inductive calibration GUI

Release V1.0

Martijn Schouten

Dec 01, 2021

CONTENTS:

This documentation documenents the code of a GUI for calibrating a Diabase H-Series 3D printer in x and y using and LDC1101EVM evaluation module. The easiest way to run a frozen binary which can be found in releases

The inductive calibraiton GUI consists of three classes. The mainwindow of the app contain the entire GUI. The diabase class implements the communication with the diabase 3D printer and the ldc1101evm class implements the communication with the LDC1101EVM evaluation module.

APP MAINWINDOW CLASS

class app.**MainWindow**(**args*, ***kwargs*)
Bases: `PyQt5.QtWidgets.QMainWindow`

apply_offsets()
Function for handling the apply offset button being pressed. This will send the measured offsets to the printer. :return: None :rtype: None

ascend = True
If the tools should be calibrated in ascending (True) or descending (False) order.

ascend_changed()
Function for handling the ascend checkbox being pressed. This will update the ascend setting and deselect the descend checkbox. :return: None :rtype: None

calibrate(*cal_x*)
Function for performing a calibration in x or y. This will just record the LDC1101EVM sensor values until the stop button is clicked and store the result in the file specified in the filename textbox. :param cal_x: If True, calibrate in the x direction. If False, calibate in the y direction. :return: False if unsucceful, True if succefull :rtype: Boolean

calibrate_x()
Function for handling the calibrate x button being pressed. This will run the calibration procedure and find the x offsets. :return: None :rtype: None

calibrate_y()
Function for handling the calibrate y button being pressed. This will run the calibration procedure and find the y offsets. :return: None :rtype: None

clear_figure()
Function for handling the clear figure button being pressed. This will clear the graph in the GUI and reinitialise it.

Returns None

Return type None

closeEvent(*event*)
Function for handling the window being closed. This makes sure the settings are saved when the window is closed. :return: None :rtype: None

connect()
Function for handling the connect button being pressed. This will attempted to connect to the selected COM ports.

Returns True if succesfull, False if unsuccesfull

Return type Boolean

`connected = False`
: If a connection to the LDC1101EVM and diabase has already been made

`descend_changed()`
: Function for handling the descend checkbox being pressed. This will update the ascend setting and deselect the ascend checkbox. :return: None :rtype: None

`duet_port = ''`
: The name of the port the duet is connected to

`find_symmetry_axis(`*x*, *y*`)`
: Function for calculating the point of symmetry of a a symmetric curve :param x: List of x coordinates :param y: List of y coordinates :return: The oint of symmetry :rtype: float

`func(`*x*, *o*, *a*, *b*, *c*, *d*, *e*`)`
: Polynomial function fitted to the measured inductance curve to determine the point of symmetry :param x: List of x coordinates at which the function should be evaluated :param o: The point of symmetry :param a: Constant offset :param b: Constant before the square :param c: Constant before the to the power 4 :param d: Constant before the to the power 6 :param e: Constant before the to the power 8 :return: The output of the polynomial function :rtype: Boolean

`load_settings()`
: Function for loading settings to a settings.yaml file :return: False if unsuccesful, True if succesfull :rtype: Boolean

`offset_direction = True`
: If True the last run calibration was in the x direction, if False it was in the y direction

`offset_list = []`
: A list with the last found tool offsets belonging to the tools in *`MainWindow.offset_tool_list`*

`offset_tool_list = []`
: A list of the tool numbers belonging to the tool offsets in *`MainWindow.offset_list`*

`output_to_terminal(`*new_text*`)`
: Function for writing output to the terminal text box.

  Returns None

  Return type None

`reload()`
: Scans all COM ports and checks the name of all COM ports. If a name with “USB Serial Device” or “Duet” is found this it is selected as the printer port. If a name with ‘EVM’ is found this port is selected to be the port with the LDC1101EVM.

  Returns None

  Return type None

`save_settings()`
: Function for saving settings to a settings.yaml file :return: None :rtype: None

`stop()`
: Function for handling the stop button being pressed. This will set a variable that will stop the running processes when possible.

  Returns None

  Return type None

`stop_button_clicked = False`

Becomes True if the stop button has been clicked, until the measurement is stopped, then it becomes False again

`test_sensor()`

Function for handling the test sensor checkbox being pressed. This will just record the LDC1101EVM sensor values until the stop button is clicked and store the result in the file specified in the filename textbox. :return: None :rtype: None

`update_tool_list()`

Function for reading out the selected tools and the reference tool and putting them in the right order. The reference tool always will go first, then the other tools follow in either ascending or descending order, depending on whether ascend or descend is selected.

Returns None

Return type None

`app.main()`

CHAPTER

TWO

DIABASE CLASS

class diabase.**diabase**(*port*)

Bases: `object`

attempts = 1000

The number of lines to read before deciding the 'OK' from the printer will never arrive

close()

Function for closing the serial communication with the printer

Returns None

Return type None

get_current_position()

Function for getting the current position of the printer using a M114 command

Returns Dict with the current position. The dict contains a key 'x', 'y' or 'z' with the current position in the corresponding direction.

Return type Dict

set_tool_offset(*tool*, *pos*)

Function for setting tool offsets.

Parameters

- **tool** – The tool number of the tool of which to set the offsets
- **pos** – Dict with the tool offsets. The function expect a key 'x', 'y' or 'z' with the tool offset in the corresponding direction.

Returns None

Return type None

set_tool_offset_differential(*tool*, *extra_offset*)

Function for setting tool offsets relative to the current tool offsets. To do so the printer will:

- Select the tool
- Get the current position
- Set the tool offset to zero
- Measure the position again
- -Set the tool offset to the last measured tool offset plus the addional tool offset

Parameters

- **tool** – The tool number of the tool of which to set the offsets

- `extra_offset` – Dict with the additional tool offsets. The function expect a key ‘x’, ‘y’ or ‘z’ with the additional tool offset in the corresponding direction.

Returns None

Return type None

`store_offset_parameters()`

Function for storing the current tool offsets in flash such that they will still be there when the printer is restarted.

Returns None

Return type None

`write_line`(*string*)

Write a line of GCODE to the printer. This function will wait for an ‘OK’ from the printer, meaning that the command has finished executing (except for G1 commands). If it takes too to many attempts for the printer give an answer it will be assumed something went wrong and the function will return anyways.

Parameters `string` – The line of GCODE to write to the printer.

Returns None

Return type None

CHAPTER

THREE

LDC1101EVM CLASS

class ldc1101evm.**ldc1101evm**(*port*)

Bases: `object`

Csensor = 1.2e-09

Value of the capacitor soldered onto the LDC1101EVM. This will affect the measured inductance since the LDC1101EVM determines the osciallation frequency of an LC tank with this capaictor and the inductor to be measured.

LHR_init()

Function for initialising a high resolution measurement. A high resolution measurement is 24 bit and has no R measurement.

Returns None

Return type None

close()

Close the serial connection and tell the daemon to go kill itself.

Returns None

Return type None

flush()

Delete all currently stored measurements

Returns None

Return type None

get_LHR_data(*down_sample_ratio*)

Function getting the inductance measured by the LDC1101EVM in LHR mode. To put it in LHR mode run *ldc1101evm.LHR_init()* first. This function blocks until an inductance value that has not been read is available. To delete all currently stored measurements run *ldc1101evm.flush()* first.

Parameters **down_sample_ratio** – How much the output should be downsampled. This reduces the sampling rate but increases the effective resolution by taking the average.

Returns The measured inductance

Return type float

lock = <unlocked _thread.lock object>

Mutex for making sure the serial daemon and the other functions don't try to access *ldc1101evm.received_bytes* at the same time

received_bytes = b''

Stores all the bytes received from the LDC1101EVM

`serial_daemon()`
The serial daemon which is run in a seperate thread as the rest and just puts all the received bytes in *`ldc1101evm.received_bytes`*

Returns None

Return type None

`stop_thread = False`
If set to True, the serial daemon will kill itself

PYTHON MODULE INDEX