eBirdSoft Documentation

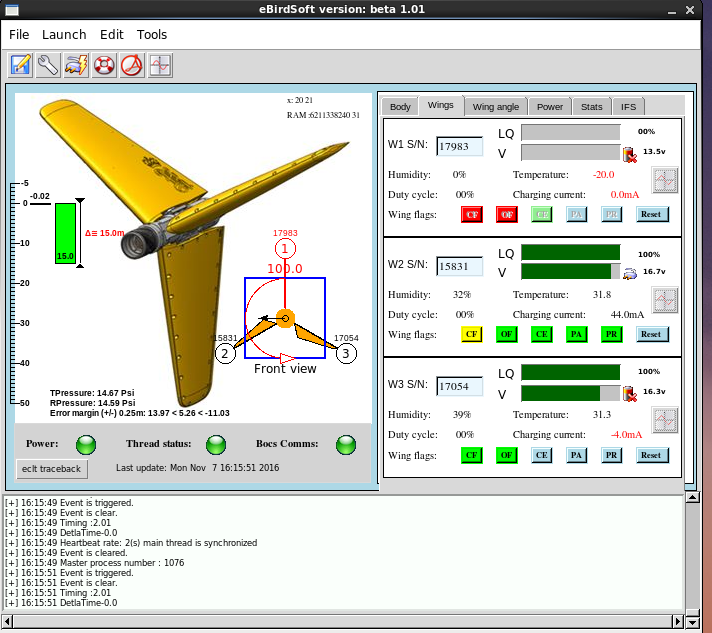


Table of contents

* How to start

1. Gnome terminal
2. The pre-launch checks

* Main display

1. eBird body Depth indicator
2. Depth sensor margin
3. Rotation display
4. The body TAB
5. eBirdSoft status
6. eclt trace back

* The Wing TAB

1. Wing angle
2. Wing test
3. Wing Performance

* The power TAB
* The stats TAB
* IFS stats TAB
* eBirdSoft database
* Contribution
* Project Presentation

**Actual version**: beta 1.01

First version is in used on ATS and consists of a testing/reporting platform for eBird bodies and wings.

It is fully operational however some of the options in the main menu and icons are not yet implemented.

The reporting PDF function and IFS platform are also under construction.

This feature will be implemented in the next version (if the project gets approval from the office).

eBirdSoft exists in two different modes, first, a testing mode (emulating eclt command response) and secondly a production mode.

**I will explain the production mode in this documentation.**

|  |
| --- |
| - How to start eBirdSoft |
|  |

Double click on the Desktop shortcut 

Three windows pop up on the screen.

1. First, the ‘**gnome terminal’** is launched and shows all the background commands and critical information generated by eBirdSoft (error messages, warnings and threads executions etc.).

If you close the gnome terminal, eBirdSoft will exit without saving any data into the database.

This is not recommended as you can quit the main environment via the contextual menu (option exit).

However, closing eBirdSoft via the gnome terminal will not cause any harm to the installation and will not alter or corrupt the database or any other essential scripts.

1. **The ‘Pre-launch check’** window will appear at the side of the main window.

This regroups information concerning the machine and running of the program.

Before starting the main display, the script performs a list of essential checks to make sure that the platform is suitable for running the main program.

The checks include:

* **Basic** **memory check**

eBirdSoft requires a lots of RAM to run due to is conceptual in-memory database running in the background. I will explain the concept later in the document and the benefits vs traditional database.

* **Screen dimension checks.**

Check if the current screen dimension is suitable for eBirdSoft.

* **Checking operating system.**

eBirdSoft is designed to run on various platforms including Linux, window and Mac OS. Tested proficiently under Linux

* **Python version**

Python 2.6 has been used for coding the core of eBirdSoft.

* **Tcl/Tk version**

Version 8.6 is thread-enabled operations (multi-threaded programming tasks).

A warning will be issued if the version does not meet this requirement.

Below version 8.6 mtKinter (thread-safe version of Tkinter ) will be used.

The GUI will not start if any of the above checks are not satisfied or if the check process is aborted.

|  |
| --- |
| - Main display figure 1 |
|  |

The GUI is regrouping every essential piece of information on the same page, sub divided into different categories.

On the left side of the main window, is a depth reporting system that gives useful information about the eBird body depth status.

The scale goes from ‘-5m to 50m’ .

‘-5m to 0m’ represent depths above the sea level and ‘0m to 50m’ are the depths below the sea level.

Figure 2 shows the eBird body depth to be reading -0.05m (at ambiant atmospheric pressure).

**The current depth** is represented with a **red bar (**when delta>0.25m,) and green when delta <0.25

**Target depth** is represented with a green bar( see figure 2 below).

|  |
| --- |
| eBird body depth indicator, figure 2 |
| The difference between the target depth and the current depth is displayed via the delta variable ( 10.3m in the example)  This option is very useful when trying to check the depth calibration.  The red bar dimension is proportional to the pressure applied to the depth sensor. |

|  |
| --- |
| Depth sensor error margin, figure 3 |
|  |

**Depth sensor error margin (figure 3)**

**Tpressure** : is the theoritical pressure (in psi) at a given depth

**Rpressure** in psi : is the real pressure (in psi) ;pressure measured via the sensor

**Error margin** (+/- ) 0.25m is reel time algorithm showing

the range of linear coeficients acceptable in beetween a specific tolerance of +/- 0.25m.

If the linear coeficient at a given depth is fluctuating above +/-0.25m, the error margin will turn to red for a brief time, flagging an issue.

Typically you will observe such direptancies when pumping the air with the druck PLV into the ebird body sensor.

The abrupt change in the air pressure will be flagged by the algorithm as a non-linear value and be reported.

Testing the linearity is essential to determine if the sensor is reading accurately or if any salt crystals or slime are altering the eBird body ‘s sensor caracteristics.

To test the coeficient liniearity, close the Druck PLV 620 and let the air drop down slowly while monitoring the ‘error margin’ values and color . The air should drop slowly and error margin values should remains in specs.

|  |
| --- |
| - Rotations display figure 4 |
| This graphic representation an eBird body ( front view).  This works the same that BOCS display. Serial numbers will appear in the 3D display when detected.  Torque / lateral and vertical force are represented the same way than BOCS  A wing not detected or not present will be represented in red |

|  |
| --- |
| - eBird body TAB figure 5 |
| The program will update at regular intervals ( scan interval ) the following values reported by the eBird Body.   * serial number * firmware version * Rotation * Roll ( in degrees) * Humidity (%) * Pressure (in hpa) * Temperature (in degree Celcius) * Depth reading ( in meters) * Assigned depth ( in meters)   ‘None’ value will be used if the data has not been collected by the eBird body.    Uploading the Firmware version has not yet been implemented. |

 Values above specs will be flagged in red and reported by a warning sign

 click on this icons to modify :

* The rotation count or rotate the eBird Body (first window from the left)
* Change the target depth (middle window)
* Recalibrate the eBird Body depth (right window)

|  |
| --- |
| Figure 6 |
|  |

* Restart bird button : will restart the ebird body ( the process take up to 30 seconds).
* Sync button: will sync all the eBird wings ( this also takes 30 seconds)
* “Last update” is showing when the last scan has been performed

|  |
| --- |
| eBirdSoft Status figure 7 |
| This Frame is giving the status of   * Power supply * Threads activity * Bocs communication   “Last update” is showing when the last scan has been performed |

**Power indicator**

 Green when at least one line is ON (367 Volts)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Indicator status** | **L1** | **L2** | **Voltage L1** | **Voltage L2** |
|  | ON | ON | 367 | 367 |
|  | ON | OFF | 367 | 0 |
|  | OFF | ON | 0 | 367 |
|  | ON | DIAGNOSTIC | 367 | 70 |
|  | DIAGNOSTIC | ON | 70 | 367 |

 Yellow when one or more lines are in DIAGNOSTIC and none in full power

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Indicator status** | **L1** | **L2** | **Voltage L1** | **Voltage L2** |
|  | DIAGNOSTIC | DIAGNOSTIC | 70 | 70 |
|  | DIAGNOSTIC | OFF | 70 | 0 |
|  | OFF | DIAGNOSTIC | 0 | 70 |

 Gray when L1 **and** L2 are off (no power)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Indicator status** | **L1** | **L2** | **Voltage L1** | **Voltage L2** |
|  | OFF | OFF | 0 | 0 |

**Thread status**

|  |  |  |
| --- | --- | --- |
| **Indicator** | **Thread status** | **Explanation** |
|  | Working | eBirdSoft is checking vital ebird body function at regular interval. When the light is flashing red, a scan has been generated and data are refreshed. |
|  | Paused | Program is stopped, no scans ( data are not refreshed) |
|  | Solid red, program has stopped working | The program has stopped working.Quit and start the program again. |

BOCS comms

|  |  |  |
| --- | --- | --- |
| **Indicator** | **Thread status** | **Explanation** |
|  | Active | Good comms, fast reponse time. |
|  | Slow response | Slow response. Still working but will eventually drops data if persistant (Response time > 30ms). |
|  | Solid red, Inactive | Inactive, eBirdSoft cannot fetch data from the eBird body or wing(s) during a period exceeding 1s. Possibly due to consecutives timeout (busy interface) |

**ECLT Traceback**

This frame is showing the master thread activities in real time.

When a scan is performed, an event is sent ( also known as process number) to all threads to synchronize and process the data.

An event is occuring at regular interval ( 6s in figure 8).

In a future version, the user will be able to change the interval between events, but for now this variable is set to 6 seconds.

eBirdSoft controls the timing and will display a warning message if the delay between events is offset and threads not synchronized ( late threads).

|  |
| --- |
| Figure 8 |
|  |

|  |
| --- |
| - Wings TAB figure 9 |
| This window regroups all the essential information concerning the wings connected to the eBird body.  The data below is refreshed after every event (scans)   * Serial number * Humidity value in % * Duty cycle value in % * Temperature ( in degree Celcius) * Charging in mA * Wing flags * Link quality * Voltage   The TAB is divided in three rows (one for each wing)  When a value is abnormal or consider out of specs, it will be display in red. |

Please note output current below 0mA will be displayed in red to be able to spot more easily the wing currently charging.

* When a wing is charging, the symbol  will be displayed on the right hand side of the voltage bar and the symbol  will be used for wings not charging.
* When the internal wing temperature has reached a critical point a warning symbol will be display 

* The right hand side of the temperature value is to alert the operator of a possible danger and take necessary precaution.

**Voltage**

|  |  |  |
| --- | --- | --- |
| **Colour** | **Voltages** | **Status** |
|  | From 16.2v to 17v or above | Go to go |
|  | From 15.8v, 16.2v not included | Voltage a bit low, operator to decide |
|  | From 15v to 15.8v not included | Orange need to be put on charge |
|  | Anything below 13.5 to 15 not included | Cannot be used in production |

**Link Quality**

|  |  |  |
| --- | --- | --- |
| **Colour** | **Link Quality** | **Status** |
|  | >80% to 100% | Considered good \* |
|  | >60% to 80% not included | Need investigation (operator to decide) |
|  | 60% or below | Need investigation but likely to be bad |

\* A link quality of 80% or above is considered good when stable.

General of tests can be done to make sure the LQ is not fluctuating or dropping sharply while under test.

1. First remove any radio devices from the close proximity of the eBird body as certain range of frequencies may interfere with the pairing process.
2. Move the wing in a different port to see if the Link Quality remain above 80%
3. Move the wing to +15, -15 to see if the LQ stays above 80%.
4. Observe the LQ when the wing is receiving current charging
5. Physically rattle the wing to make sure the base plate is firmly connected to the port.
6. Check wing historical work order if necessary.

If the LQ remains above 80% after completion of these tests, then LQ is good to go.

For LQ above 60% but below 80%, the operator needs to swap the wing to a different port and see if the value improves.

If the value is stable above 77% after taking the steps 1 – 5 this should be considered a good spare. (It is assumed that based on the operators experience that they should be able to determine if the wing passes). The operator can also ask a supervisor if any doubt.

Investigate in IFS if the wing has already been removed (historical work order) for the same issue (low LQ).

LQ below 60% is always problematic.

 If a wing does not communicate with an eBird body try rebooting the eBird body.

Check a spare wing on the same port to make sure the eBird body is working properly.

**Understanding the wing flags**

CF : Communication failure

OF : Operation failure

CE : Charge enabled

PA : Power activated

PR : Power received

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Colour** | **CF** | **OF** | **CE** | **PA** | **PR** |
|  | Loosing comms | Operation failure | N/A | N/A | N/A |
|  | Had lost comms | failed at least once(1) | N/A | N/A | N/A |
|  | Never lost comms | Never had issue | Enable | Enable | Enable |
|  | N/A | N/A | Has been enabled | Has been enabled | Has been enabled |
|  | Unknown status | Unknown status | Has not been enabled | Has not been enabled | Has not been enabled |

(1) Means that the wing is working but had issue.

|  |  |
| --- | --- |
|  | **Typical when a wing is not connected** |
|  | Wing connected and receiving current |
|  | Wing loosing comms. |
|  | Wing had lost comms, light green (CE, PA,PR) is proving that the wing has been charging |

**Reset button**

 The reset needs to be used every time a new wing is tested.

Resetting does the following actions:

1. Clear data from the flag matrix
2. Clear data from the plots

**Plotting Voltage and Current**

 Click on the icon to trigger voltage and current plot for a specific wing (the serial number will be displayed on the top of the window)

The plots are monitoring voltage and current data for 30 scans (3.5 minutes of continuous monitored data, then roll the data (FIFO) to have the latest data on the right.

|  |  |
| --- | --- |
| **Wing voltage plot** | **Wing current plot** |
|  |  |
| **The plot is divided into 2 separate axis.**  **The first axis (top of the graph) represents the voltage as numeric values as a function of the time or scans (a scan by default is equal to 6 seconds).**   * **The blue line represents the voltage variations** * **The green line is the average voltage**   **The second axis is the voltage matrix.**  **A voltage level corresponds to a colour gradient.**  **The colour scale goes from 13.5v to 16v+.**  **The matrix is indexed to the voltage values ( not the average)**  **Below the first axis, the legend is displaying the current voltage, the average voltage value, including Maxima and minima but also Δ.**  **Δ value is very useful to flag wings with voltage spikes**  **Dark Green, green colours : wing with sufficient voltage**  **Light green, pale green or cream : need to be put on charge**  **Red gradient : votlage too low** | Current plot is also divided into two separate axis.  The top one is representing the current draw by the wing in function of time.  The scale goes from -100 mA to 100mA ( Y axis is autoscale and will change if values goes over +/- 100mA.  - The blue line represents the current being used  - The red line represesents the average current being drawn  The legend displays the current being used but also the average values and the current consumption in mAh (milli Ampere hour)  The second axis represents the matrix of current values.  Green colour : neutral, 0 mA  Red gradiant : The wing is charging  Blue gradiant : The wing is depleating ( current being drawn)  The matrix is indexed on the current draw value (not onto the average value). |

You can close and restart the plots anytime.

|  |
| --- |
| -Wing Angle TAB Figure 10 |
| This window is used for controlling the wing rotation  Slide the widget to the right or left and select the desired angle.  The wing will move to the selected angle after pressing on the play button.  The yellow triangle (wing angle) will move slowly to the assigned position. The graphic will be refreshed every 6s or after every scans.  The wing should stop right before entering the grey area.  A wing going into the red area has a motor issue.  A wing test can also be performed via the statistic button  When a wing is not connected to the port, all the menu for that specific wing will be frozen |

Wing test

A selected wing will perform a given series of rotations while the program is collecting data in order to represent graphically the wing characteristics.

The graph is divided into two distinct areas (see figure 11)

1. A plot (wing signature)
2. Overall performances and wing statistics

The Y axis goes from -15 to +15 degrees and corresponds to the range of angles the wing will oscillate into.

The X axis represents the time (0s start of the test, 57.3s the final time)

* The red dotted line correspond to the wing signature (data interpolated)
* The green line represents the perfect model (movement or mechanism without friction).

The model frequency is always matching the frequency of the wing (wing period).

If the wing signature is in phase with the theoretical model then we can assume that the motor module is moving very smoothly and the wings overall performance is rather good as long as the timing remains in specs, (test

average is around 60 seconds here on ATS pool)

|  |
| --- |
| Figure 11 |
|  |

If the red dots are plotted after the green line then the wing is showing a bit of delay compared to the mathematical model (ex: around 10 to 18 seconds, see figure 11)

We can see a slight deviation from the response time at 10s when the angle as reached -5 degrees

The same issue is occurring during the next cycles around 34s and then 55 seconds.

These imperfections are giving a specific signature to the wing s/n W18345.

If the red dots are plotted before the mathematical model, then the wing response is slightly faster and this could be symptomatic of another issue illustrated in the next wing signature s/n 21091 (figure 12)

|  |
| --- |
| Figure 12 |
|  |

|  |
| --- |
| Performances |
|  |

The overall performance is a good indication of the eBird wing motor module condition.

A very fast motor module will increase the streamer stability, depth keeping and roll keeping functions.

|  |  |  |
| --- | --- | --- |
| Best Performance (s) | Result average (s) | Worst performance(s) |
| 70.7 | 73.76 | 76.5 |

eBirdSoft is collecting the data of all wings tested during a specific session and then transfers the tests to the local database when the user is exciting the main GUI.

The test results will be compared to each other to calculate and display the best, average and worst results (as shown above). So it is important to do more than one test for each wing to compare the timing between the tests.

The speed is displayed in degrees per seconds, for example:

## From 0 to -15 the wing performances have been evaluated to 2.67 degrees per second. The time needed for the wing to go from 0 to -15 is equal to 15/2.67 = 5.6 seconds

## From -15 to 15 : 30/1.89 = 15.8 seconds and so on

 To start a new wing test, you need to close the wing signature window first.

|  |
| --- |
| - Power TAB figure 13 |
| Streamer power control status of L1 and L2 are accessible in this TAB  Below the power switches, you can observe the numerical values of voltages, currents and Leakage.  L1 and L2 scales are not activated for controlling the eBird power.  This option will come in a different version.  For the time being they are used to show the power states on L1 and L2. The colour convention is identical at the section “power indicator” explained previously.   |  |  | | --- | --- | | Voltage for L1,L2 in (volts) | V1,V2 | | Current supplied (L1,L2) in (Amperes) | I1s,I2s | | Current returned (L1,L2), in (Amperes) | I1r,I2r | | Leakage line L1 in (Amperes) | I1s-I1r | | Leakage line L2 in (Amperes) | I2s-I2r |  * When a leakage is detected a warning sign will be displayed in the leakage section to alert the operator (Leakage given by current supplied – current returned)   The leakage current is displayed in mA. The graph “L1 and L2 leakage” will show the leakage historic (refreshing the display every scans).  Minor variations are common; the leakage detection has been implemented for a variation above 1mA |

|  |
| --- |
| - Stats TAB figure 14 |
| The stats have been developed to track graphically all eBird wings and body internal sensors and functions.  It becomes easy to compare the duty cycle against the voltage or against any other functions.  Users can choose one of the following tab:   |  |  | | --- | --- | | WING TABS | BODY TABS | | And slide across the following functions for wing monitoring:   * Link Quality * Voltages * Temperature * Humidity * Duty cycle * Current * MSG Loss (not implemented yet) | * **Rotation and Roll** * **Roll polar graph** * **Humidity and temperature** * **Pressure and depth** | |

Below an example of eBird body Pressure and Depth monitoring and Roll 3D graph

|  |  |
| --- | --- |
| Figure 15 |  |
|  | **Pressure graph**  - Pressure applied (blue line)  - Average pressure (red line)  - Target pressure (corresponding to the target depth)  **Depth monitoring.**  - Current depth ( blue line)  - Average depth (red line)  - Target depth (yellow line)  The **delta monitoring** graph represents the  value = Current depth – Assigned depth  During a test, delta should be close to zero and should not go over +/- 0.25m when the pressure applied is equal the target pressure  **Delta matrice:**  Here another gadget to represent the delta value colour coded this time. |

|  |
| --- |
| Figure 16 |
| Roll 3D is used to graph the eBird body roll sensor but also the rotation count ( red line on the background)  When the body is moving around his axis, the roll values will takes the following ranges of values (0 to +179) and (0 to -179).  When the eBird body has completed a full rotation around his axis, 1 rotation will be added to the counter and display with the red line.  This graph is very useful to check the rotation counter against the roll sensors and determine if the rotation are real.  This example is from an eBird tested on a fix stand (not rotational) |

|  |  |
| --- | --- |
|  | Roll sensor graph of an eBird body rotating **anti-clockwise (**front view)  6 rotations |
|  | Roll sensor graph of an eBird body rotating **clockwise** (front view)  6 rotations |

|  |
| --- |
| IFS TAB figure 17 |
|  |

This TAB is not yet fully implemented.

I believe that this will be coded on the next version if agreed with the office.

The IFS section will provide crucial information concerning the equipment undertaking the test:

* Has this specific piece of equipment has already a fault report? ( to avoid duplicate)
* Has this piece of equipment already had an external repair order?
* The equipment historical will be displayed to identify trends or past issues.

In a future version I will implement an automatic way to create a fault report or external repair order to make life easier and transfer the PDF report into IFS.

|  |
| --- |
| - Database |
|  |

When exiting the main GUI eBirdSoft will convert all the data recorded during the whole session from an in-memory database into a SQL database.

The process is described below (see figure 4)

1. Access to the local SQL database “eBirdSoft.sq3”
2. Exporting the data to disk via a dump file (conversion from in-memory into SQL format)
3. Migrate the data directly from the in-memory database to “eBirdSoft.sq3” (local DB)
4. Closing all the remaining applications

|  |
| --- |
| - Contribution or Collaboration |
|  |

If anybody is interested to contributing or has any questions regarding the project, please contact me at the following address:

yoyoberenguer@gmail.com

Yoann Berenguer

Observer ATS

|  |
| --- |
|  |
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. |

eBirdSoft Project Presentation

What is eBirdSoft?

eBirdsoft is a project coded in Python (version 2.7, with Tkinter) providing a real-time GUI interface in order to test seismic equipment such as eBird body and wings technology.

It is designed to work onto various platforms such as Linux RedHat for backward compatibility with the BOCS system but will work on most platforms including Mac OS and windows (tested proficiently on Linux and windows).

eBirdSoft is using multithreaded/multiprocessing applications to utilise the full potential of Konsberg Seatex ECLT commands. It will perform background tasks and regroup results in a concise manner.

The results are displayed into tables to simplify the way the equipment functions are represented

This application comes with powerful graphic tools to analyse equipment behaviours but also to store data in a local SQLITE database. Plots and graph can be generated to more easily understand the overall mechanisms such as current charging, voltage variations, eBird body current leakage etc.

|  |
| --- |
| Below are snapshots of eBirdSoft graphic interfaces |
| 30 |

|  |
| --- |
|  |
| snapshot23 |

Actual version of eBirdSoft and future improvements:

A final version will eventually arrive in the first quarter of 2017, compiled into an RPM package

New features/improvements could be added to a future version for example:

- IFS reporting - Historical repair order for a specific equipment,or failure rate statistics.

Auto creation of repair order with related issues and performance monitoring.

- PDF reporting system, - Certifying that the equipment complies with PGS standards (benchmark tests).

- Real-time - Online analysis tools to clearly identify issues on streamers using matrix colour coded information.

- Offline database analysis - To extract pool statistics and graphically represent issues on certain types of equipment.

What can eBirdSoft provide in addition to BOCS ?:

It provides a testing platform with a very concise and intuitive way to monitor and test eBird equipment via graphical tools

It is ideal for intermittent issues as eBirdSoft is capable of continuously monitoring eBird body functions

It offers the possibility to implement IFS applications to generate fault reports.

It uses online real-time graphical tools to monitor critical conditions ( rotation matrix, force matrix etc)

Collaboration or contributions?:

If anybody is interested to contributing or has any questions regarding the project, please contact me at the following address:

yoyoberenguer@gmail.com

Yoann Berenguer

Observer ATS