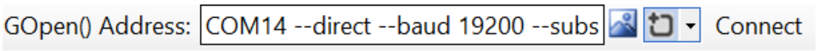
Designed by Patrick Kon, 2019-2020. For enquiries, reach me at: patrikon2-c@my.cityu.edu.hk

Foreword: This user manual is intended to be concise. It will start with a few "quick start" sections to get you moving quickly. For those who encounter problems or require more detail, do check the next few sections on movement types and PNA scanning options. Finally, I outline some known issues and less frequently used features.

Please read this carefully. If all else fails, you may reach me via email.

Connect:

1. Choose from the available IP addresses/ COM port numbers listed in the System Output menu.
2. In the following image, the chosen address/port to connect to is "COM14":

   GOpen() Address: COM14 --direct --baud 19200 --subs

   Replace your chosen connection address/port with "COM14", then press connect.

2a. Note, the default address (as of June 2020) is **192.168.1.1**

Quick start to movement (details in proceeding sections):

Preliminary steps (applicable to all movement types):

1. Connect
2. Select "Unit of choice" under settings.

Next steps specific to general relative movement:

1. Click any button under "General Relative Movement" tab.

Next steps specific to special movements:

1. Select "Mode" under "Special Movement" tab.
2. For manual input:
   a. Set 1 start point.
   b. Set 1 end point.
   c. Set **any number** of intermediate points.
   d. Configure PNA scanning options (section below).
   e. Click "Start special movement" button.
3. For segment input:
   a. Choose "Normal" or "Grid" mode.
   b. Set current segment coordinates by clicking on the respective "start", "end", and "delta" buttons.
   c. Click "Add segment" button to add the current segment to the queue (FIFO) of segments that will actually be run by the controller.
   d. To add more segments, repeat steps b-c any number of times.
   e. Configure PNA scanning options (section below).
   f. Click "Start special movement" button.

Designed by Patrick Kon, 2019-2020. For enquiries, reach me at: patrikon2-c@my.cityu.edu.hk

Quick start to PNA usage (details in proceeding sections):

*Note:

- **general relative movement does not support PNA scanning.**
- **Scanning is initiated every time a specified coordinate is reached.**

Pre-movement steps:

1. Select your options in “Special Movement” -> “PNA”

Post-movement steps:

1. Save PNA data
   Or
2. Clear PNA data

Designed by Patrick Kon, 2019-2020. For enquiries, reach me at: patrikon2-c@my.cityu.edu.hk

Directions of the motor:

There are **3 axes**: Axis-a, axis-b, axis-c.

*To understand intuitively, press the buttons in "General Relative Movement".

Axis-a: +ve (front), -ve (back)

Axis-b: +ve (left), -ve (right)

Axis-c: +ve (downwards), -ve (upwards)

The reference point/**origin** (0, 0, 0) is user determined: Click the "Set as Origin" button.

Henceforth, some terminology:

coordinates will refer to any integer combination of the 3 dimensions: Axis-a, Axis-b, Axis-c.

points are synonymous to coordinates.

segment will refer to a straight-line segment that has 3 parameters for Axis-a and Axis-b:

start Axis-a (resp, Axis-b) coordinate, end Axis-a (resp, Axis-b) coordinate, and increment-by-unit in Axis-a (resp, Axis-b).

grid will refer to a sequence of segments that has the same parameters as an individual segment:

In pseudocode, you can imagine the movement to be 2 for loops with Axis-a movement as the inner loop, and Axis-b movement as the outer loop.

```
for start Axis-b to end Axis-b:
    for start Axis-a to end Axis-a:
        Increment-by-unit in Axis-a;
    Increment-by-unit in Axis-b;
```

In other words, with each increment-by-unit in Axis-b from start Axis-b to end Axis-b, the controller will move along Axis-a from start Axis-a to stop Axis-b, in increment-by-units of Axis-a.

Designed by Patrick Kon, 2019-2020. For enquiries, reach me at: patrikon2-c@my.cityu.edu.hk

Unit types:

1. Stepper units: the actual displacement is inherent to each controller. The only way to gauge is via manual measurement.
2. Mm: a user calibrated estimation of standard mm.

Steps for user calibration of mm:

1. Select the tab under: “Settings” -> “Calibration (reset on close)”
2. The value inserted into the text box will be the user’s estimation of
   number of stepper units / mm

2a. Note, this calibration is reset to its default on closing the application. Currently, there is no way to persist the calibrated unit. If you require this feature, contact Patrick by email (stated above), and we can discuss.

3. Press the enter key. The calibration is complete.

Movements:

The movement types currently supported are:

Movement type	Brief explanation/ comparison
General relative movement	On button click, moves a set (user determined or default) number of **stepper units**.
Special Movement -> Manual Input	On “Start Special Movement” button click, moves iteratively to individual **coordinates** set by the user. Stepper units and mm are both allowed.
Special Movement -> Segment Input -> Normal	On “Start Special Movement” button click, moves iteratively to line **segments** set by the user. That is, the user can specify a queue of line segments that are to be traversed iteratively. Of course, the line segments can be disjoint. Stepper units and mm are both allowed.
Special Movement -> Segment Input -> Grid	On “Start Special Movement” button click, moves along the **grid** set by the user. The grid (above section) pattern is fixed. However, the user can choose the start and end points and increment-by units. Stepper units and mm are both allowed.

Other predefined convenience movements (located on top menu as buttons):

1. Return to Origin
2. Set as Origin

Movement type explanations in more detail:

General relative movement:

1. Only in stepper units.
2. Default value: 10000 stepper units.
3. The value applies to all axes uniformly.
4. To change the value:
    a. In the textbox of “Stepper Units (>300):
        i. Key in new value.
        ii. Press enter key.

All special movements:

1. The movement pattern is as follows:
    a. Move to coordinate specified by axis-a, axis-b, axis-c **resting** position (set via “Axis-c resting position” button).
    b. Stop.
    c. Move only axis-c to **drop** position (set via “Axis-c drop bar by” button).
    d. Start and complete PNA scanning.
    e. Move only axis-c to resting position.
    f. Continue to next coordinate and repeat (i.e. repeat steps a-e).
2. All initial values (i.e. start, stop, increment positions for manual input mode, and corresponding positions for segment input mode) are initialized to 0.

Special Movement -> Manual Input:

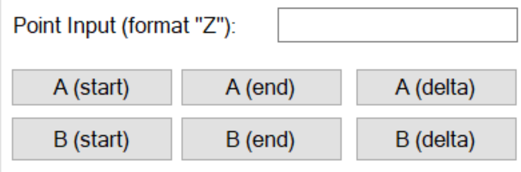
1. Input format: “Axis-a-coordinate, Axis-b-coordinate”

    Note: the comma and space

2. Input your values in the following child tabs of “Special Movement”:
    a. Manual input
    b. Axis-c input

Special Movement -> Segment Input -> Normal and Grid mode:

1. Input format: “Axis coordinate”
2. Input your values in the following child tabs of “Special Movement”:
   a. Input by segment
   b. Axis-c input
3. Axis-c resting and drop-by positions are **fixed**. That is, it remains the same for each coordinate traversed.
4. Explanation on the coordinates that are to be set in “Input by segment”:



- Start coordinate: **(** A (start), B(start), C(rest) **)**
- End coordinate: **(** A (end), B(end), C(rest) **)**
- The movement progresses from start to end via steps of **(** A(delta), B(delta), 0 **)** increments
- These conditions must be fulfilled:

$$\frac{A(end) - A(start)}{A(delta)} = \frac{B(end) - B(start)}{B(delta)}$$

$$\frac{A(end) - A(start)}{A(delta)}, \frac{B(end) - B(start)}{B(delta)} \in \mathbb{Z}$$

- Note therefore: none of the A and B options need to be the same, only if it takes the **same** number of steps to reach from A(start) to A(end), and B(start) to B(end).

PNA Scanning:

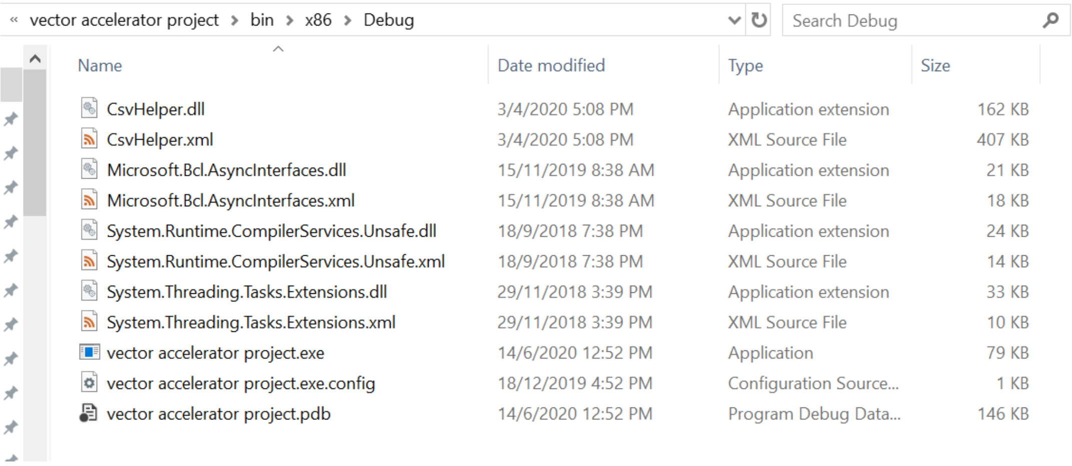
1. Scanning options are located in “Special Movement” -> “PNA”
   a. The interpretations are identical to the original PNA GUIs, please refer to their documentation.
2. On saving and clearing PNA data (located in the top left corner of the application):
   a. Data is appended to the existing cached PNA data, that is, data obtained from previous scans, after the completion of movements initiated by clicking the “Start special movement” button.
   b. To clear the cached data, press the “Clear PNA data” button.
   c. Else, press the “Save PNA data” button.

Slew speed configuration:

1. To change the slew speed for each axis: locate the buttons under textbox “Settings” -> “Slew speed”, key in your value
2. Default values (expressed in stepper units/s):
   a. Axis-a and Axis-b: 5000
   b. Axis-c: 400000

Known issues:

1. Startup issues troubleshooting:
   Make sure the required packages are available. For reference:



Name	Date modified	Type	Size
CsvHelper.dll	3/4/2020 5:08 PM	Application extension	162 KB
CsvHelper.xml	3/4/2020 5:08 PM	XML Source File	407 KB
Microsoft.Bcl.AsyncInterfaces.dll	15/11/2019 8:38 AM	Application extension	21 KB
Microsoft.Bcl.AsyncInterfaces.xml	15/11/2019 8:38 AM	XML Source File	18 KB
System.Runtime.CompilerServices.Unsafe.dll	18/9/2018 7:38 PM	Application extension	24 KB
System.Runtime.CompilerServices.Unsafe.xml	18/9/2018 7:38 PM	XML Source File	14 KB
System.Threading.Tasks.Extensions.dll	29/11/2018 3:39 PM	Application extension	33 KB
System.Threading.Tasks.Extensions.xml	29/11/2018 3:39 PM	XML Source File	10 KB
vector accelerator project.exe	14/6/2020 12:52 PM	Application	79 KB
vector accelerator project.exe.config	18/12/2019 4:52 PM	Configuration Source...	1 KB
vector accelerator project.pdb	14/6/2020 12:52 PM	Program Debug Data...	146 KB

2. When trying to run the executable file (vector accelerator project.exe), receive an error of the form: “Requires .NET framework X.XX.X , install now?”
   a. Solutions (choose either):
      i. reconfigure target framework when building the project, to align with the .NET framework version installed on your PNA machine. To check for the version (2nd solution): https://stackoverflow.com/questions/1565434/how-do-i-find-the-net-version
      ii. Download the required version onto the PNA (untested)

To compile/build and develop for a PNA machine:

Important note: this is tested on a PNA machine that:

1. Runs Windows 7, x86 architecture.
2. Has a USB port

Designed by Patrick Kon, 2019-2020. For enquiries, reach me at: patrikon2-c@my.cityu.edu.hk

Development platform: Visual Studio (2017/2019)

To develop for other machines: again, this has been only tested with the above setup. However, I am quite certain it will run properly on Windows 10. If USB port is not present, you will need to figure out how to connect the PNA to your PC (for transferring the .exe files, etc), via LAN.

<u>Step by step guide:</u>

1. Obtain the source code from google drive, with this path: "PNA Project Cloud Files -> GCLIB_GUI-Newmark-Agilent-".
2. On your development PC, install gclib package:
    a. Not all versions work. Depends on Windows version.
    b. For PNA with Windows 7 installed, install the gclib package from the executable located on google drive, with this path "PNA Project Cloud Files -> galil_gclib_450.exe"
    c. Purpose: this allows you to use the galil library functions, that control the motion controller.
3. On your development PC, install PNAProxy.exe
    a. Obtain this .exe file from either:
        i. (strongly recommended) On drive, with this path "PNA Project Cloud Files -> PNAProxy.exe"
        ii. Any PNA machine. Commonly at this path "Program Files / Keysight / Network Analyzer / Automation ".
    b. Note: PNAProxy.exe files on certain PNA machines are broken. I.e. the following error message will be prompted: "Error Reading Setup Initialization File". This is possibly due to file corruption. Fortunately, a PNAProxy.exe file from **any** PNA machine can work.
    c. Place this .exe in your development PC. Execute it, following the instructions.
        i. Note: the Computer Name needs to be filled in properly. Locate it by following the instructions, on your PNA machine.
    d. Once the installation completes, navigate to this path: "C:\Program Files (x86)\Common Files\Agilent\PNA". Note the file **835x.tlb**
    e. On Visual Studio, load your PNA project (i.e. the Visual Studio Project that you will use to build the executable file, complete with the GUI for use, that will be run on the PNA machine).
        i. Right click on the project in the "Solution Explorer".
        ii. Click on "Add -> References".
        iii. Browse to the location of the **835x.tlb** file, and press OK.
        iv. Purpose: this loads the PNA API onto your development PC.
4. Ensure that you have installed CsvHelper on Visual Studio.
    a. Note: this comes pre-installed if you used downloaded source code in Step 1, as your Visual Studio Project.
5. Build the project with these settings:
    a. for a x86 CPU.
    b. Target Framework: .NET Framework 4.7.2
    c. Note: these settings varies according to the PNA machine used.
6. Navigate to the "bin/ x86 / debug " directory of your Visual Studio Project.
7. Copy the entire folder's contents. Place them in your PNA machine. Ready to run.

Designed by Patrick Kon, 2019-2020. For enquiries, reach me at: patrikon2-c@my.cityu.edu.hk

To obtain the current built (working) executable file:

1. Navigate on google drive to this directory: "PNA Project Cloud Files -> Final executable backup".
    a. Or alternatively, you may obtain this same directory, "Final executable backup", at: https://github.com/patrickkon/GCLIB_GUI-Newmark-Agilent-
2. Copy the entire directory, place it on your PNA machine.
3. Execute vector_accelerator_project.exe"