BELFAST MAINE TIDE STATION TECHNICAL DISCRIPTION

SEPTEMBER 9, 2023

1. Overview

The Belfast Tide Station provides internet access to real-time and predicted tide and weather information for use by the Belfast boating community and other interested parties. It also provides a local display in the Harbormaster's office for use by the Harbormaster staff to help evaluate tide and weather conditions contributing to Belfast harbor operations. The information provided includes the current tide level measurement, a graph of measured and predicted tides, current weather conditions, and forecast weather and marine conditions for Belfast and the Upper Penobscot Bay regions. The Tide Station also provides the capability of plotting historical tide and weather data for further analysis.

The hardware and software comprising the Belfast Tide Station consists of a Raspberry Pi (RPi) computer, various sensors, Python scripts, External Interfaces, and the InfluxDB and SQLite3 Data Base Management Systems. The tide station collects and processes tide and weather information for presentation on a 12-inch display located at the Belfast Harbormaster's office and provides internet access via the home page found at the URL belfasttide.org.

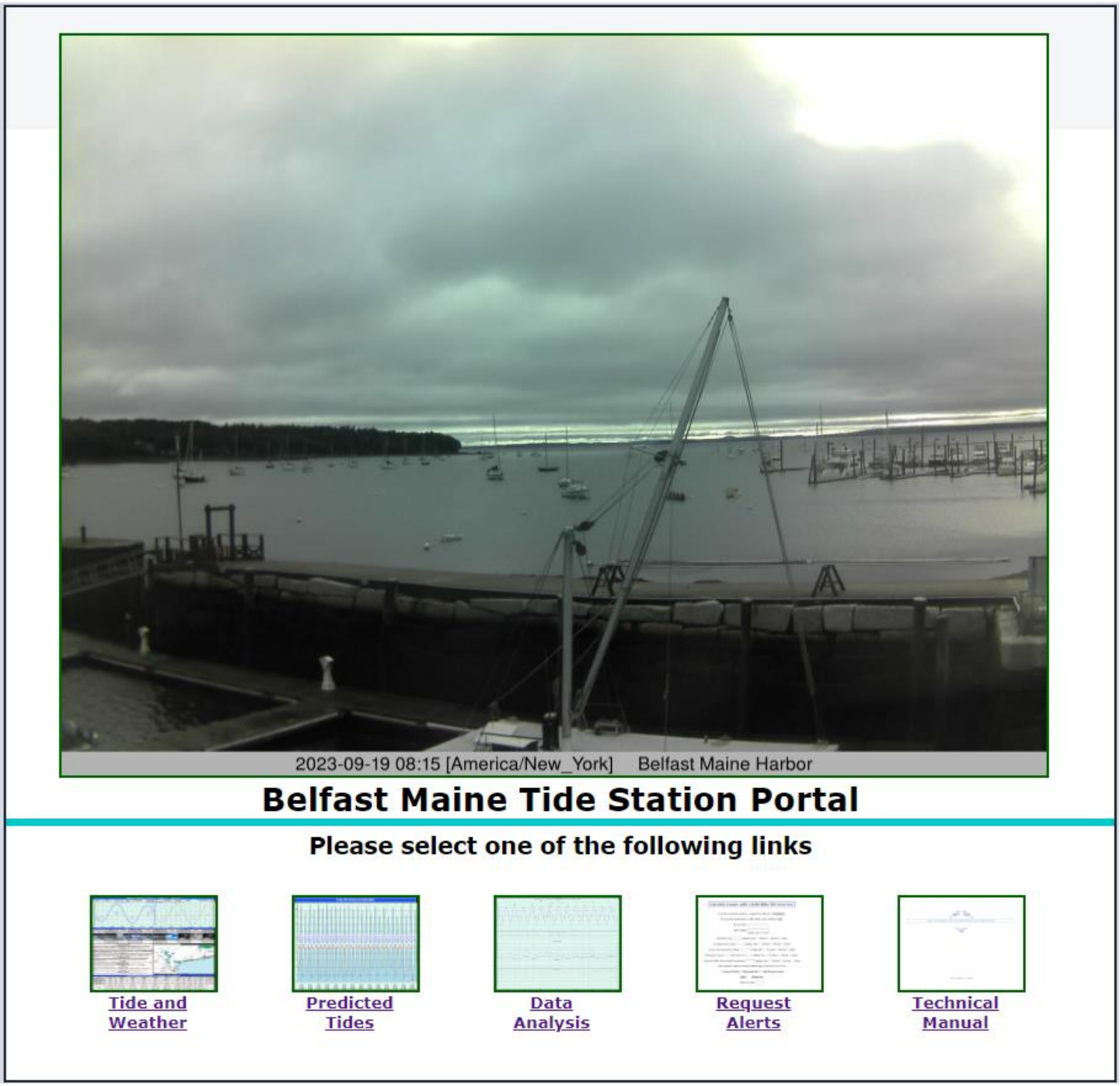

Figure 1. Tide Station Website Home Page

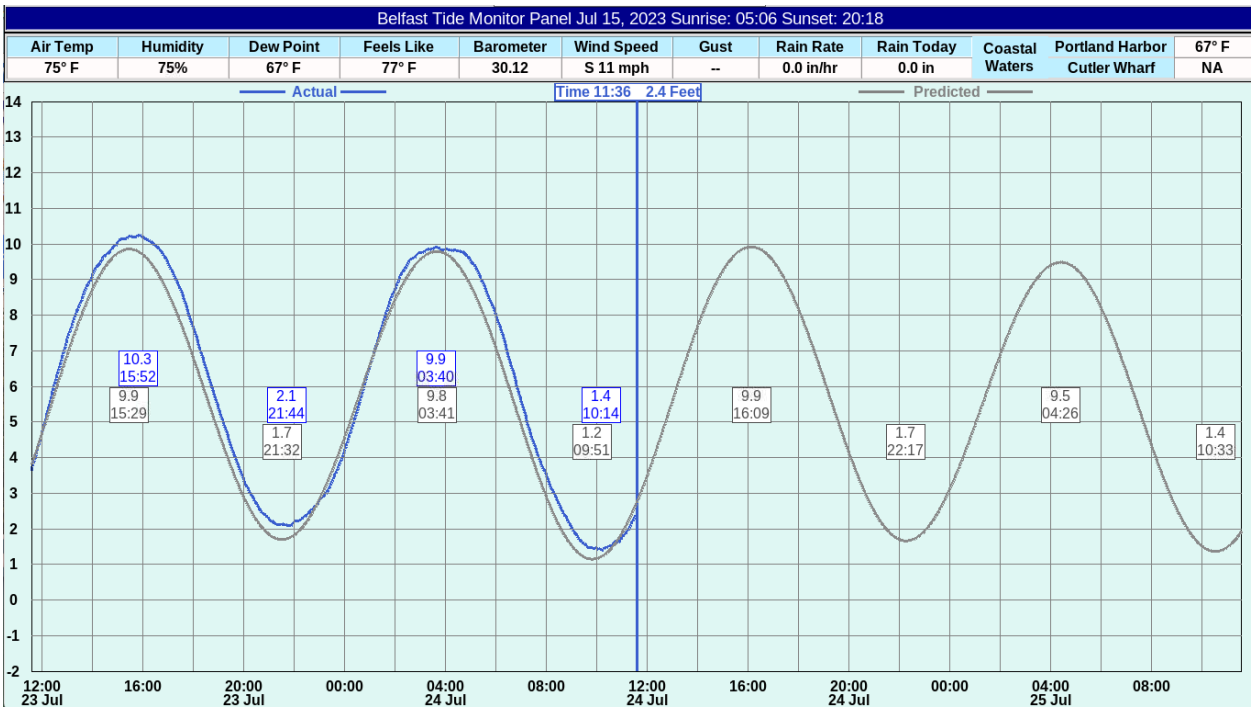

Figure 2. Belfast Harbormaster Tide Station Monitor Panel Local Display

2. Tide Station Components

The Tide Station utilizes the following input and sensor data:

a. Ultrasonic and radar tide-level sensors, located at the Belfast City Dock.
b. Weather conditions (temperature, humidity, wind, rain, barometric pressure) provided by a Davis Instruments Vantage Pro weather station, located at the Belfast City Dock.
c. Tide predictions for Belfast provided by the National Oceanic and Atmospheric Administration (NOAA).
d. Water Temperatures for Cutler Wharf and Portland Maine provided by NOAA.
e. 5-day weather forecast for Belfast provided by the National Weather Service (NWS).
f. Hourly marine forecast for Upper Penobscot Bay provided by the NWS.

3. Tide Level Sensors

Ultrasonic and radar sensors are installed at the Belfast City Dock in a manner which allows a direct vertical line-of-sight to the water's surface. Reflective signals and time-based sensor logic are used to measure the distance between the sensor head and the water's surface. The measurement is converted into the corresponding tide level using calibration parameters that were derived from a geodetic survey of the sensor head heights. The sensor locations were selected to ensure an adequate water depth even at extreme low tides.

The raw sensor data is relayed via an RF link to a Raspberry Pi Computer located in the Harbormaster's office, which in turn forwards the time-tagged information to an online InfluxDB cloud-stored database.

4. Davis Instruments Vantage Pro Weather Station

The Davis Vantage Pro weather station is a research-grade weather station that provides professional-grade data. It reports real-time wind speeds in mph, temperature and humidity, barometric pressure, rainfall, and other information such as heat index and dew point. Weather data obtained from the weather station is uploaded to weatherlink.com to provide online access for the display of current and historical weather.

5. External Interfaces

a. https://www.influxdata.com
   Provides ultrasonic and radar tide level measurements, which are obtained from the sensors located at the Belfast City Dock and stored in the InfluxDB cloud database.
b. https://api.weatherlink.com/v2/current/
   Provides current weather data obtained from the Vantage Pro Weather Station.
c. https://tidesandcurrents.noaa.gov/api/datagetter/
   Provides predicted tide data for Belfast and water temperatures for Portland Harbor and Cutler Wharf.
d. https://api.weather.gov
   Provides weather forecast data for onshore and marine weather conditions.
e. smtp.office365.com
   Email server for issuing alert email messages from the alert@belfast-tide.org account.
f. Twilio.com (via the Twilio Python module)
   Text message server for issuing alert text messages.
g. Cloudflare.com, cloudflared service. Account name cfi1513840@gmail.com, password _____.
   Provides secure internet tunneling services for accessing the tide station. The URL, belfasttide.org, is registered with the cloudflare.com DNS server. The cloudflare.com server redirects internet traffic to a tunneling service which has been configured to interact with the cloudflared client installed on the RPi. This provides seamless access to the Tide Station web services.
h. The Apache2 web server, which is integral with the RPi operating system, provides the external interface required for user internet access to the tide station displays.
i. rclone (see following section, Cloud Storage).

6. Cloud Storage

Microsoft cloud storage at https://onedrive.live.com is used as a repository for the Tide Station source code, and SQLite3 database backups under the account name belfastalert@outlook.com, password _____. The utility rclone is installed on the RPi to provide cloud storage access. The rclone cloud storage specification is onedrive:Belfast, which defines the remote mount point and login credentials. A “crontab” entry under the “tide” account causes the /home/tide/bin/uploadtides.sh script to be executed each day at 00:15. The script, in turn, causes execution of an rclone command which results in all modified files located in the /home/tide/Uploads/ directory to be copied to the onedrive:Belfast cloud storage. A listing of files contained in the cloud can be found by entering the command “rclone lsl onedrive:Belfast”.

7. Remote Network Access

Due to the physical location of the tide station RPi computer, the operation and maintenance of the system and associated software requires administrative access to the computer via the internet. This access is provided by two different methods, each of which is used for slightly different purposes.

The first method is the Secure Shell Protocol (ssh), which is implemented via the ssh.belfasttide.org URL. Internet access to this URL is handled by the cloudflared tunneling server, which redirects requests to the ssh port of the RPi computer. The ssh protocol is used for remote login and command line interface to the RPi, and to provide file transfer capability using the Secure File Transfer Protocol (sftp) (example “sftp tide@ssh.belfastide.org”). The user must supply the RPi tide account password when using the ssh protocol.

The second method uses the Virtual Network Computing (VNC) protocol, to establish a remote desktop environment on the user’s local computer. This provides the user with full access to all the features of the RPi computer, with a desktop GUI identical to a local login session. The VNC service is provided by RealVNC.com using a free non-commercial account subscription. The necessary account and configuration information is managed within the RealVNC network and access to the tide station is through the RealVNC viewer, which is required to be installed on the user’s local machine.

Following installation of the RealVNC Viewer, the user must sign-in to the RealVNC.com tide station account. The sign-in credentials are username belfastalert@outlook.com, password _____.

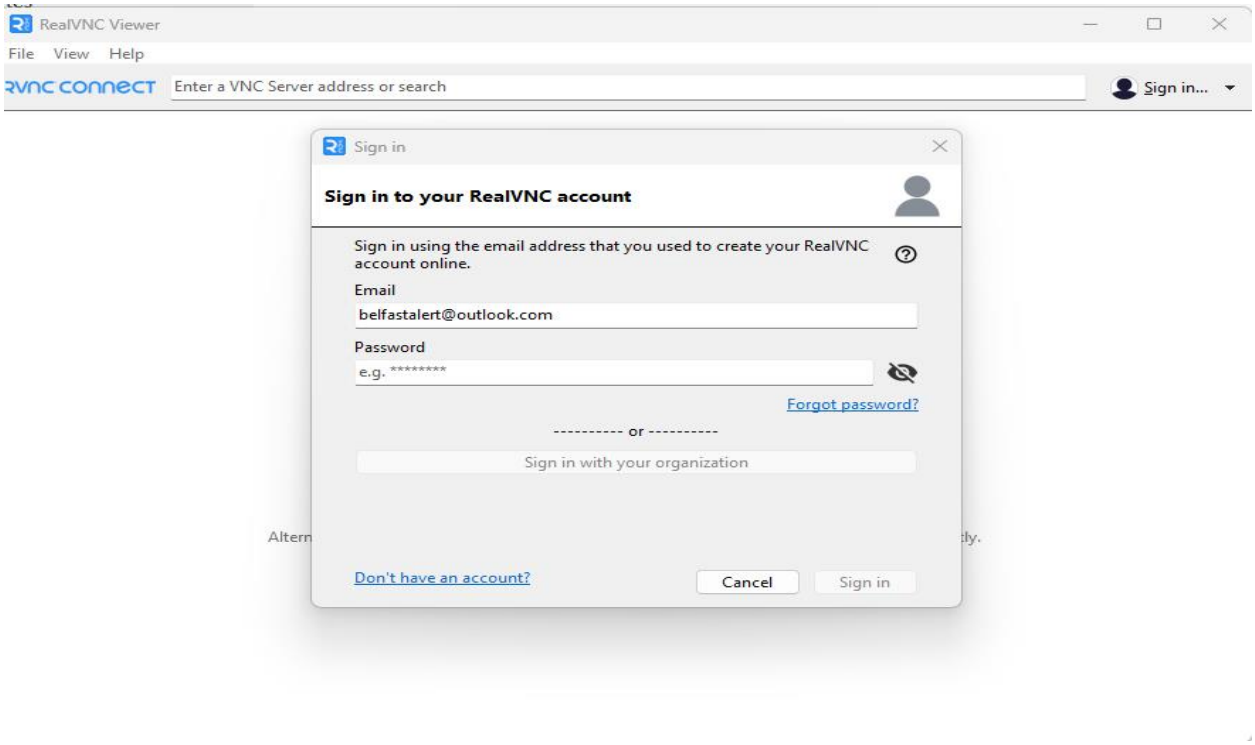

Figure 3. RealVNC Account Sign In Window

Once sign-in to the RealVNC account has been completed, the tide station computer will appear for selection. Initial selection will require the tide station account and password information.

8. SQLite3 Database Structure and Organization

The local, RPi-resident SQLite3 database is used to provide storage for initialization parameters, predicted tides, actual tide measurements, and weather station readings. It is also used to store user-related alert requests. Predicted tide data is updated once per day at 00:00 and extends for a period of 15 days into the future. Actual tide and weather data is stored once per minute and has no specific expiration or purge date. This is to allow historical analysis of previous conditions which may be of interest. The working database is located at /var/www/html/beltides.db.

Database entries are stored according to the following schema. The tables and fields are presented in sequential order as they are defined in the database structure.

Sunrise/sunset table - suntimes

Field Name	Field Type	Description
alertmsg	text	Provisional webpage banner (not used)
sunrise	text	Belfast sunrise time
sunset	text	Belfast sunset time
dispdate	text	Current date

Measured tide table - tides

Field Name	Field Type	Description
dtime	datetime	Date & Time
tide	float (5,2)	Tide Level
trend	text	Tidal trend (rising, falling, low, high)

User Alert table -- useralerts

Field Name	Field Type	Description
dtime	datetime	Date & Time
emailaddr	text	User’s email address
tmsgaddr	text	User’s phone number for text message alerts
tlevel	text	Tide level alert value (feet)
enalev	text	Tide level enable flag
levstat	text	Tide level alert status
daylight	text	Tide level daylight reporting only flag
atemp	text	Temperature alert value (degrees F)
enaair	text	Temperature enable flag
airstat	text	Temperature alert status
activated	text	User alert activation flag
dayair	text	Temperature daylight reporting only flag
wind	text	Wind speed alert value (mph)

enawind	text	Wind speed enable flag
windstat	text	Wind speed alert status
daywind	text	Wind speed daylight reporting only flag
windir	text	Wind Direction specification (N,NE,E,SE,S,SW,W,NW)
vari	text	Tide level variance from predicted tide value
enavari	text	Tide level variance enable flag
varistat	text	Tide level variance alert status
dayvari	text	Tide level daylight reporting only flag
watertemp	text	Water temperature alert value (degrees F)
enawater	text	Water temperature enable flag
waterstat	text	Water temperature alert status
datwater	text	Water temperature daylight reporting only flag

Parameter initialization table - iparams

Field Name	Field Type	Description
ultracal	float(6,3)	Ultrasonic sensor calibration value
radarcal	float(6,3)	Radar sensor calibration value
banflag	text	Provisional banner display flag
banner	text	Provisional website display banner
sleepmode	int	Local display sleep mode (2=auto)
sleepstart	text	Sleep mode start time
sleepend	text	Sleep mode end time

Weather data table - wxdata

Field Name	Field Type	Description
dtime	datetime	Date & Time
temp	int	Air temperature (degrees F)
baro	float(5,2)	Barometric Pressure (inHg)
humid	int	Humidity (%)
wind	int	Wind speed (mph)
winddir	int	Wind direction in degrees relative to magnetic north
windgust	int	Wind gust (mph)
barotrend	float(5,3)	Barometric pressure trend (inHg)
dewpoint	int	Dewpoint (degrees F)
heatindex	int	Heat index (degrees F)
rain60m	float(5,3)	Rain rate (inches)
watertemp	float(4,1)	NOAA station 1 water temperature (degrees F)
watertempx	float(4,1)	NOAA station 2 water temperature (degrees F)
rain24h	float(6,3)	Daily rain amount (inches)

Predicted tide table - predicts

Field Name	Field Type	Description
dtime	datetime	Date & Time
tide	float (5,2)	Tide Level (feet)
state	text	High or Low tide indicator

9. Tide Station Configuration, User Account and Script Starup

The Tide Station RPi computer is configured with the user account “tide”, password _____, from which all tide station operations are conducted. The Python script beltide.py will be started automatically following initial power-up or reboot of the computer. This is accomplished through the RPi’s system service manager, systemctl. The command “man systemctl” may be used to obtain additional information on the use of systemctl. Startup of the beltide.py process is controlled by the file /etc/systemd/system/beltide.service, whose contents are as follows:

```
[Unit]
Description=Tide Station
After=network.target
[Service]
TimeoutSec=5min
ExecStartPre=/bin/sleep 180
Environment=DISPLAY=:0
ExecStart=/usr/bin/python3  /home/tide/bin/beltide.py tk log
WorkingDirectory=/home/tide/bin
StandardOutput=inherit
StandardError=inherit
User=tide
[Install]
WantedBy=multi-user.target
```

If necessary, the script can be started from the command line following login by entering the commands:

```
>>>cd bin
>>>python beltide.py tk log
```

Note that the “tk” parameter is required to activate the local Harbormaster’s display. The “log” parameter will cause all informational and error messages to be written to a log file at /home/tide/log/beltide.log.

Care should be taken to ensure that only one instance of beltide.py is running, as multiple instances will cause unpredictable results. Multiple instances could be present if the script is started manually following a reboot, without first terminating the currently running script. Verification of only one instance can be performed by executing the following command:

```
>>>ps aux | grep beltide
```

10. Task Scheduling

Two tasks are run automatically utilizing the RPi's "cron" task scheduler service, which is driven by the following entries in the "tide" account "crontab" table.

15 0 * * * /home/tide/bin/uploadtides.sh

*/15 * * * * /usr/bin/python /home/tide/bin/belcreatewxhtml.py

The first entry causes the uploadtides.sh shell script to be run every day at 00:15. The script calls the rclone process to upload all modified source code to the cloud storage.

The second entry is run every 15 minutes. It invokes belcreatewxhtml.py to obtain and process the latest forecast weather and create the HTML script file /home/tide/bin/belwx.html. This file is appended to the main display HTML file created by belcreatehtml.py at each one-minute update, and together they comprise the tide and weather display found at the belfasttide.org/beltide.html URL.

11. Tide Station Python and HTML scripts

The following python-language scripts and HTML files are employed to process tide and weather data and provide the local tide display and internet access services. All python scripts are run from the /home/tide/bin directory, with the exception of the CGI scripts, which are copied to the /usr/lib/cgi-bin/ directory for execution by the Apache2 web server. HTML files are copied to the /var/www/html/ directory for access by the Apache2 web server.

a. **beltide.py** – This is the main Python script, normally started at boot time. It is used to collect, store and process sensor and other input data, and to format and display the tide station monitor panel at the Harbormaster's office. Upon initial execution, the script reads calibration parameters and other initialization data from the SQLite3 database. Thereafter, the script is executed once per minute to perform the following functions:
   1. Update the date and time.
   2. Obtain via the internet, the latest InfluxDB values for the ultrasonic or radar tide sensor readings.
   3. Obtain via the weatherlink.com internet interface, the latest data values from the Vantage Pro weather station at the City Dock.
   4. Store time-tagged tide sensor and weather data in the local SQLite3 database file /var/www/html/beltides.db.
   5. Update the Monitor Panel display with the latest weather data and plot the current predicted and actual tide displays.
   6. Read and process the user alert table located in the SQLite3 database in conjunction with the current tide and weather data and issue user alerts if the current conditions meet the alert criteria for each user.
   7. Execute the subprocess belcreatehtml.py, which processes and formats tide and weather data to create the main HTML script for access via the internet URL belfasttide.org/beltide.html.

   In addition to the one-minute processing tasks, the script will perform the following tasks at the specified intervals:

Daily at 00:00:

8. Execute an internet query to obtain the Belfast predicted tides from NOAA for the following 15 days and update the “predicts” table in the SQLite3 database.
9. Calculate and display the Belfast sunrise and sunset times.

Every 15 Minutes:

10. Execute the subprocess belgetpredicts.py, which processes the NOAA tide predictions to produce an HTML file containing a plot of the predicted tides for the following 14 days. The result is accessible via the internet URL belfasttide.org/belpredicts.html.

11. Execute a subprocess to copy the active SQLite3 database to the directory /home/tide/Uploads. The database copy is used to provide data for plotting and analysis, and to prevent interference with the active database access by the beltide.py script.
    The database copy is also used as a source for performing a database backup to the cloud on a daily basis. Refer to the Cloud Storage section for a description of this process.

12. Execute the subprocess belplot.py, which processes analytical tide and weather data to produce an HTML file for display via the URL belfasttide.org/belplot.html. The SQLite3 database copy located at /home/tide/Uploads/beltides.db is used for this purpose.
    Note: the execution of subprocesses that are run on 15-minute intervals are staggered by one minute.
    Note: one advantage of using the subprocess execution method is that the called subprocess must be completed before execution of the main python script will continue. This prevents database access conflicts from occurring between processes.

Hourly:

13. Execute an internet query to obtain the Portland Harbor and Cutler Wharf water temperatures from NOAA and update the appropriate displays and the “wxdata” table in the SQLite3 database.

b. belcreatehtml.py – This process is called from beltide.py once per minute to read and process tide and weather data from the SQLite3 database. The process creates an HTML script which serves as the front end for the tide station internet display. The process will then append the file /home/tide/bin/belwx.html to the script to create the final HTML file, which is copied to /var/ww/html/beltide.html to provide for user internet access.

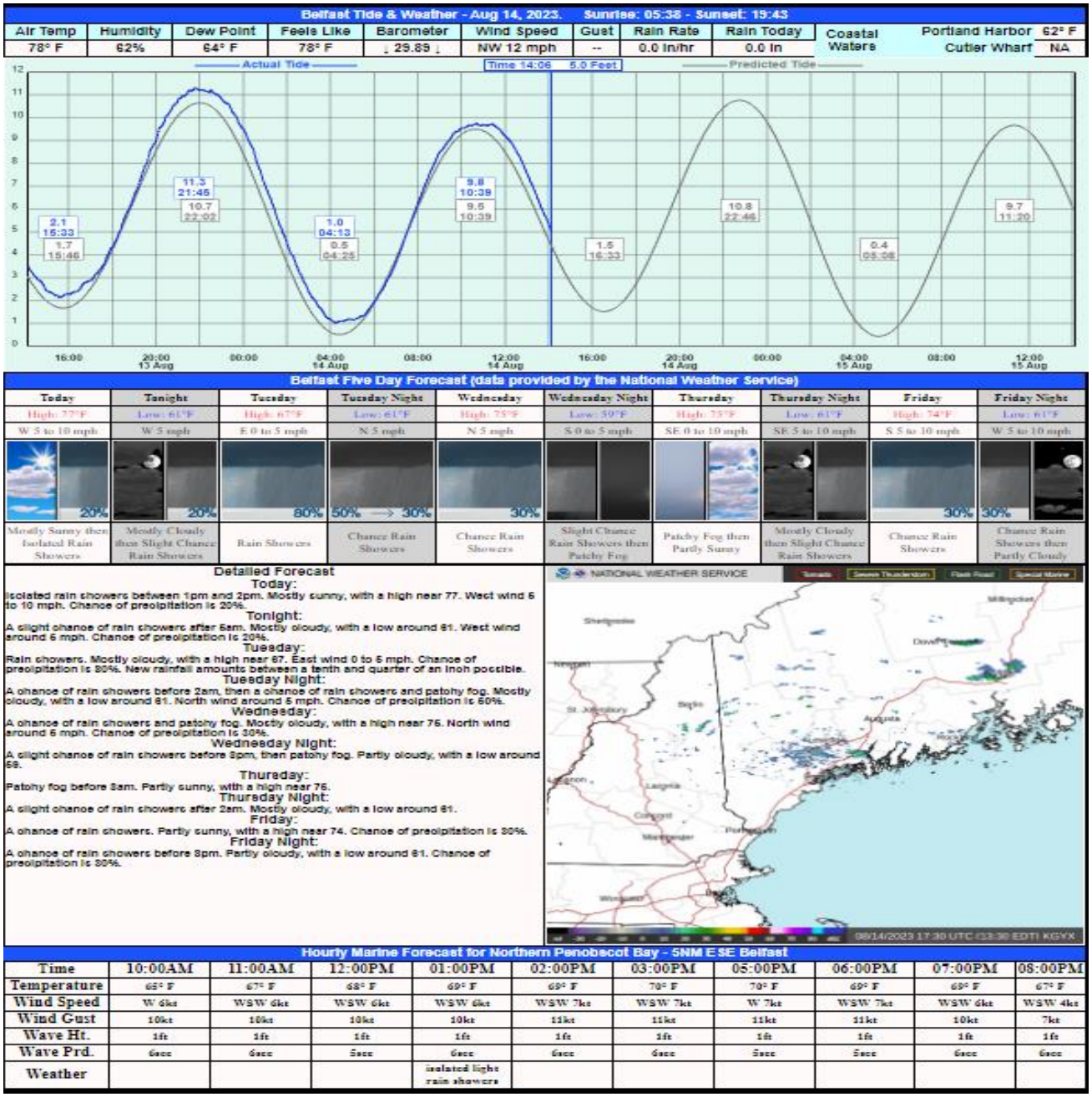

Figure 4 – Tide & Weather Webpage Display

c. **belcreatewxhtml.py –** This process is called from the "cron" task scheduler once every 15 minutes to provide forecast onshore and marine weather conditions for display via the primary tide station webpage found at belfasttide.org/beltide.html. The process queries the NWS online database using parameters which specify the locations for the "point" weather forecasts at Belfast and the Upper Penobscot Bay. The NWS website returns JavaScript Object Notation (JSON) formatted data containing all weather-related parameters associated with the requests. All appropriate weather data is extracted from the JSON response and formatted for display via the creation of an HTML script. The HTML script is stored in /home/tide/bin/belwx.html. It is then appended to the front-end HTML as previously described to create the /var/www/html/beltide.html webpage.

d. **belgetpredicts.py** – This process is called from beltide.py once every 15 minutes to provide the 14-day predicted tide display. The data used to create the display is read from the “predicts” table of the SQLite3 database. The copy of the database located at /home/tide/Uploads/beltides.db is used for this purpose. The process creates an HTML script which is copied to the file /var/www/html/belpredicts.html to provide for user access via the internet.

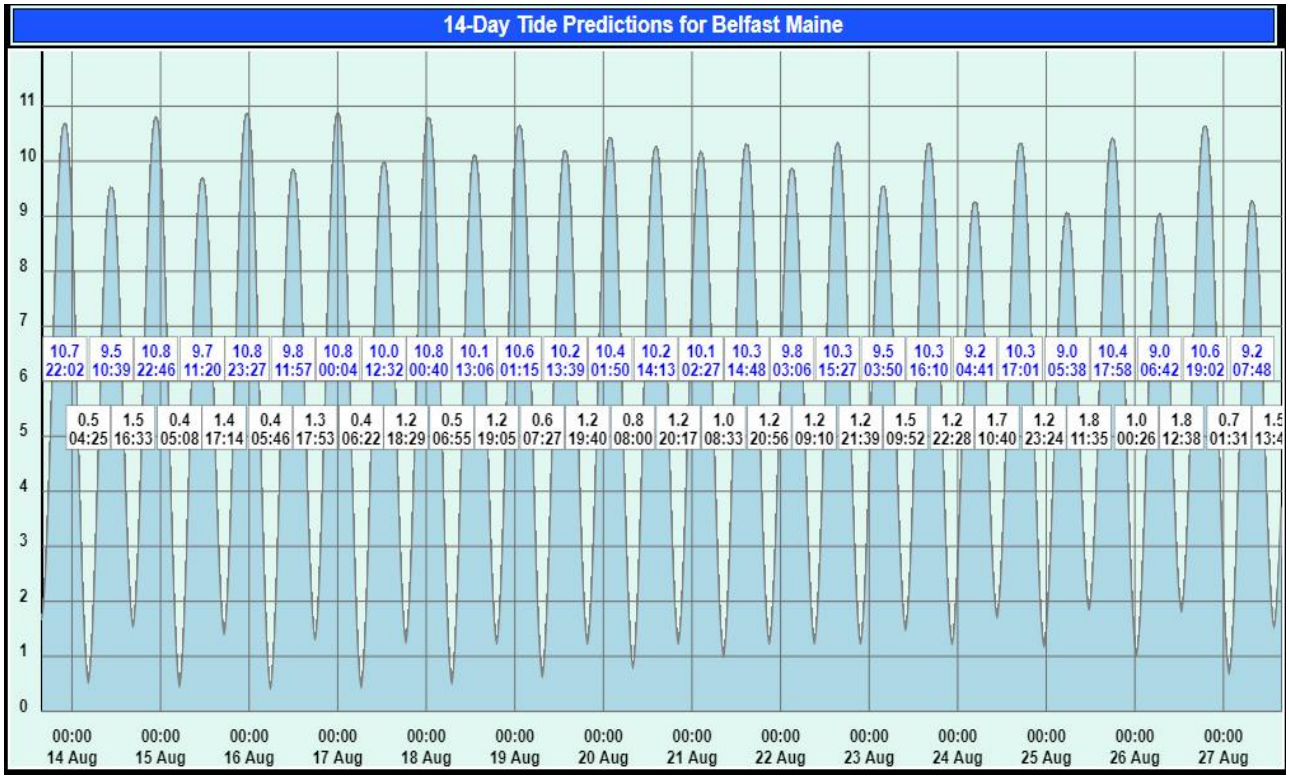

Figure 5 – Tide Prediction Page

e. **belplot.py** – This process is called from beltide.py once every 15 minutes to extract and process analytical tide and weather data to produce an HTML file for display via the URL belfasttide.org/belplot.html. The default date, time and timespan for the preconfigured plot display is the current date with a timespan of 14 days. The default plot parameters are the measured and predicted tides with the associated high and low tide tags, the variance between measured and predicted tides, the wind speed and direction, the amount of daily rainfall, and the temperature. The HTML webpage display also contains parameter selection widgets that can be used to request user-specified formatting and time parameters. If user-specified parameters are requested via the “Refresh” button, the form is resubmitted, causing the execution of the belplot.cgi python script, which will utilize the user-selected parameters instead of the default, as previously described. Thus, execution of belplot.cgi produces a plot that is tailored to the user’s selection criteria. Multiple selections and submittals can be accomplished to view a variety of conditions. Note that there may be some delay involved in the response time depending on the time span requested for the plot. Data discontinuities due to network or other interface errors will be evident as an absence of plotted data, as shown in the example in the following figure.

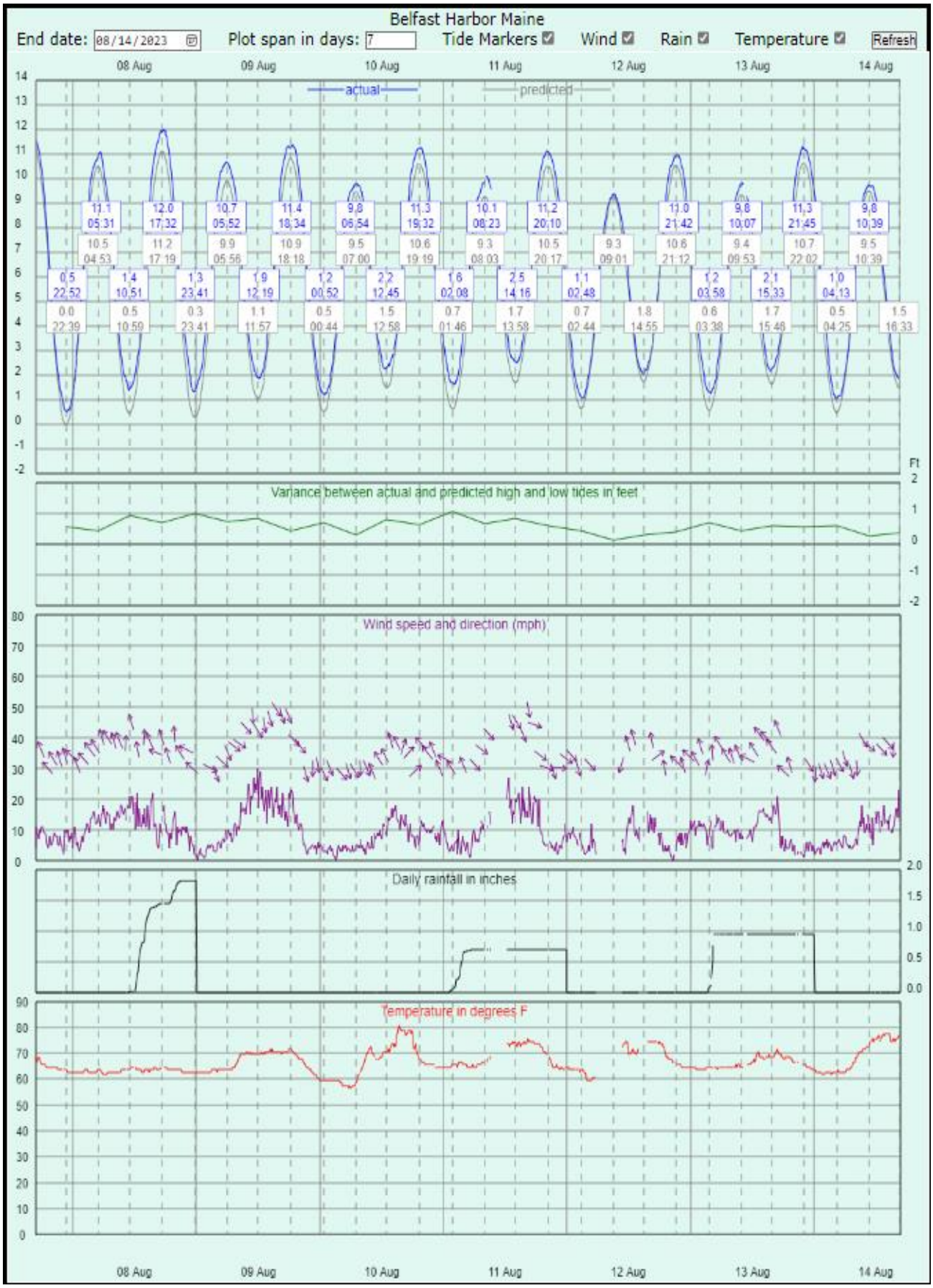

Figure 6 – Plot of Tide & Weather Data by Time Span

f. **belplot.cgi** – This process is essentially identical to belplot.py, except that HTML script is generated on the fly based on the user-specified parameter input.

g. **belgetuser.cgi** – Process user alert requests and update the user alert table in the SQLite3 database. The CGI script is called following submission of the beluserform.html webpage form to validate user alert requests and activate alert processing.

h. belindex.html – This HTML script is copied to /var/www/html/index.html to provide the main webpage for the tide station. There are links on the page to all the other tide station displays and functions.

i. beluserform.html – The user alert request form. The form provides a means by which the user may specify parameter values and conditions for generating email and text message alerts**.** The form also includes a field for providing administrative functions, such as activating or deactivating user requests. Note, the admin values _____, _____, and _____ are used to display or activate/deactivate user requests.

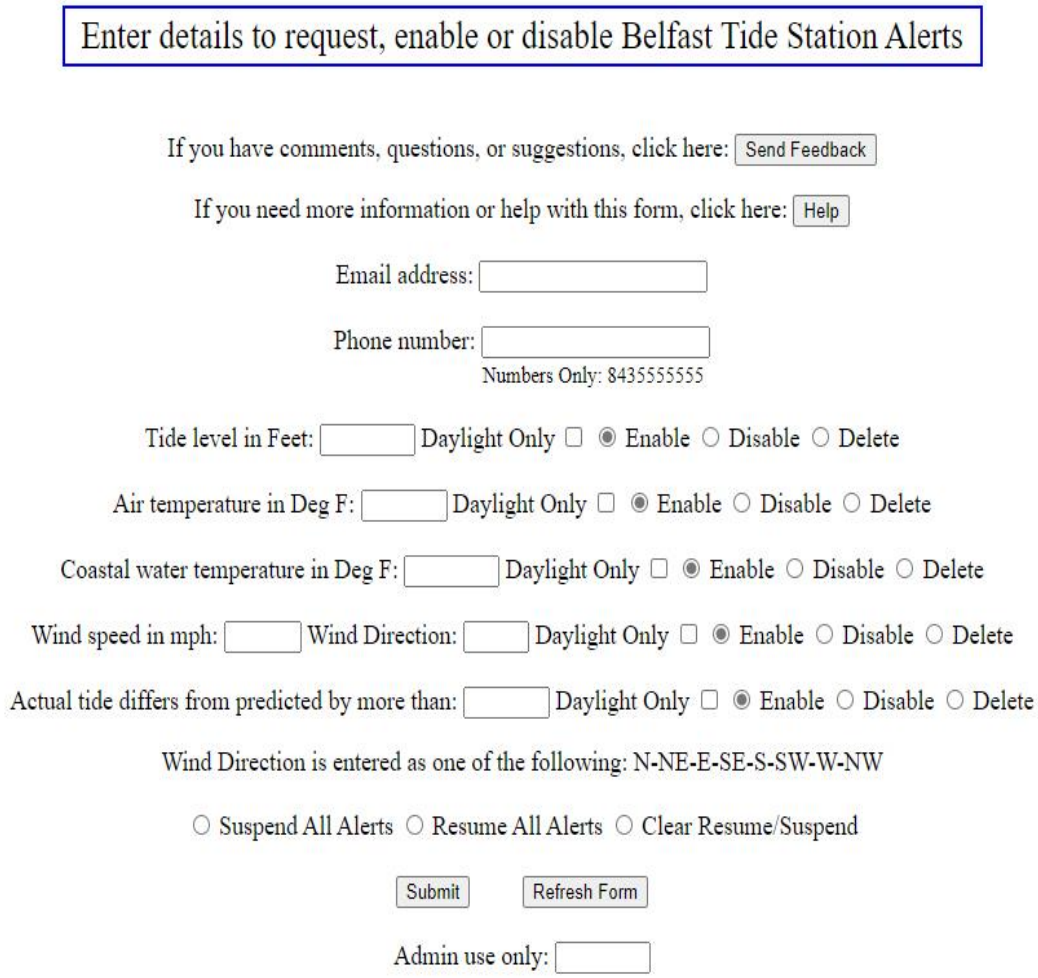

Figure 7 – User Alert Request Form

j. **beluserhelp.html** – This file provides a webpage describing the use of the user alert request form.

12. Source File Configuration Control.

All python scripts and HTML files are identified by a one-up version number scheme using the syntax [filename]vn.n.[type]. Thus, the file beltidev5.1.py would denote version 5.1 of the beltide.py script. Normally, the latest version-numbered file would be copied to the base filename to configure the executable configuration. This allows easy regression to a previous version, if required. All source code resides in the cloud and is copied to the /home/tide/bin directory.
Note: CGI scripts that are created or edited on a Windows computer have an incompatibility with the Unix end-of-line character sequence. In this case, the “dos2unix” utility should be run on the RPi computer to convert the files to the correct format. This should be performed before copying the files to /usr/lib/cgi-bin/.

13. Source File List

The following source files constitute the tide station software library:

beltide.py	Core process.
belcreatehtml.py	Tide & current weather HTML script.
belcreatewxhtml.py	Forecast weather HTML script.
belplot.py	Tide & weather HTML plot script.
belgetpredicts.py	14-day predicted tides.
belplot.cgi	Tide & weather HTML plot script (via web request).
belgetuser.cgi	Process user alert requests.
belindex.html	Tide Station Home page (copied to /var/www/html/index.html.
beluserform.html	Webpage form to submit user alert requests.
beluserhelp.html	Webpage help file.
uploadtides.sh	Shell script that executes rclone to copy files to cloud storage.