

Foucault Pendulum Electronics Kit.

D08_Description of the GUI Software.

www.foucaultpendulum.nl

Document version	2026-05-26
Related Documents	Source code GUI_Pendulum (Lazarus project). D02_Options. D03_Functionality GUI_Pendulum. D06_Arduino PinUse_Messages. D09_Processing_PositionData. D14_Development_Environment & Coding Rules.

In brief:

The GUI_Pendulum program (short: the GUI) is a PC program which monitors and displays the pendulum movements, allows the operator to set many parameters and logs the behaviour of the pendulum and the equipment.

This document gives short descriptions of the files in the project and explains several calculations which are done to present the data in a clear way.

The program is written in FreePascal with the Lazarus IDE. and communicates with the hardware (PCB BobControl en associated electronics) by Ethernet via the local area network (LAN)

The program is published only as the (complete) source code. The user needs to install the development environment (Lazarus IDE) on his/her own PC.

In almost all cases the user will need to make changes to the program to make it fit for the particular pendulum.

The program source code consists of a number of files, in the project directory. (downloadable as a .zip file).

GUI_Pendulum.pas: The startup program. This is a very short piece of code which rarely needs attention.

Most of the work is done in the following 5 units and therefore they are prone to need changes:

u_main.pas

- Contains all controls on the main form, as much as possible grouped by the *GroupBoxes*.
- Procedures for reading and writing the the settings to/from the Parameter file.
- Has the procedure FormCreate where the other forms are initialized and the *GroupBoxes* get their sizes and positions.

If you need to add / remove / change controls it should be done here.

u_messages.pas

- Contains the message preparation, exchange and decoding for the Ethernet communication with the electronics.
- Here most of the values are put into the textboxes and status indicators are set.
- During processing the *InMessage* the procedure *ProcessPositionData* in **u_calc.pas** is called.

u_calc.pas

- Contains the more sophisticated calculations with the data from the Position Measuring System (PMS) like finding the Precession Angle and the lengths of the major and minor

ellipse axis. From here also the Compass display is maintained.
A detailed description of the processing step is in DO09_Processing_PositionData.doc

u_logging.pas

- Has the routines for Data Logging and generating new logfilenames.
If you need to change the logging format, here is where to do it.

u_timedisplay.pas

- Maintains the graphs on the TimeDisplay.

The data are stored in *traces* from where they are displayed. Data points enter on the right side of the graph and shift to the left on each entry. We have a "fast" and a "slow" set of traces. The "fast" set is updated at each HalfSwing of the pendulum, the "slow" set goes in 15 minutes, 1 hour or 5 hours per division.

And we have a number of supporting units which will rarely or never need to be changed.

u_scope.pas

- Basic functions for the TimeDisplay.

u_compass.pas

- Basic functions for the CompassDisplay.

u_numstrings.pas

- Some numeric to / from string conversions.

u_timestrings.pas

- Some time functions, a.o. for filename generation.

and the files from the **synapse library** for the ethernet communication: **blcksock.pas** and names starting with **syn*****.pas**. You should never need to edit one of these.

There are two other forms in the project:

u_large_adc.pas

Shows a form with large textboxes showing the raw data from the A/D converters.
This form is helpful during the gain adjustment of the PMS channels because it can be read from quite a distance.
It is activated by the checkbox *Show Large Form* in the PMS pane.

u_corrections.pas

Shows a form with some textboxes to enter certain correction data for future use.
Background: I've found that the PMS suffers from a small amount of non-linearity and cross talk between the NSEW channels, resulting in some pillow shaped distortion. I've planned to develop correction algorithms for this.
It is activated by the checkbox *Show Corrections* in the PMS pane, but currently does nothing.

Other files:

There are no special library files to be installed in the Lazarus / FreePascal IDE.

Files with the extension **.lfm**, **.lpi** and **.res** are maintained automatically and never need attention.

The file with extension **.lpi** is the project startup file. It should have a file association set such that double clicking on it from within a file manager will start the *Lazarus IDE*.

When the program has been running a file "Parameters.txt" is generated. This file contains the saved values of a number of parameters, and is read at the start of the program. Data are saved at each "Change Parameters" buttonpress.
Also a subdirectory "LogFiles" is created for the logfiles and saved .PNG images.