiring details spreadsheets, as described in section 6 above. The *coord space* field describes which facility is used for this location plan, and then the following fields describe position and orientation of the sensor along with any comments or notes.

All surveys and instrument locations are reported using the HWRL coordinate system. The HWRL coordinate system is defined as follows: The x-axis is the cross-shore coordinate. Its origin ($x = 0$) is at a vertical plane that best fits the face of the wavemaker piston when it is neutrally positioned. The x-axis is measured in meters and positive onshore (away from the wavemaker). The z-axis is the vertical coordinate. The z-axis origin ($z = 0$) is at the average elevation of the tank floor. The z-axis is measured in meters and positive upwards. Finally, the y-axis is the alongshore coordinate (parallel to the wavemaker piston). The y-axis origin ($y = 0$) is at the alongshore centerline of the tank, i.e. halfway between two vertical planes that best fit the tank walls. The y-axis is measured in meters and positive to the left when facing onshore, so that the coordinate system is right-handed. In the Large Wave Flume, x is positive north and y is positive west. In the Directional Wave Basin, x is positive west and y is positive south.

8. Calibration

All HWRL project instrument calibrations are tracked using calibration summary spreadsheets. The spreadsheets are stored in the `data/inter/Calibration/` subdirectory in the project folder. The file names include the project name and a local timestamp. Every change to instrument calibrations is noted in an updated spreadsheet. Calibration summary spreadsheets and their filenames and parent directories should not be changed or modified; doing so will break automatic scripts used for post-processing. An example of the top of a calibration summary spreadsheet is shown below:

data column	sensor name	driver	driver channel	Calibration source file	slope	slope units	offset	offset units	Notes
standard	VSTD-8241	NA	1	NA	1.000	V/V	0.000	V	
wmstart	wmstart	LWF-WM	AO1	NA	1.000	V/V	0.000	V	
wmdisp	TMPO-LWM	LWF-WM	AO2	NA	0.440	m/V	0.000	m	
wmwg	RWG-LWM	LWF-WM	AO3	NA	0.400	m/V	0.000	m	
level	PRES-9959	SG-SC-6381	1	cert_pres_9959_20131009	1.043	m/V	-0.002	m	
wg1	RWG-2260-01	RWG-2260	1	YourProjectName_wave_gauge_calibration_20150126_1330	0.226	m/V	0.000	m	

Entries in the first four columns (*data column*, *sensor name*, *drive*, and *driver channel*) must match those in the corresponding wiring details spreadsheets, as described in section 6 above. The *calibration source file* field describes what source file was used to generate the provided calibration. This could be “NA” for calibrations that have no source file; for example the wavemaker output calibration coefficients are set directly at the wavemaker control computer. The calibration slopes and offsets and accompanying units are in the remaining columns, along with any comments or notes. Since raw data is acquired in volts, the slopes will always be expressed as a ratio of physical units to volts, and the offset will be in the physical units. Some signals have physical units of volts, for example *standard* (a voltage standard) and *wmstart* (a wavemaker output that is set to +5 V when the wavemaker is moving and 0 V when it is not).

Most HWRL project instrument calibrations are traceable to NIST standards, and calibration certificates document that traceability. Applicable calibration certificates are referred to in the calibration source file field of the calibration summary sheet, and they are available upon request. Vectrinos are calibrated at the factory (NorTek Vectrino+, 4 m/s maximum velocity, $\pm 0.5\% \pm 1$ mm/s accuracy) and do not require recalibration unless they are damaged or bent. Other instruments, such as acoustic range finders, laser range finders, load cells, and pressure gauges, are traceably calibrated either yearly or as needed. The HWRL DAQ, voltage standards, and associated diagnostic equipment are calibrated routinely either by the manufacturer or by NIST-certified calibration services. Finally, the manufacturer (MTS Systems Corporation) calibrates the wavemaker annually.

All wave gauges are calibrated when changing the tank water level (filling and draining). Fill and drain calibration methods are the same. While the water level is slowly changing, the HWRL DAQ observes all analog inputs over a sample period, typically sampling at 100Hz for a 1-minute duration. Sampling of analog inputs is typically done every 5 minutes for an entire fill or drain, which can take up to 9 hours. The mean voltage of every input channel is estimated for each sample period. Mean voltage estimates are then put into a wave gauge calibration spreadsheet that calculates a linear least-squares fit between the observed wave gauge voltages and the calibrated water depth observations from a traceably-calibrated pressure sensor, referred to as *level* in the HWRL DAQ and in the previous example of a calibration summary sheet, above.

In addition, some wave gauges (referred to as self-calibrating wave gauges) can be calibrated without changing the water level. The self-calibrating wave gauges are equipped with a lead screw of known pitch, a stepper motor, and a rotary encoder. Each set of self-calibrating wave gauge wires are driven up and down by the stepper motor, and the accompanying encoder position is observed. Combining the observed position of the encoder with the known pitch of the lead screw provides an observation of vertical motion of the wave gauge. The observed voltages are linearly least squares fitted to the observed vertical motion, generating a calibration slope. Self-calibrating wave gauges are typically calibrated at the beginning and end of every day.

9. Wave Generation

The wavemaker for the Large Wave Flume (LWF) is a single-channel piston-type hydraulically driven system. It consists of a vertical wall, referred to as the wavemaker piston, which is suspended from a steel support structure. The piston and its support structure travel back and forth on linear bearings that are bolted to the walls of the flume. The piston is driven by a set of counter-balanced hydraulic actuators that can exert up to 50,000 lbs of force with a maximum hydraulic flow rate of 650 gpm. The piston can move at up to 4 m/s and has a 4 m total stroke, which is defined as the distance between its most offshore and most onshore positions. The maximum still water depth of the LWF is 9 ft (2.743 m), and the maximum still water depth for running tsunamis (solitary waves) is 2 m.

The wavemaker for the Directional Wave Basin (DWB) is a 30-channel piston-type belt-driven servomotor system. It consists of a segmented vertical wall, referred to as the wavemaker piston. Each segment of the wall has a drive point on either side; there are 29 segments in total being driven by 30 drive points. Each drive point is moved by a 21.3kW servomotor and toothed belt that moves the piston back and forth on a bearing rail connected to supporting steel that is bolted into the basin floor. Each drive point can move at up to 2 m/s and has a 2.1 m total stroke, which is defined as the distance between its most offshore and most onshore positions. The maximum difference between adjacent drive points is 0.25 m. The maximum still water depth of the DWB is 4.5 ft (1.372 m), and the maximum still water depth for running tsunamis (solitary waves) is 1 m.

Each wavemaker can use built-in code to run regular (monochromatic) linear wave conditions of the desired height and period for a given water depth. Each wavemaker can also use built-in code to run solitary waves of the desired height for a given water depth. All other options for wavemaker operation are generated using MATLAB code and then provided as ASCII text input files to the wavemaker control computer as a time series of board displacement. These options are described below.

The HWRL MATLAB toolbox includes a set of routines for wavemaker theory. All routines generate time series of board displacement and free surface elevation at the wavemaker piston. For unidirectional monochromatic waves, the routine `wavemaker_reg_2d` automatically selects the best theory to generate waves that match the input height and period for a given water depth. The theory selected will be one of the following: fully nonlinear second-order Stokes wavemaker theory, ad-hoc stream function wavemaker theory, or cnoidal waves combined with shallow water wavemaker theory. For unidirectional random waves, the routine `wavemaker_n1` generates waves using fully nonlinear second-order Stokes wavemaker theory. For directional regular or random waves, the routine `wavemaker` generates waves using linear wavemaker theory. All routines generate ASCII text input files that are converted to binary files by the wavemaker control computer. Those binary files are then loaded by the wavemaker control computer and used to run waves in displacement file mode.

All custom source code used to specify and/or generate wavemaker files is stored in the `sw/wavemaker` subdirectory of the share, as described previously in section 2.4 above. All ASCII text and converted binary files are stored in the `wavemaker` directory of the share, as described previously in section 2.6 above. Source code from the HWRL MATLAB toolbox may also be provided in the `sw/toolbox` subdirectory; the section specific to wave generation would then be in the `sw/toolbox/wavemaker` subdirectory.

10. Post-processing

Post-processing is the conversion of data from a raw state to an intermediate state. Raw data is the same as recorded by the DAQ without any modification. Intermediate data is the output of post-processing. Post-processing includes the updating of metadata to include locations (section 7 above), the application of calibrations (section 8 above), despiking and data cleanup. Later steps may include estimation of incident and reflected waves, directional spectra, or statistical quantities such as significant wave height or peak wave period.

All custom source code used to post-process is stored in the `sw/data_processing` subdirectory of the share, as described previously in section 2.4 above. Source code from the HWRL MATLAB toolbox can also be provided in the `sw/toolbox` subdirectory; the section specific to post-processing data from the `data/raw` subdirectory to the `data/inter` subdirectory would then be in the `sw/toolbox/raw` subdirectory. Every processing stage retains headers from the previous stage, as described in section 3 above.

The top-level routine for post-processing is `process_raw_experiment`. It calls `process_raw_trial` for every trial in the input experiment. Both of these routines still have minor project-specific customizations, and so they are not yet in the HWRL MATLAB toolbox. The `process_raw_trial` routine loads all data found for the specified trial. Instrument locations are automatically applied from the most recent instrument location plan (section 7 above). Calibration coefficients and offsets are linearly interpolated between calibration coefficients found in summary sheets (section 8 above) for each channel of data. The system is able to handle calibration coefficients that are updated more frequently, such as the twice-daily calibration of self-calibrating wave gages. Vectrino data is cleaned up, despiked, and rotated into the HWRL coordinate system (section 7 above). Ultrasonic range finders are cleaned up and despiked.