PLATFORM DEVELOPMENT TO EXTRACT FEATURES FROM PEN TABLETS

A PROJECT REPORT

Submitted by

SOUVIK CHAKRABORTY (RA1611017010051)
SOUHARDYA MOULIK (RA1611017010076)

Under the guidance of

DR P.A SRIDHAR
Assistant Professor
Department of Electronics and Instrumentation Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND INSTRUMENTATION

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
May 2019

BONAFIDE CERTIFICATE

Certified that this project report "PLATFORM DEVELOPMENT TO EXTRACT FEATURES FROM PEN TABLETS"

is the bonafide work of SOUVIK CHAKRABORTY (RA1611017010051)
SOUHARDYA MOULIK (RA1611017010076)

who carried out the project work under my supervision.

SIGNATURE

SUPERVISOR
DR. P. A. SRIDHAR
DEPT. OF ELECTRONICS AND INSTRUMENTATION ENGINEERING.
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS:

1. PD	- PARKINSON’S DISEASE
2. API	- APPLICATION PROGRAMMING INTERFACE
3. UPDRS	- UNIFIED PARKINSON’S DISEASE RATING SCALE
4. CISP	-COMPOSITE INDEX OF SPEED AND PEN-PRESSURE

ABSTRACT

The speed and pressure data from a person handwriting stroke can have many applications in medical and psychological analysis. This data will be gathered from the “subject” writing on a commercial pen tablet and subjected to possible mathematical analysis. The relationship of penmanship and outlining of the winding has been built up in PD in beginning periods. Nonetheless, one weakness in the utilization of penmanship or outlining is the requirement for a specialist to decipher the portrayals, particularly in the beginning times of the malady. With the accessibility of advanced gadgets that are appropriate for account hand-drawing, there is the potential for machine-based appraisal of composing and outlining. These gadgets are likewise appropriate for acquiring the dynamic highlights of penmanship, which are reasonable for continuous and solid examination. These highlights can be gotten consequently permitting fast on-line appraisal of patients and produced for applications, for example, biometrics and characteristic markers for PD. The point of this work was to set up a dependable PC based winding portraying technique for evaluation of the seriousness of the infection. This examination has researched the elements of drawing a winding to recognize sound subjects and PD patients with various dimensions of seriousness and proposes another component with more grounded relationship with the seriousness of the sickness. The prior investigations have set up that speed and pen-weight amid outlining lessen with the headway of the sickness, however did not think about the blend of these two parameters.

CHAPTER 1

INTRODUCTION

1.1 AIM

To design a software prototype which will monitor the various parameters associated with handwriting data. This project's purpose is to create a windows-based software capable of capturing handwriting data through capable hardware.

The software shall be proprietary and will work with a pen-digitizer hardware setup of the manufacturer: "WACOM". The recorded data shall be stored locally on the system running the software. This recorded data sets to be in CSV and text formats. The point of this work was to set up a dependable PC based winding portraying technique for evaluation of the seriousness of the malady. This examination has researched the elements of portraying a winding to recognize solid subjects and PD patients with various dimensions of seriousness and proposes another component with more grounded relationship with the seriousness of the illness.

1.2 OBJECTIVES

1. To create a windows compatible application to work with WIN32 API and WINTAB drivers.

2. To create a software DAQ to accumulate various captured data to locally stored files.

CHAPTER 2

LITERATURE REVIEW

2.1 Distinguishing Different Stages of Parkinson’s Disease Using Composite Index of Speed and Pen-Pressure of Sketching a Spiral

The speed and pen-pressure while drawing a winding are lower among Parkinson's ailment (PD) patients with higher seriousness of the illness. In any case, the relationship between these highlights and the seriousness level (SL) of PD has been accounted for to be 0.4. There is a requirement for recognizing parameters with a more grounded connection for thinking about this for precise determination of the sickness. This examination has proposed the utilization of the Composite Index of Speed and Pen-weight (CISP) of portraying as a component for breaking down the seriousness of PD. A sum of 28 control gathering (CG) and 27 PD patients (absolute 55 members) were enlisted and evaluated for Unified Parkinson's Disease Rating Scale (UPDRS).

The exploratory convention was endorsed by RMIT University Human Research Ethics Committee and as per Declaration of Helsinki (updated 2004). All members were educated about the examination, and they gave their oral and composed assent before the beginning of the investigation. One inadequacy in the utilization of winding illustration is the critical variety between the diverse investigations. One alternative is the place the members are allowed to draw the winding, which has the impediment that there is critical between member changeability (23, 24). The other format is by giving a consistent winding to the members to follow, which anyway isn't attainable for some old members.

CHAPTER 3
IMPLEMENTATION

3.1 WORKING MODEL

1. The total occurrences of “pen up-down”, i.e, change in coordinate data when pressure data is nil.
2. The UI of the app will consist of a launcher window containing recording session screen.
3. The main UI will consist of the pen ink visualization, based on the pen data.
4. The recorded data will be exported as CSV files for further analysis.
5. A preliminary report consisting of scalar data will be exported in TXT format.

Fig 1.1: A typical pen tablet setup.

3.2 BASED ON WIN32 API

The Windows application programming interface (API) lets you develop desktop and server applications that run successfully on all versions of Windows while taking advantage of the features and capabilities unique to each version. The Windows API can be used in all Windows-based desktop applications, and the same functions are generally supported on 32-bit and 64-bit Windows. Differences in the implementation of the programming elements depend on the capabilities of the underlying operating system. These differences are noted in the API documentation. This was formerly called the Win32 API. The name Windows API more accurately reflects its roots in 16-bit Windows and its support on 64-bit Windows.

API Sets are strongly named API contracts that provide architectural separation between an API contract and the associated host (DLL) implementation. API Sets contains a subset of the Windows API. The decoupling offers many engineering advantages including reducing the number of DLLs loaded in a process.

Win32 API is a lot of capacities characterized in the Windows OS, at the end of the day it is the Windows API, this is the name given by Microsoft deeply set of utilization programming interfaces accessible in the Microsoft Windows working frameworks. It is intended for use by C/C++ programs and is the most immediate approach to interface with a Windows framework for programming applications. Lower level access to a Windows framework, for the most part required for gadget drivers, is given by the Windows Driver Model in momentum variants of Windows. Win32 utilizes an all-inclusive arrangement of information types, utilizing C's typedef component. These include:

- BYTE - unsigned 8-bit whole number.

- DWORD - 32-piece unsigned whole number.
- LONG - 32-bit marked whole number.
- LPDWORD - 32-bit pointer to DWORD.
- LPCSTR - 32-bit pointer to steady character string.
- LPSTR - 32-piece pointer to character string.
- UINT - 32-bit unsigned int.
- WORD - 16-bit unsigned int.
- HANDLE - murky pointer to framework information.

Obviously standard information types are additionally accessible when programming with Win32 API.

In Windows, library code exists in various structures, and can be gotten to in different ways. Typically, the main thing that is required is to incorporate into the fitting header document on the source code the data to the compiler and connecting to the .lib record will happen amid the connecting stage. This .lib document either contains code which is to be statically connected into gathered article code or contains code to enable access to a powerfully connection to a double library(.DLL) on the framework. It is likewise conceivable to create a double library .DLL inside C++ by including suitable data, for example, an import/send out table when gathering and connecting. DLLs represent Dynamic Link Libraries, the essential document of capacities that are utilized in certain projects. Numerous more current C++ IDEs, for example, Dev-CPP bolster such libraries. Basic libraries on Windows incorporate those given by the Platform Software Development Kit, Microsoft Foundation Class and a C++ interface to .Net Framework congregations. In spite of the fact that not carefully use as library code, the Platform SDK and different libraries give a lot of institutionalized interfaces to objects open by means of the Component Object Model actualized as a major aspect of Windows.

3.3 BASED ON WINTAB API

Wintab is an API that is included in Wacom tablet drivers. Wintab is a de facto API for pointing devices on Windows. The Wintab specification is an open industry standard interface that provides access to pointing devices such as a pen tablet, for example. The API was developed by LCS/Telegraphics.

The Wintab module is a wrapper around the Wintab Developer Kit and can be used to add tablet support to your Python application. Before you can get any data from the tablet you have to create an instance of the Context class, set the desired parameters and open the context. Once the context is open you will either receive messages from the tablet driver or you can poll the current tablet state. Either way, you will receive the tablet data via Packet objects that contain all the data that was generated by the tablet. The constants used in this module are either available in the wintab module or in the wintab.constants module. The latter can be used if you want to import only the constants into your namespace.

- cgkit.wintab.available()

Returns True if the Wintab functionality is available. As the Wintab Developer Kit is only available on Windows, this function will always return False on other operating systems.

On Windows, this function can still return False if either the Wintab drivers are not installed or cgkit was compiled without Wintab support.

If this function returns False, an exception will be raised whenever you try to call another function or instantiate a class from this module.

- cgkit.wintab.info(category)

This function returns a dictionary with global information about the interface. category specifies the category from which information is being requested. It can be one of the values in the following:

- WTI_INTERFACE: Global interface identification and capability information
- WTI_STATUS: Current interface resource usage statistics
- WTI_DEFCONTEXT ...
- WTI_DEFSYSCTX ...
- WTI_DEVICES+n :Capability and status information for a device
- WTI_CURSORS+n : Capability and status information for a cursor
- WTI_EXTENSIONS+n : Descriptive information and defaults for an extension

3.4 SETUP OF THE PROJECT

3.4.1 TOTAL OCCURRENCES OF "PEN UP-DOWN", I.E, CHANGE IN COORDINATE DATA WHEN PRESSURE DATA IS NIL.

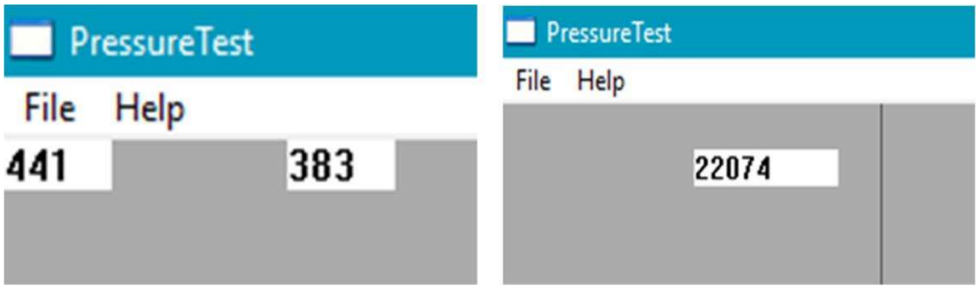
A ScribbleDemo.cpp file was developed to fetch data from the hardware setup. This module fetches various data as listed before according to some defined parameters and stores them locally for processing. This fetching takes place every 1.8 ms. The official WIN32 API used for this project is open source. The technical details and code is presented in the appendix. The other libraries used are:

1. <vector>,<map> : Vectors are same as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted, with their storage being handled automatically by the container. Vector elements are placed in contiguous storage so that they can be accessed and traversed using iterators. In vectors, data is inserted at the end. Inserting at the end takes differential time, as sometimes there may be a need of extending the array. Removing the last element takes only constant time because no resizing happens. Inserting and erasing at the beginning or in the middle is linear in time. In this project, we have mainly used it for data structure purposes.

2. <sstream>,<iostream>,<fstream> : The stringstream, ostringstream, and istringstream objects are used for input and output to a string. They behave in a manner similar to fstream, ofstream and ifstream objects. Here, in this project we have used it to create output text.

3. <time.h> : The time.h header defines four variable types, two macro and various functions for manipulating date and time. The time function returns the number of seconds elapsed since midnight (00:00:00), January 1, 1970, Coordinated Universal Time (UTC), according to the system clock. The return value is stored in the location given by destTime. This parameter may be NULL, in which case the return value is not stored. time is a wrapper for _time64 and time_t is, by default, equivalent to __time64_t. If you need to force the compiler to interpret time_t as the old 32-bit time_t, you can define _USE_32BIT_TIME_T. This is not recommended because your application may fail after January 18, 2038; the use of this macro is not allowed on 64-bit platforms. In this project, we are using it to create unique filenames based on date and time of data creation.

The raw X and Y coordinate data is printed on the screen along with the pressure value obtained as an unsigned int, as shown in Fig 1.2.



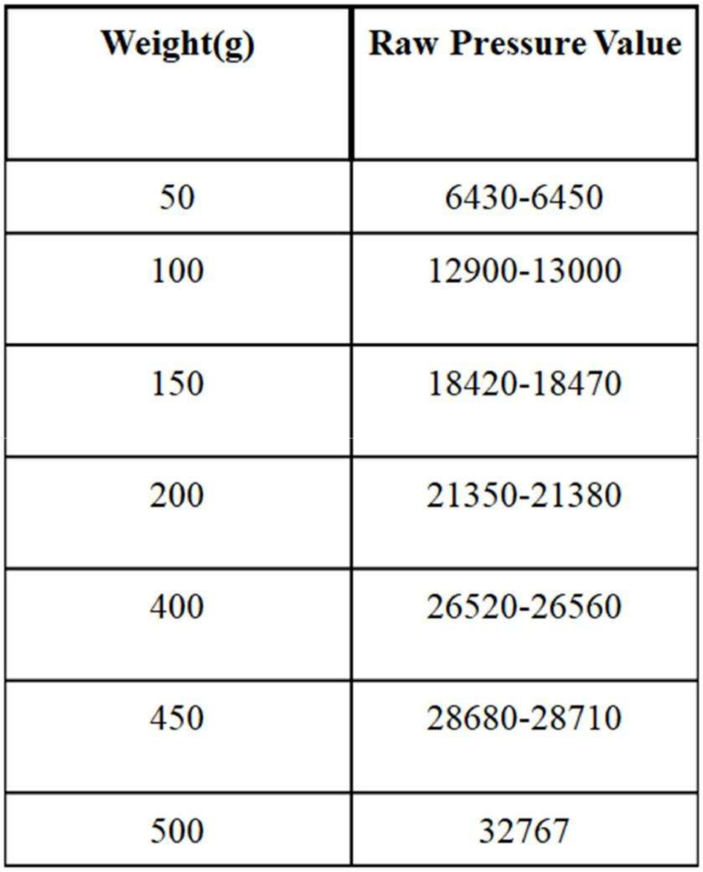
```
441                383
```

```
22074
```

Fig 1.2: The UI interface demo showing coordinate and pressure data of patient's pen position

The normal pressure exerted by the pen on the surface of the tablet was recorded. Through hit and trial experimentation the raw pressure values recognized by the device was found to be in the range of "0-32767" unsigned integer. These values were calibrated against physical weights and the actual pressure values were found.

The calibration table shown on the next slide was performed using a setup consisting of weights attached to the pen with it being held upright by a support. Before the calibration it was made sure that the pressure registered by the device was zero.



Weight(g)	Raw Pressure Value
50	6430-6450
100	12900-13000
150	18420-18470
200	21350-21380
400	26520-26560
450	28680-28710
500	32767

Fig 1.3: Weight v/s pressure calibration test tabulation

3.4.2 PEN INK VISUALIZATION THROUGH UI

To provide a visual representation of the drawing a real time handwriting capture module was developed. Using coordinate and pressure values obtained from the subject writing on the tablet, a plot is dynamically made. This plot is nothing but the visual representation of the writing on the tablet surface, as shown in Fig 1.4. The plot uses raw real-time pressure and coordinate data from the pen. These data are then exported into a CSV File format for further calculations and analysis. “coordinatedata#datetime#.csv” and “pressuredata#datetime#.csv” are the two files generated in the root directory of the application. An additional “Report Generated#datetime#” is also generated as a text file in the root directory of the application.

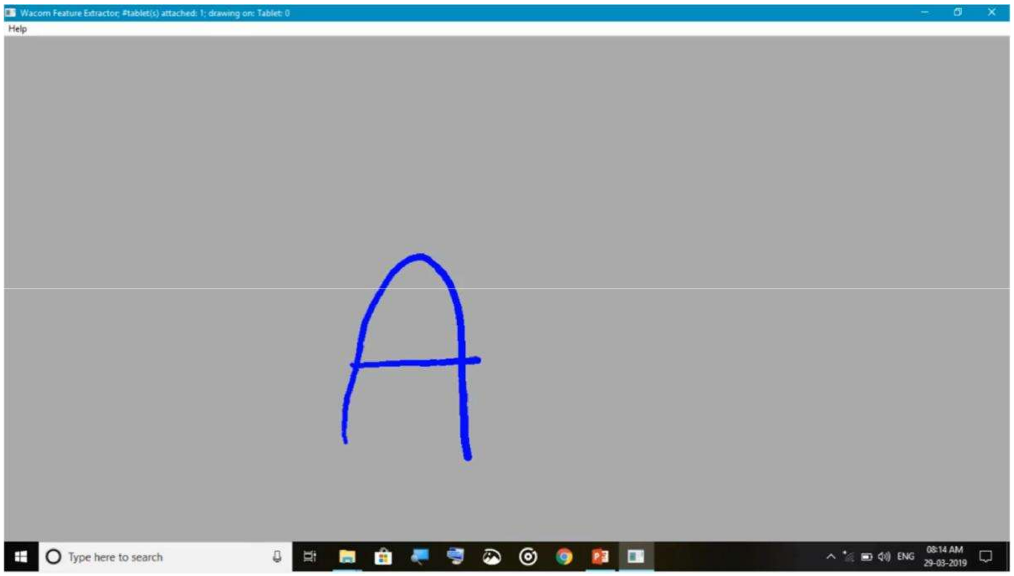

Wacom Feature Extractor; #tablet(s) attached: 1; drawing on: Tablet: 0
Help
Type here to search
ENG
08:14 AM
29-03-2019

Fig 1.4: Pen Ink visualization through application UI

CHAPTER 4

RESULTS

4.1 NOVELTY OF THE PROPOSED WORK

1. During this development, we have identified interoperability challenges related to the collection and processing of data from pen-tablet devices. We have developed a Win32 module which is compatible with a Wacom pen tablet Operating System.
2. The end-product is memory and processing power efficient.

4.2 DELIVERABLES

1. The project can significantly help in creating data-sets without hassle.
2. It is an alternative to manually obtaining such data through unspecialized third party application.
3. Given the lost-cost resources it needs to operate, this can be operated from even a Windows IoT version OS.
4. The simplistic on-point UI makes it easy for the user to operate and extract data using the application. Structured error messages for debugging is also supported.
5. Easy integration with any Wacom device makes it ready to use with multiple Wacom devices.

4.3 DATA SAMPLE TABLES

The patients were instructed to draw spirals (Fig 1.5) using both affected and unaffected hands. Multiple samples were recorded measuring to over 5000 KB of data. Presented below is a snippet of the data. The sketching of the spiral was recorded using commercially available A3 size tablet (Wacom Intuos Pro Large). The A3 paper was placed on the tablet, and Ink pen (Wacom Intuos ink pen) was used to sketch the spiral. This pen senses the location of contact, x and y, and the pressure, pr, between the tip and the paper. The left-corner of the sheet was considered to be the (0,0) point.

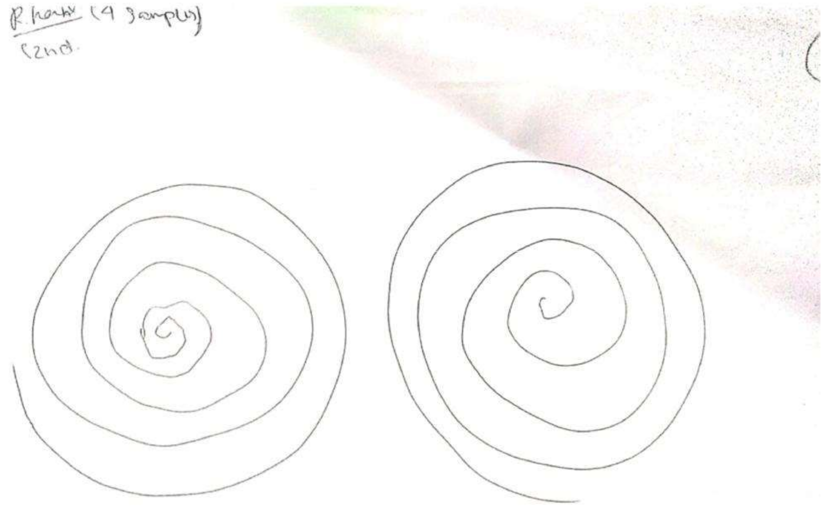


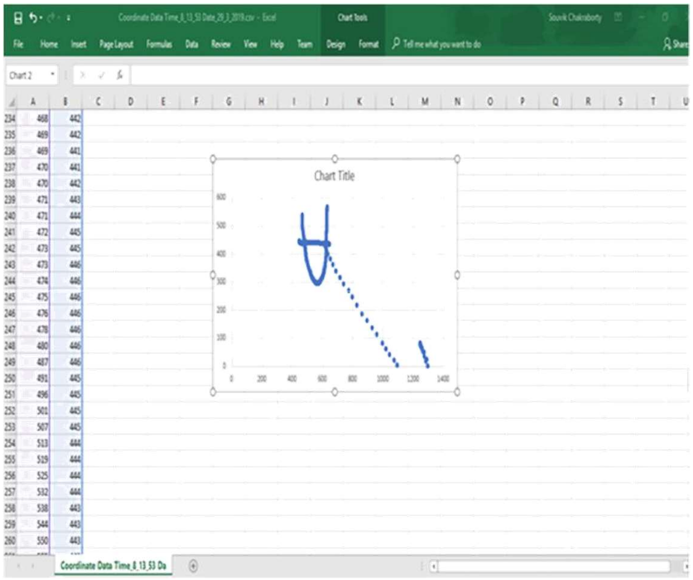
Fig 1.5: Spirals used to record the following data

Table 1.1: A sample of coordinate data obtained.

Coordinate X	Coordinate Y
507	305
508	303
508	302
508	301
508	299
508	298
509	297
509	295
510	294
510	293
511	290
512	289
513	288
513	287
514	286
532	268

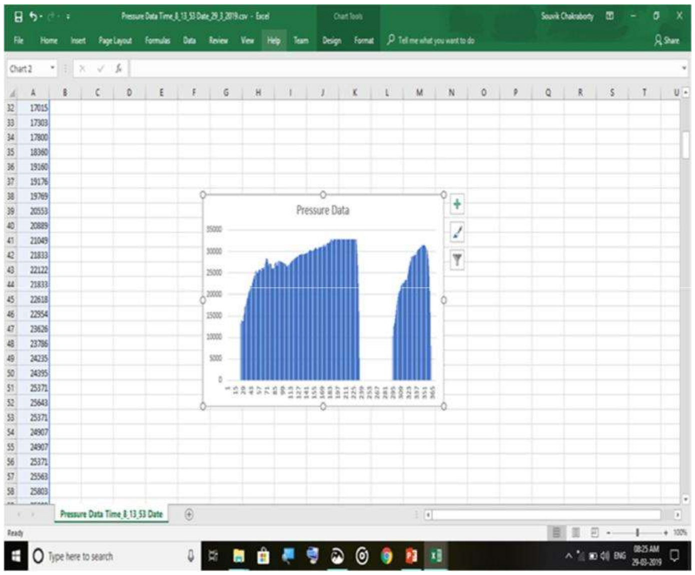
Table 1.2: Corresponding pressure data sample

Pressure value
3937
3889
3873
3873
3921
3921
3761
3745
3745
3745
3745
3761
3729
3761
3729
3825



	A	B
234	468	442
235	469	442
236	469	441
237	470	441
238	470	442
239	471	443
240	471	444
241	472	445
242	473	445
243	473	446
244	474	446
245	475	446
246	476	446
247	478	446
248	480	446
249	487	446
250	491	445
251	496	445
252	501	445
253	507	445
254	513	444
255	519	444
256	525	444
257	532	444
258	538	443
259	544	443
260	550	443

Fig 1.6: Plot of the coordinate data of a letter ‘A'



	A
32	17015
33	17303
34	17800
35	18360
36	19160
37	19176
38	19769
39	20553
40	20889
41	21049
42	21833
43	22122
44	21833
45	22618
46	22954
47	23626
48	23786
49	24235
50	24395
51	25371
52	25643
53	25371
54	24907
55	24907
56	25371
57	25563
58	25803

Fig 1.7: Plot of pressure data for a given letter ‘A’

CONCLUSION

The plots show that the obtained data makes sense in cross verification of characters. The various data that can be obtained from the plots show the characteristics of the patient's handwriting. Given a character 'A', the gap in pressure reveals a period of pen-up to dash the middle line of the character. This study has shown that speed, pen-pressure, and CISP of sketching of a spiral are negatively correlated with the severity of PD.

REFERENCES:

- https://docs.microsoft.com/en-us/cpp/windows/walkthrough-creating-windows-desktop-applications-cpp?view=vs-2017
- https://docs.microsoft.com/en-us/windows/desktop/WinProg/windows-data-types
- Zham Poonam et al.: Distinguishing Different Stages of Parkinson's Disease Using Composite Index of Speed and Pen-Pressure of Sketching a Spiral, Front. Neurol., 06 September 2017. https://doi.org/10.3389/fneur.2017.00435