

# Codebook for Ice beacon speed calculation script

By Victory Iyakoregha

November 2nd 2019

# Contents

<b>1</b>	<b>ABOUT</b>	<b>1</b>
1.1	Description . . . . .	1
<b>2</b>	<b>SETUP</b>	<b>1</b>
2.1	Script-Specific options or packages required . . . . .	1
<b>3</b>	<b>SCRIPT DETAILS</b>	<b>2</b>
3.1	Speed_calculator_settings.ini . . . . .	2
3.2	Speed_calculator.py . . . . .	3
3.2.1	Perform Calculations method . . . . .	4
3.2.2	Calculate distance method . . . . .	4
3.2.3	Calculate speed method . . . . .	4
3.2.4	Main . . . . .	5
<b>4</b>	<b>Log</b>	<b>6</b>

# 1 ABOUT

## 1.1 Description

This codebook describes the contents, structure, and use of the speed\_calculator.py script. Speed\_calculator.py is python script written to calculate the speed between any two points in all Ice Beacon data files.

Instrument:	Iridium Ice tracking beacons, or ice drift beacons deployed in the Canadian Arctic (ex. Oceanetic (1989) model 703)
Data Format:	Input files are in Comma separated value, XLSX, or XLS format Output files are produced depending on the format of the input files.
Project(s):	Ice beacon data standardization

# 2 SETUP

## 2.1 Script-Specific options or packages required

Visit <https://docs.python.org/3/library/> to learn more about these packages.

1. Os -  
The Operating system(os) package is a package used to provide a means to access certain operating system specific functionality. Using the Os package, we can define certain operating system specific behaviour for different operating systems.
2. Configparser -  
Configparser module provides the functionality to write python scripts which can be customized by end users easily. In this case, Speed\_calculator.ini.
3. Pandas -  
Pandas is a module used to extract data from input files, and also for data analysis and manipulation.
4. Pathlib2 -  
This is the module used to access file paths on each users computer.
5. Math -  
Math is the module used to gain access to the mathematical functions and equations.
6. Geopy -  
Geopy is a module used in geolocation calculations given latitude and longitude coordinates.

## 3 SCRIPT DETAILS

Script name(s): Speed\_calculator.py, Speed\_calculator\_settings.ini

### 3.1 Speed\_calculator\_settings.ini

Speed\_calculator\_settings.ini is an initializer file. This file allows end-users to easily customise the script without making any changes directly to the scriptSpeed\_calculator.py. Users can specify the following:

1. Input data directory  
This is a variable for users to provide the directory path for the input files to be processed.
2. Project  
This is a variable for users to provide a project name for the current bash of files being processed.  
  
NOTE: The script uses information from each file to be able to calculate the distance and speed. The following variables allow the user specify the names that have been given the columns that contain the longitude, latitude, altitude, and timestamp.
3. Longitude dec deg  
This is a variable for users to provide the column name for the column that contains the longitude.
4. Latitude dec deg  
This is a variable for users to provide the column name for the column that contains the latitude.
5. Altitude  
This is a variable for users to provide the column name for the column that contains the altitude.
6. Timestamp seconds  
This is a variable for users to provide the column name for the column that contains the timestamp.

Here is an example of the file with the directory path, project name, and variable names all filled in.

```
[INPUT]
INPUT_DATA_DIRECTORY = C:/Users/Desktop/Ice Beacon folder
PROJECT = Ice Beacon
```

```
[INFO]
LONGITUDE_DEC_DEG = lon_decdeg
LATITUDE_DEC_DEG = lat_decdeg
ALTITUDE = alt
TIMESTAMP_SECONDS = time_utc
```

### 3.2 Speed\_calculator.py

Speed\_calculator.py is the main script that calculates the speed for all Ice Beacon files. This script only accepts input files with CSV or XLSX extensions. Files that satisfy this requirement are then passed along to methods set up to calculate the speed between each point in the file.

Each file records the timestamp, latitude, and longitude. Speed cannot be directly calculated from the data recorded. So first the script calculates the distance between each point using the latitude and longitude. The distance is calculated using a Python module called Geopy. Using the geodesic distance function in Geopy the distance between two points given the latitude and longitude of each point can be calculated.

Here is an example of how the scripts calculates the distance between two points.

Date	Time_UTC	ID	Lat_DecDeg	Lon_DecDeg
2011-01-09	0:00:00	462170	74.6463	-127.786
2011-01-09	1:00:00	462170	74.645	-127.784

Initial distance = (74.6463, -127.786)

Final distance = (74.645, -127.784)

GeopyDistance = Distance (Final distance, Initial distance)

NOTE: Distance, as used in the line above, is a function of the python module Geopy.

Finally, the speed is calculated using the formula below

Speed = (final distance – initial distance) meters / (final time – initial time) seconds.

Speed\_calculator.py uses four methods to carry out calculations and processing.

### **3.2.1 Perform Calculations method**

This method accepts a single parameter. This parameter is the path to the file currently being processed by the script. As a failsafe, this method checks to ensure the file being processed is of the required extension type. All header names are then converted to lowercase to ensure uniformity of names while performing the rest of the processing. This method also calls both the calculate speed and calculate distance methods.

### **3.2.2 Calculate distance method**

This method accepts a single parameter. This parameter is a CSV or XLXS file opened in the perform calculations method. This method uses a function from the Geopy module to calculate distance. This method calculates the distance between every two points in an Ice Beacon file. The initial distance covered is always zero; Consequently, the distance of the first point in each file is zero. Finally, a new column is created to record the calculated distance.

### **3.2.3 Calculate speed method**

This method accepts a single parameter. This parameter is a CSV or XLXS file opened in the performance calculations method. This method is called by the performance calculation method after the calculate distance method is done running. Speed is calculated using the formula:

Speed = (final distance – initial distance) meters / (final time – initial time) seconds.

As with the distance calculations, the initial speed travelled is zero; Consequently, the speed of the first point in each file is zero. Finally, a new column is created to record the calculated speeds.

### **3.2.4 Main**

The main function is necessary for executing any script. In this script, the main function loops through the input file directory using the Pathlib2 function "path". The "path" function loads all files in the directory specified by the user into a list. The main function then explores this list for files with the CSV or XLSX extensions. For each file found the perform calculations method is called and that file is processed. This process is repeated for all CSV or XLSX files in the selected directory.

## 4 Log

1. For more information on python visit <https://https://www.python.org/>
2. For more information on Geopy visit <https://pypi.org/project/geopy/>