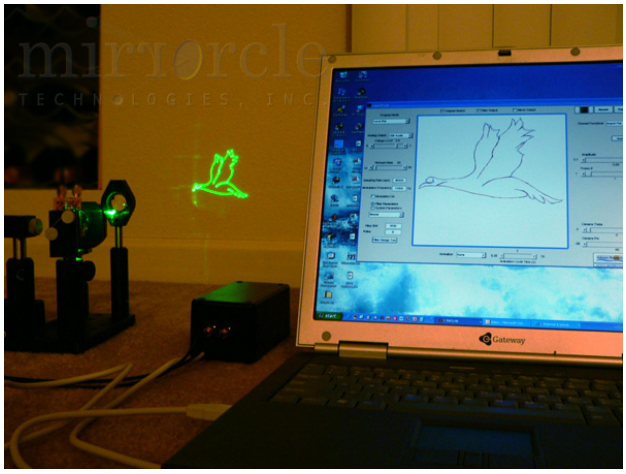
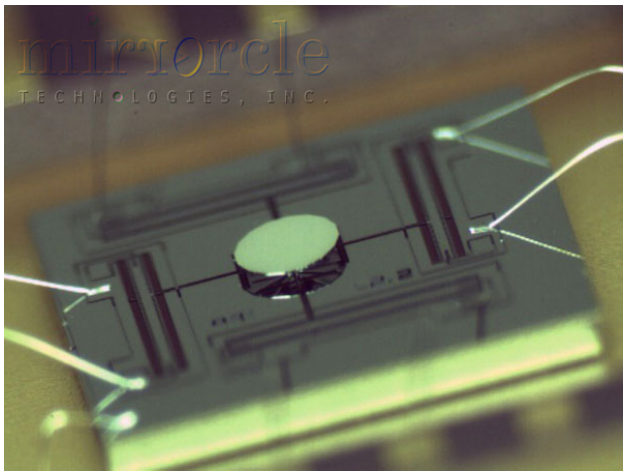
Mirrorcle Software Suite Manual & Users Guide

Contents:





Authors:

Kenneth Castelino, Ph.D.
Veljko Milanovic, Ph.D.
Andrew Miner, Ph.D.
Mirrorcle Technologies, Inc., Albany, CA
support@mirrorcletech.com

MirrorcleDraw

Introduction

MirrorcleDraw is a vector-scanning software that can be used to control two axes (*tip* and *tilt*) of a MEMS micromirror in a laser-beam steering system so as to deflect the laser beam in a desired angle or direction. For laser beams that are terminated on a flat screen or a wall, this system will create "drawings" which can be static or animated. The general operation of the program is to take inputs from a user in terms of the desired pattern that the laser beam is supposed to follow, the refresh rate which determines the speed at which the laser will trace that pattern and other optional inputs. The program then computes the necessary voltage signals for each axis to achieve that result and sends them to the user's desired output port of the computer. If the output device is an NI-DAQ card, then the X,Y data signals are sent using the 2 analog outputs. If a NI-DAQ card with correlated DIO, such as the PCI-6221 is used, a synchronized digital blanking signal is output in addition to the X,Y data signals (on Digital Output ch. 0 or P0.0). Kits that ship with software versions 8.0 and newer allow the programmer to address all 8 bits of this correlated DIO port, i.e. P0.0 through P0.7, however MirrorcleDraw executable software does not implement this capability but only drives one line, P0.0. C++ examples in MirrorcleAO SDK (8.0 and newer version) demonstrate the use of the entire correlated DIO port. Finally, in the case of the USB-based development kit based on NI USB-6221 card, signals are sent over the USB cable and converted to high-voltage X and Y axis controls. The blanking signal can be used to switch the display laser to display disjoint objects. Read the Installation and Setup section of this manual to determine if all requirements of the program and overall system are met properly. For a more technical discussion on the operating principles and applications of micromirrors, please visit the Mirrorcle Technologies website.

System Requirements & Installation

Software Requirements (depending on the version of the hardware:)

1) Windows XP is the preferred operating system.
2) NI-DAQmx drivers 8.x or later must be installed – the DAQmx drivers are provided along with the DAQ card or can be downloaded from the NI website.

Hardware / Software Setup

1. Install pre-requisite (required) software listed above for the specific hardware version you will be using.
    1. If using an audio card, install DirectX if required for your version of Windows.
    2. If using a DAQ card (e.g. PCI-6221), install the DAQmx drivers that come along with the card. Shutdown the computer, install the card in any available PCI slot and restart. Windows should detect the card and install required drivers. Verify that the card is present by opening the **Measurement & Automation Explorer (MAX)** from the National Instruments program group in the Windows Start menu. Sometimes the item in the program group appears only as "**Measurement & Automation."** Open the device list under **My System/Devices and**

Interfaces/NI-DAQmx Devices, and check that your DAQ card is listed under 'DAQmx devices'.

3. If using the Data Translation USB Controller, install Data Translation software and drivers, and then plug in the HV Amplifier box USB cables to the computer and complete device installation by following the install new hardware screens. The drivers should automatically install due to the prior installation of the Data Translation CD.

2. Unzip the executable Mirrorcle Software Suite file on the MirrorcleDraw CD. This will automatically extract all of the files into default directories already pre-determined in the setup file. This will automatically create the files in:
   C:\Program Files\Mirrorcle\
   And will also create a Windows Start Menu program group Mirrorcle with shortcuts to your software and user guides.
   Please allow the executable extractor to place the files in the pre-determined (default) directories.

Program Operation:

An executable file is provided for output via the National Instruments DAQ cards: MirrorcleDraw-Daqmx.exe

In addition to the windows executable MirrorcleDraw, a second executable file, MirrorcleAO-DAQmx.exe is included with the CD. This runs a very simple example of C code that was created using the included MirrorcleAO-C++ Software Development Kit (described in this document.) The source files for this example are all included in the MirrorcleAO-Cpp directory and should allow a programming-skilled user to quickly develop new applications that use the MirrorcleTech devices.

Global Controls:

1) Max Voltage: The slider bar limits the maximum voltage sent to the device. This is a scaling parameter and depends on the output voltage of the audio/DAQ card and the amplifier gain.

2) Sampling Rate: Most sound cards support 44100 sps. For DAQ devices the default rate is 100000 sps. The program automatically determines the maximum sampling rate and prevents the user from exceeding this rate.

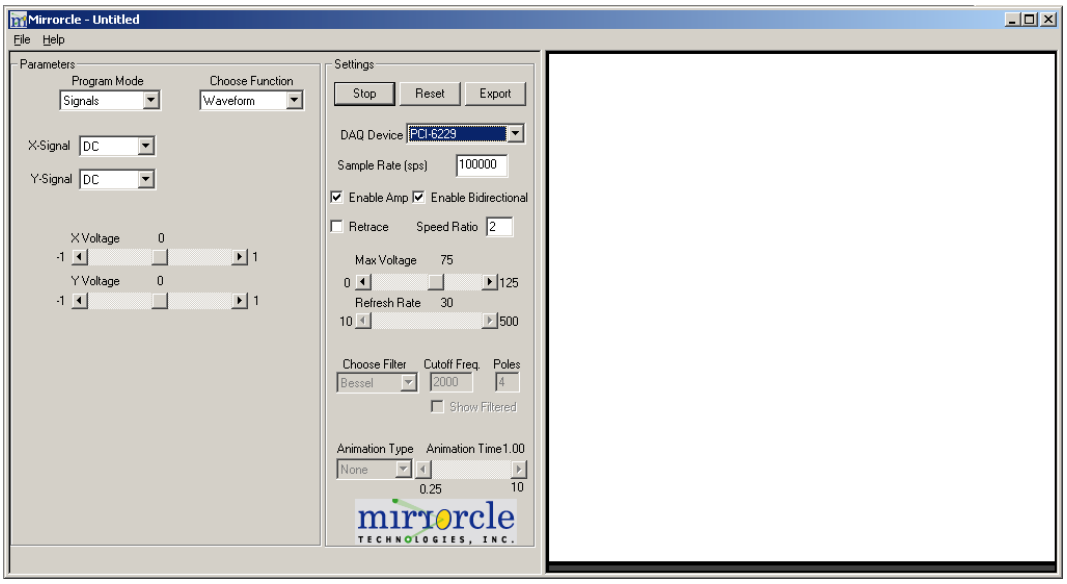
3) Refresh Rate: This slider sets the effective refresh rate of the drawing. For a given sampling rate, as the refresh rate is increased, the number of points per frame drops resulting in greater loss of spatial information in the drawing depending on its complexity.

3) Modulation: This is only used for audio output. If modulation is turned on, all the sketches and waveforms are modulated at the carrier frequency which can be specified. This setting is off by default but may be needed if the audio card does not support DC or low frequency signals.

5) Retrace & Speed Ratio: The on/off ratio can be used to specify the density of points when the laser is on/off. Typically the mirror can be moved much faster when the laser is off without degrading the drawing. However, using too high a ratio can cause ringing - values in the range 2-6 give reasonable results. Finally, the user can also choose to either retrace a path or go directly from the last point to the first. Retracing results cannot handle complex forms but is smoother.

6) Filter Settings: A set of filters is used to process the raw drawing signals in order to prevent resonant excitation of the mirror. Three types of filters available: Butterworth, Bessel. The user can choose the passband and number of poles for the Butterworth and Bessel filters. This filtered waveform is displayed as a red curve and is changed to reflect any changes in inputs or filter settings. The filtered output display can be toggled on/off using the Filter display checkbox.

7) HV Amplifier On: This is only present in the DAQ program. This switch toggles the HV Amp on/off using a digital line – toggling it off shuts off the amplifier and is a safety measure.



MirrorcleDraw GUI

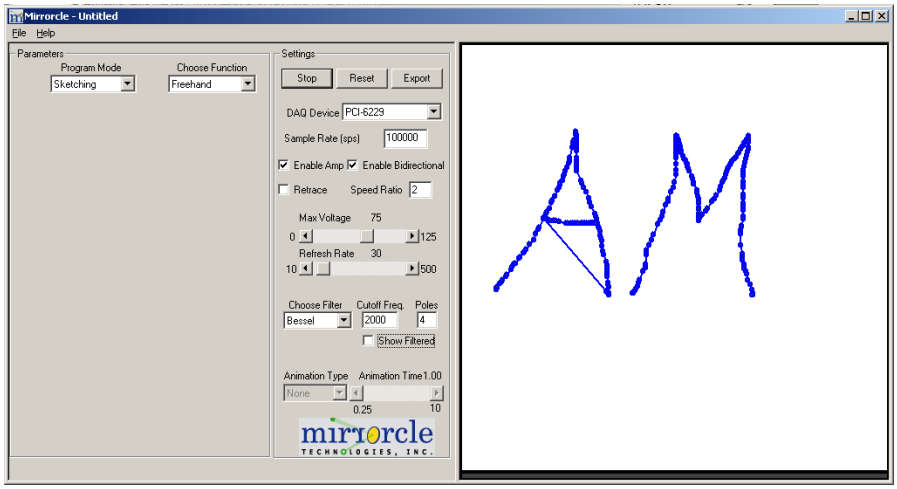
Program Modes

The program has a number of different modes which can be selected using the drop-down list at the top left corner of the GUI.

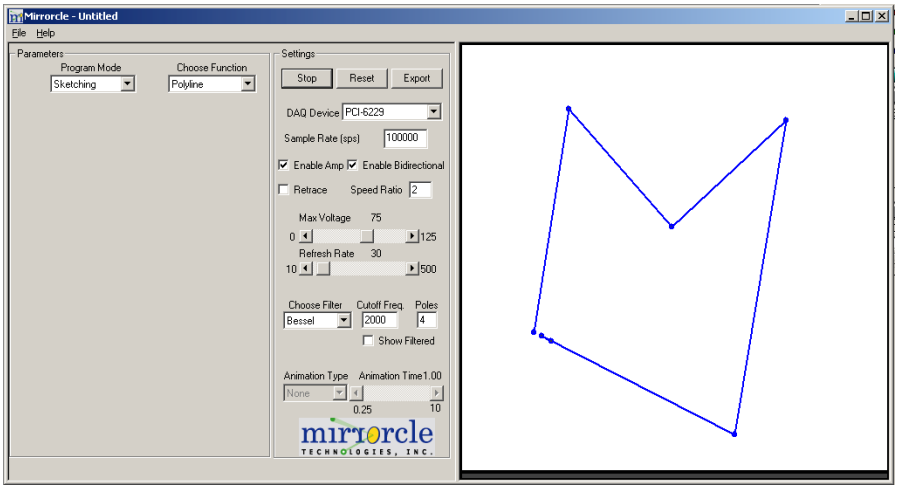
Sketch Mode:

There are two separate sketching programs in this mode:

1) **Freehand:** You can draw a shape by clicking the left mouse button and dragging to trace out the curve. Releasing the mouse button terminates the curve segment. The user can draw multiple disconnected curve segments in this fashion and the blanking information is automatically computed. Right clicking closes the sketch by connecting the first and last points. To clear the sketch, press the reset button. The pattern is displayed on the screen by the mirror and the stream can be stopped or restarted using the start/stop button.



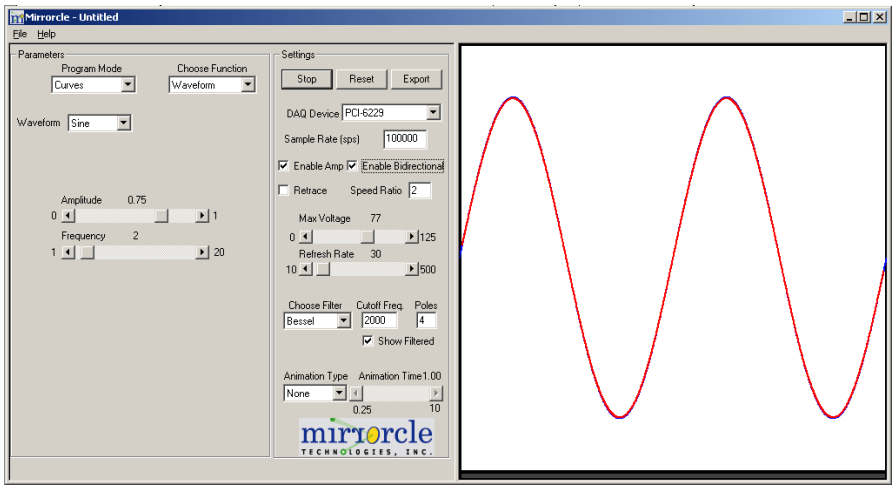
2) **Polyline:** This mode can be used to draw polygons that are open or closed. Clicking with the left mouse button adds points to the existing polygon. Keeping the left mouse button pressed and moving the mouse can be used to dynamically place the new point. Releasing the left button confirms the addition of the point. In order to close a polygon, click the right mouse button, which joins the first and last points. It is also possible to modify the existing polygon. Click the left mouse button on the point you want to move and drag the mouse to move the point to its new location. Right clicking in the vicinity of the point will delete the point. Deleting the first/last points of a closed polygon results in opening of the polygon.

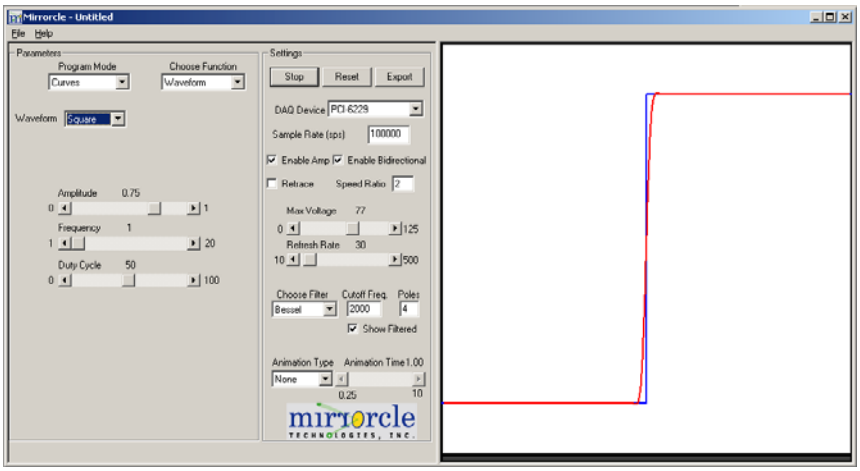


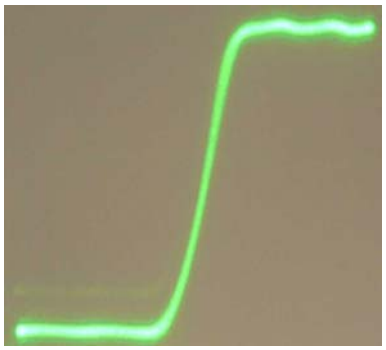
Curve Plotter Mode:

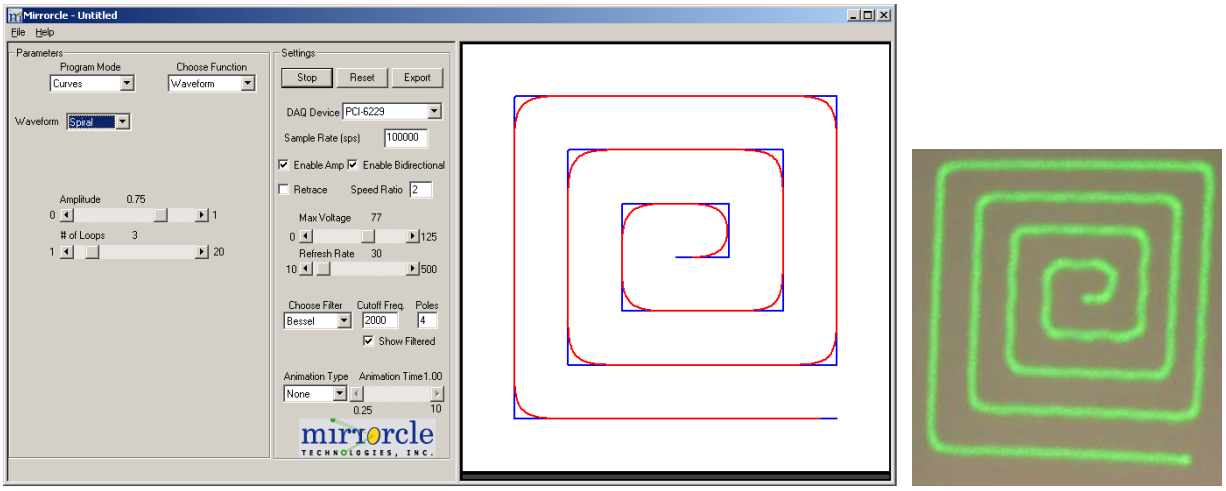
This mode demonstrates different types of mathematical curves, animations and some sample vector text. Clicking the mouse anywhere in the drawing area moves the center of the generated object to that location. This applies to all sub-modes and is useful for translating objects.

1) **Waveforms:** This function is used to draw simple waveforms such as sine, triangle and square. The user can choose the number of cycles displayed and the amplitude of the waveform. For the sawtooth and square waveforms, the duty cycle can also be adjusted. A useful exercise in this mode is to check the step response of the device and the effect of various filter parameters. In order to do this, select the square waveform with a single cycle and choose between the Butterworth, Bessel, and Inverse Square filters and check out the response of the device. Finally, there is also a spiral option, which traces out a right-angled spiral with variable number of loops and is useful for checking the fidelity of the filters in tracing sharp corners. These figures can be rotated by choosing the rotate option in the animation menu at the bottom. The animation time slider can be used to fix the total time for one cycle of the animation. Normally an animation time of 1-2 sec gives visually pleasing results. Increasing the animation time slows the rotations but takes longer to compute. Once the animation frames are computed, the program loops the animation infinitely.

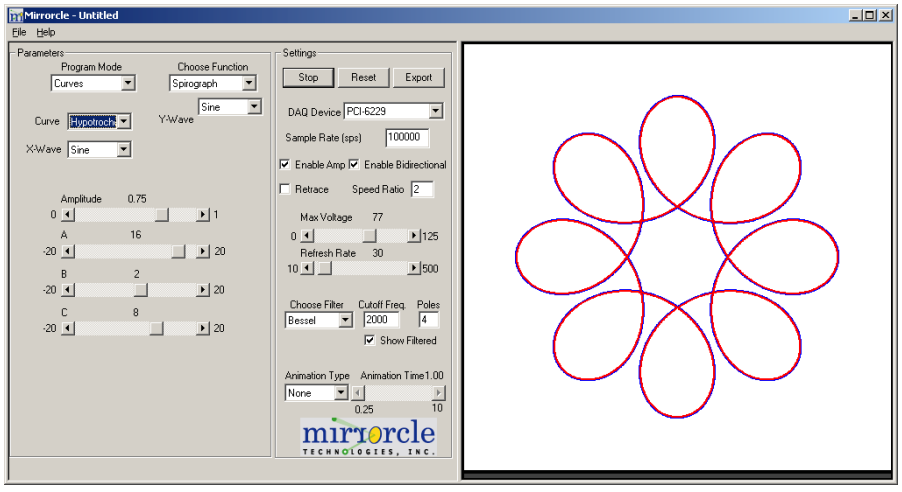


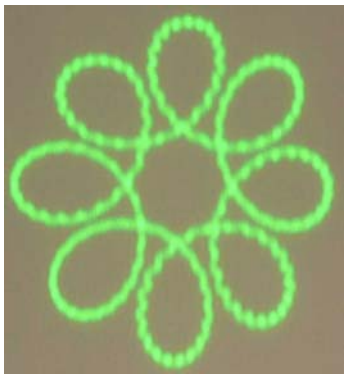


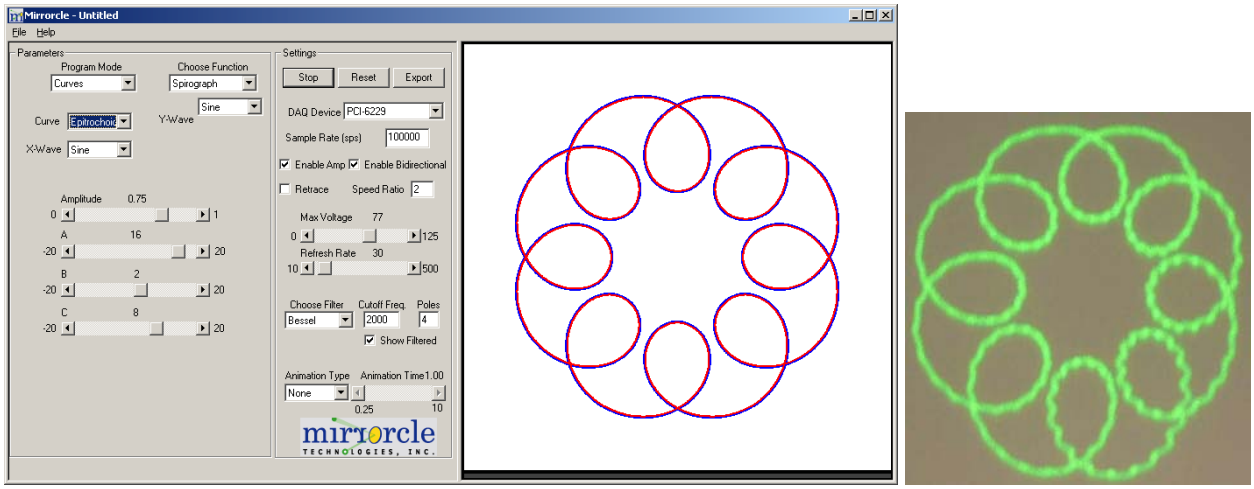


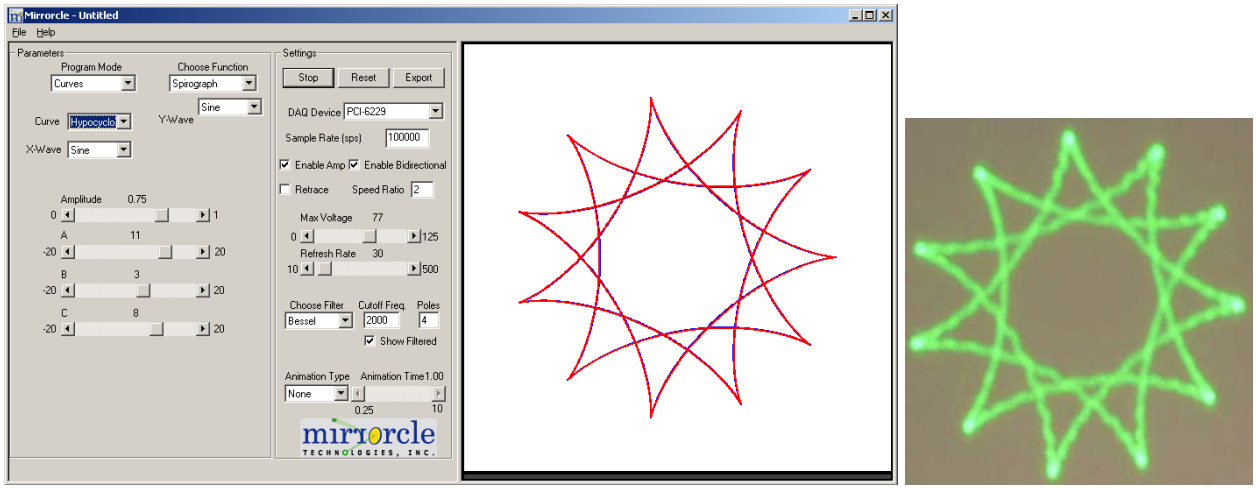


2) Spirographs: These patterns are formed by tracing the paths of points on a circle as it revolves around another circle. Depending on the relative radii of the circles and the position of the point, different curves are obtained. The program allows the user to choose between epitrochoids, hypotrochoids, epicycloids, and hypocycloids. The user can specify the parameters A, B, C that determine the shape of the curves and the size of the drawing. In this mode it is possible to have two main types of animations: in the first case, the entire curve is rotated at the rate specified by the animation time slider. The second animation option allows the user to cycle through different values of A, B, or C giving rise to a sequence of curves. The program calculates sweeps the chosen parameter (A/B/C) from the current setting through 6 values. Another curve is the rose family, which generates a set of roses with n-petals. The curve family can be generated by varying two parameters A & B. For B=1, a rose with n=A petals is obtained for odd A and 2A petals for even A. Changing B gives rise to more intricate and complicated patterns. The user can also obtain rotating animations by choosing the rotate animation option. Some sample screenshots and suggested parameter values are shown below.

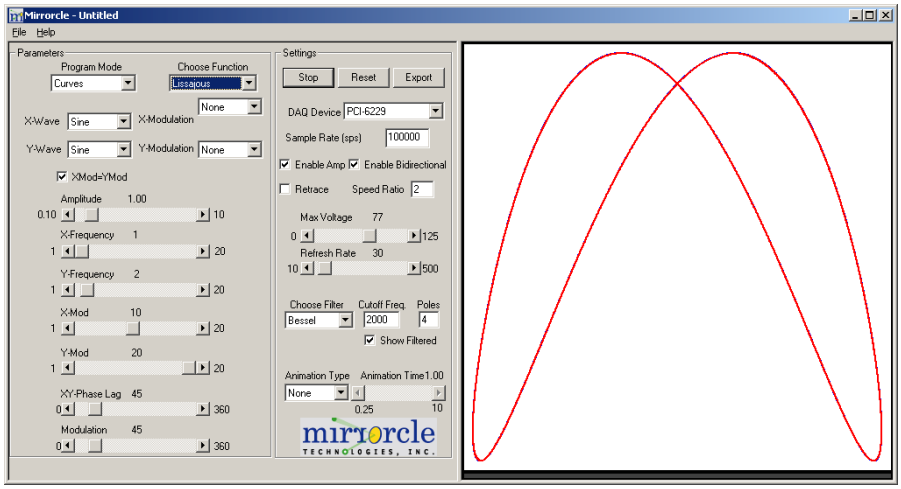


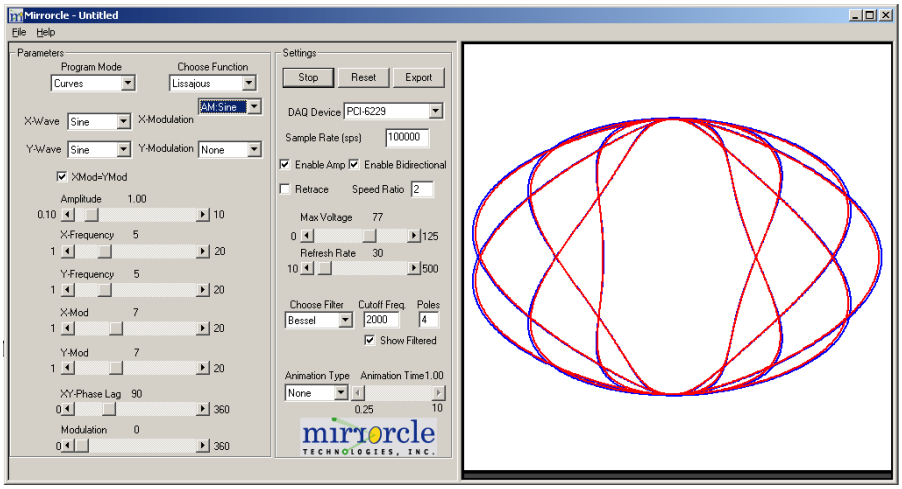




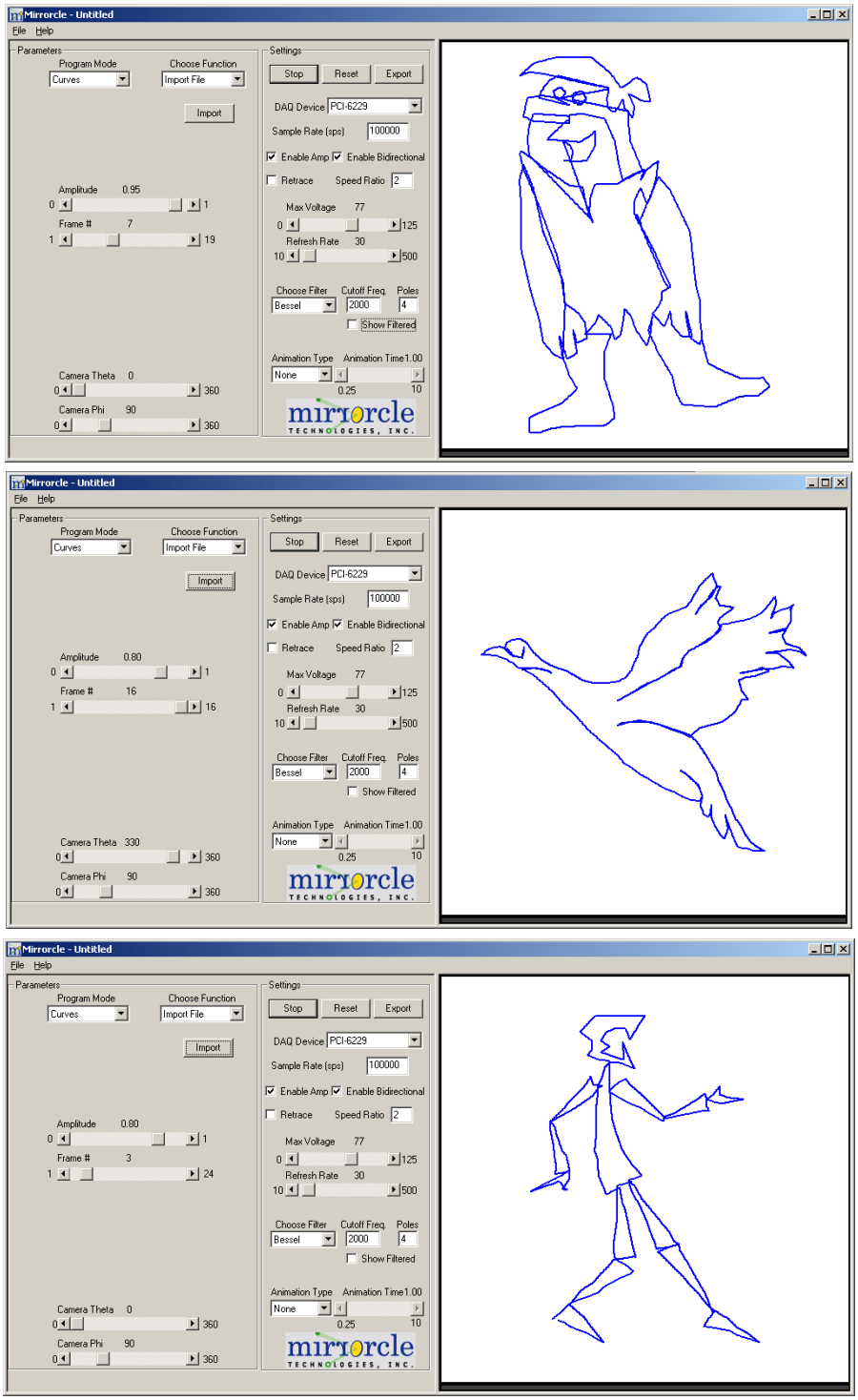


3) Lissajous: Lissajous patterns are generated by using a set of basis waveforms (sinusoidal / sawtooth) on the X and Y axes and are commonly seen on oscilloscopes. Changing the relative frequencies on either axis gives rise to a whole array of patterns. In addition to the frequency, it is also possible to change the phase and amplitude of the signal on either axis. A more complicated set of patterns is obtained by allowing modulation of the waveform on either or both axes. The user can select amplitude or frequency modulation for each axis and also the type of waveform used for modulation. The modulation index, which is a measure of the strength of frequency modulation, can be chosen by the user. As the modulation index is increased, the curve starts to get more complicated and deviates from its non modulated version. The user can choose to animate rotation of the entire curve as in previous modes or animate the X-Y phase lag. The phase lag animation sweeps the phase through 360^{o}, resulting in an appearance of rotation of the curve about its long axis. Similarly, the modulation phase, which is the phase angle between the base and modulation signals can be swept to generate another type of animation. Finally, the modulation ratio can also be swept in order to see the effect of increasing modulation strengths on the base signal.





4) **Import File (ILDA):** The International Laser Display Association (ILDA) has a standard file format for laser graphics that is used in many laser show design softwares. The ild file format is a binary format and can contain multiple frames that are part of an animation sequence. A number of sample ILDA files are included with the program. To load a new ILDA file, press the import file button and choose the file from the file dialog box. The program parses the file to extract all the frame information and displays the total number of frames. The user can display a single static frame by using the frame number slider control. Alternatively, it is possible to run all the frames as part of an animation by setting the animation type to Slideshow. As in previous modes, it is possible to create a rotation animation of a single frame by choosing the Rotate mode as the animation type. The animation time determines the time to display all the frames. If the time is too short, each frame will have few points resulting in loss of detail or smoothing applied by the filters. The other factor in displaying animations is the refresh rate. Too high refresh rates again decrease the number of points available per frame leading to loss of details. Hence it is important to keep a sufficiently large animation time and low enough refresh rate for best results. Finally, each frame can be rotated in 3D by choosing the camera theta (angle with +X axis) and phi (angle from XY plane). This can lead to cool effects in viewing frames containing 3D objects. A good starting file is cangoose.ild. Load the file and select Slideshow animation and set animation time to 2 sec. To make the bird fly faster, decrease animation time to 1s, 0.25 sec in steps.



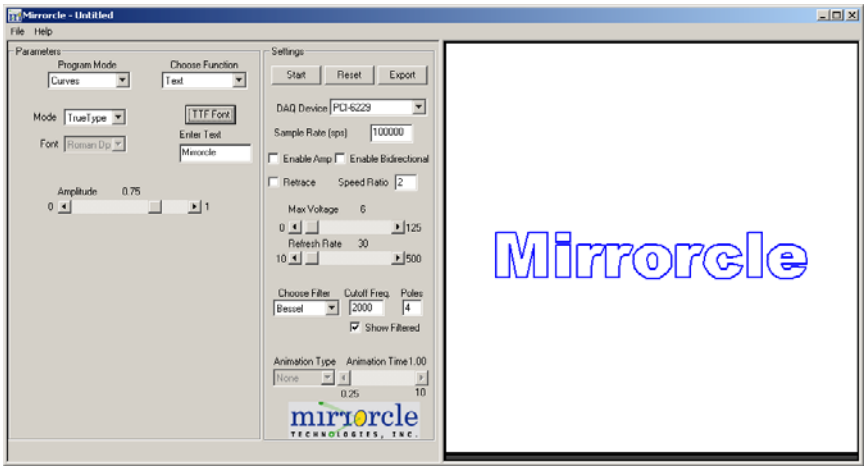
5) **Import File (ASCII File):** Another option is to import a text file with XYM coordinates. This file presents MirrorcleDraw with a list of keypoints which the mirror should move between. The data should be formed into three space-delineated columns as shown in the example below (a sample file, input_waveform.txt is included in the *exe* directory of the software installation.) The third column is the laser modulation (M) or blanking data, 1 for on trace and 0 for off trace. All coordinates are normalized to 0-1 (or -1 to 1 in bi-directional mode) inside the program, and then scaleable with the amplitude setting in the GUI.

```
0.51231     0.85026     1.00000
0.51163     0.85054     1.00000
```

```
0.51098     0.85083     1.00000
0.51035     0.85114     0.00000
0.50975     0.85144     0.00000
```

6) Text: This mode allows the user to enter a text string and uses either a vector or TrueType font in order to display the text. We use the Hershey font set, which contains a list of 22 English fonts and a variety of Oriental characters for vector text and standard TrueType fonts. There is also a character display mode which can be used to display mathematical symbols, ASCII characters, Kanji characters and various other symbols defined in the Hershey database. The user can select the character ID for the particular type of font and the program displays the font. If the text string is too long, it is possible to scroll the text across. In order to scroll the text string, check the Marquee checkbox. Note that it can take a long time to compute the marquee waveform.

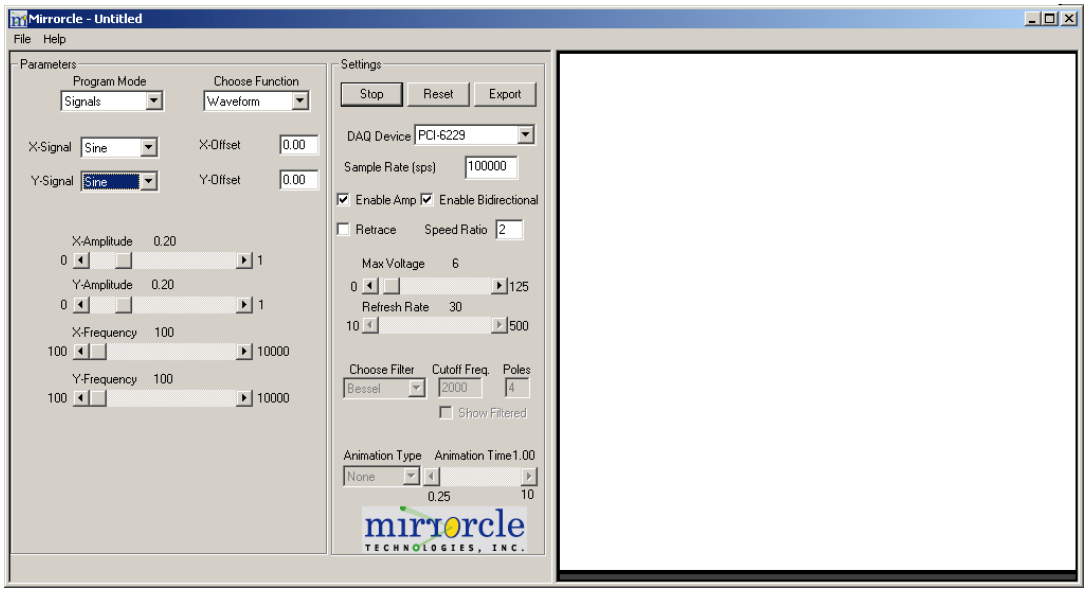
It is also possible to display TrueType fonts. The TTF Font button can be used to select any TTF font currently installed on your computer. The user can also choose whether characters will be Bold, Italics and the Encoding system used (Western European, Cyrillic, Central European etc.). This mode utilizes Windows GDI functions in order to get vectorized a string or character in any system font and displays the vectorized text. Currently, the program does not support Unicode text but it is possible to input various non-ascii symbols for Oriental and other fonts by using 'Alt-xxxx' where xxxx is the Unicode ID for the particular character in the chosen font (e.g. Choose the MS Mincho Font and Japanese Encoding in the TTF Font dialog and enter Alt-0211 in the text to display a certain Hiragana character – try out different codes or use the Windows Character Map utility to find the code for a particular character).



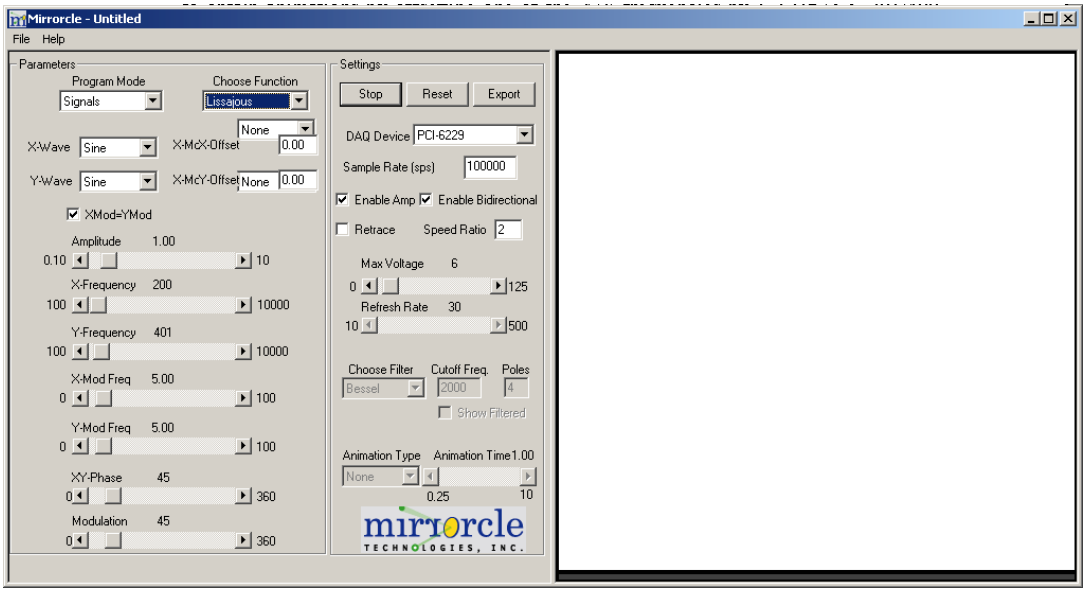
Function Generator:

This mode uses the sound card or DAQ as an analog output to directly send synthesized waveforms to the mirror. Unlike in the curve plotter mode, there is no modulation scheme used to transmit the signal to the mirror. Hence this mode must be used carefully since it is possible to excite the resonant frequencies of the device and damage it.

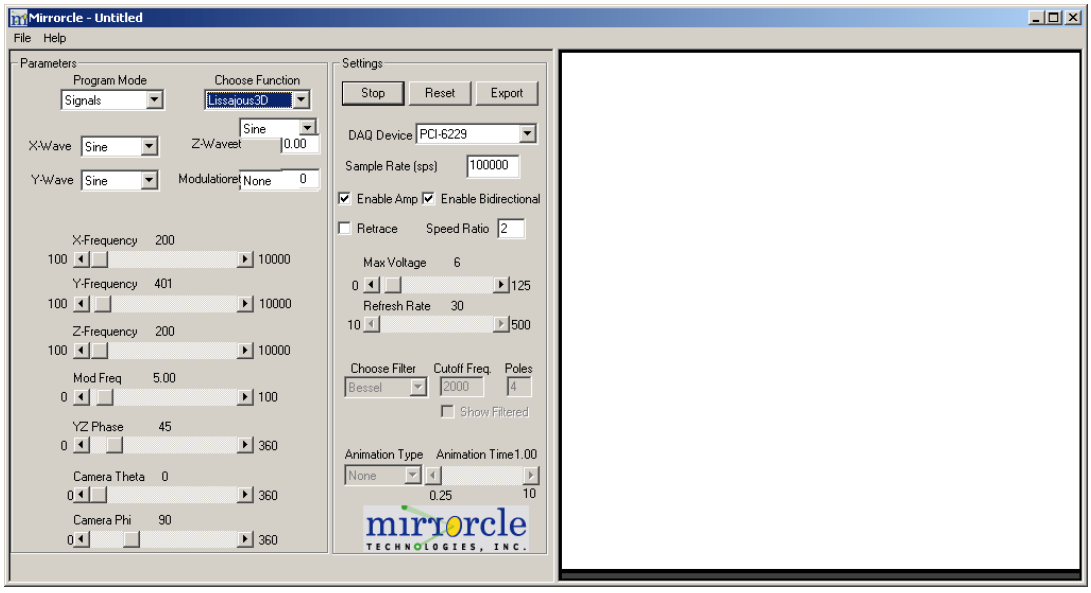
1) **Waveforms:** This mode can be used to send a set of signal waveforms on either or both channels. The user can choose between DC, sinusoidal, sawtooth, or noise waveforms. The DC mode allows the user to input any voltage level to either axis. For the sinusoidal and sawtooth waveforms, the user can choose the frequency and this can be used to find the resonant frequency of the device. This is simply done by choosing sine waves of different frequencies and observing the mirrors deflection peak at the resonant frequency, which should be visibly obvious. It is safest to keep the voltage level low since the devices have high Q and can be damaged by exciting them resonantly with excess voltage. The resonant frequency can then be used to refine the system parameters used to calculate the Inverse Square filter. Q values from 10-100 are reasonable. Another point to note is that the actual resonant frequency is twice the value on the slider since the mirror responds to the square of the voltage waveform. The noise waveform is useful for checking if the devices are functional and can occasionally be used to revive devices that may be stuck or crashed.



2) **Lissajous 2D:** This is similar to the function in the curve plotter mode but there is no modulation used in this case. The user can directly choose the base and modulation frequencies on both axes and the types of waveforms desired. Here, it is relatively simple to obtain animations by offsetting one of the X/Y frequencies by 1-2 Hz (e.g. 201/400 Hz). Another interesting set of animations is obtained near resonance. Make both X, Y base frequencies near resonance so that a large mirror amplitude is obtained. Now change the modulation frequencies to greater than 30-35 Hz so that persistence of vision gives rise to an intricate set of animations. Changing the modulation frequency to a non-integer number slows the animation and also gives more aesthetic patterns. It is possible to get a virtually limitless set of patterns using the four sets of frequency parameters.



3) **Lissajous 3D:** This is similar to the 2D function except that base waves are used on 3 axes to get a three-dimensional Lissajous figure. This curve is then projected onto an arbitrary plane in space, which is specified by choosing values for the camera theta and camera phi values. The angle theta is the rotation of the camera about the Z axis in the XY plane and phi is the angle of the camera with the positive Z axis (X-Y plane is theta=0, phi=90). Hence the user has an added dimension of flexibility in choosing the viewpoint in space and this enhances the range of possible curves and animations that can be obtained. As in the 3D case, it is possible to do modulation and also operate the mirror at resonance to get larger and more intricate images.



Settings Files:

Windows-executable versions of the software for USB and DAQmx interfaces include an ASCII text settings file which is located in the same "exe" directory as the executables. File name for the DAQmx version is:

mirrorcledraw_daqmx_settings.txt

and for the USB version is:

mirrorcledraw_usb_settings.txt

These files contain important settings that are required by the programs to run properly. The settings are purposely placed into these txt files so that they can be responsibly modified by an advanced user of the kit. The values that are entered into the settings will *directly* affect the voltages that are applied to the device and can therefore easily damage a device if improperly entered. Please make sure that you enter proper values and verify the system operation (oscilloscope, volt-meter) before operating devices.

Vmax	(float) This is the value of voltage which will be maximum allowed voltage during software operation. It is recommended to keep this value at the maximum voltage for a specific device to minimize risk of exceeding that voltage during operation. Default is 125. Maximum possible value is 150V, limited by hardware design.
Kamp	(float) Value of the amplifier gain. This value is obtained by measurement output/input before the hardware is shipped. Wrong value in this line could cause unexpected high voltages and damage devices. Default value for DAQmx kit amplifier is 14.9. Default value for USB kit amplifier is 60.
BDO Enable	(int) This value is either 0 or 1. If you want to drive bi-directional devices and have a bi-directional device kit with a dual-supply amplifier (-150V to 150V,) this value should be on. This is ONLY the case for such specific DAQ-based kits. In standard kits with uni-directional devices and in all USB-based kit cases the value should be 0.
LUT Enable	(int) This value is either 0 or 1. Default value is 0 and it means that the software will not utilize look up table (LUT) information provided below in this file. Instead, the software will by default utilize SQRT function to linearize the device position in curve and sketch modes. This is not meaningful in signal mode which does not condition signals but outputs pure function generation signals. If this value is 1, the software will read the nonlinear fit information provided in lines that follow and apply functions below.
LUTX	(float) Four coefficients for a nonlinear fit to a particular device's voltage as a function of desired angle theta on the X-axis (normalized). Coefficients are: cs_x c0_x c1_x c2_x. Example to achieve linear LUT (equiv. to having no LUT,) is: LUTX 0 0 1 0
LUTY	(float) Four coefficients for a nonlinear fit to a particular device's voltage as a function of desired angle theta on the Y-axis (normalized). Coefficients are: cs_y c0_y c1_y c2_y. Example to achieve sqrt LUT is: LUTX 1 0 0 0 After processing user inputs, the software computes a set of angles that the device should address as a set of Tx,Ty. Then it will apply inverse look-up table function Vx=sqrt(Tx) and Vy=sqrt(Ty) if the LUT Enable is 0. However, if LUT Enable is 1, it will apply following functions utilizing coefficients in LUTX and LUTY lines. Vx= cs_x * sqrt(Tx) + c0_x + c1_x * Tx + c2_x * Tx*Tx Vy= cs_y * sqrt(Ty) + c0_y + c1_y * Ty + c2_y * Ty*Ty Coefficients are normalized in such a way that maximum angle in X and Y are normalized to 1 and maximum voltage should be normalized to Vmax (in this file,) because it should never be exceeded.

MirrorcleAO (Analog Output) – C++

MirrorcleAO 6.0 Provides an interface to send arbitrary data vectors to MirrorcleTech micromirror devices. This is effectively the Software Development Kit (SDK) of Mirrorcle Software Suite that allows users to develop their own applications. C++ DLL: This interface provides a library with wrapper classes for analog output using audio, USB, and/or NI-DAQ devices. An example Visual C++ project is provided to illustrate the use of the library classes.

For users with Visual Studio C++ .NET 2003 version, there is an example code which utilizes the provided libraries and functions. This example allows the user to select whether to output 2 analog channels to the audio port, or 2 analog channels to the USB-based development kit, or 2 analog channels and one digital channel (laser modulation) to the DAQ-based development kit.

Users can easily modify the C++ code and rebuild new executables. Note that when a new exe file is built by Visual Studio, it will be created in the "Debug" directory, and should be moved into the "exe" directory with all other Mirrorcle executables because the necessary dlls are there as well.

MirrorcleAO-DAQmx.cpp example:

This example C++ program demonstrates to the user three possible and different ways of actuating MirrorcleTech devices. When you execute the pre-compiled version of this program within the exe directory, you will be asked to run one of those three modes, i.e. "Point-to-Point" mode, "Scanning" mode, or "Import File" mode.

*Scanning mode example**:***

This example C++ program is designed to send a stream of co-ordinates to the micromirror device via NI DAQ output ports. It demonstrates the use of functions such as SendDataStream() to communicate to an analog/digital output port of the computer and eventually to the MirrorcleTech device. The user will be first prompted which DAQ device he or she uses, and which maximum voltage to scale the pattern of co-ordinates through. The .cpp file is fairly self-explanatory and commented to provide easier use and modification to user's desired application. For example, the user may choose to output the stream with infinite repetition, for example to create a repeating raster patter, or he or she may choose to output the stream in a single or multiple-cycle burst. This is done with the last "loop" parameter of the SendDataStream() function. It should be noted that in addition to creating X and Y arrays of data to send to the port, the user must also create a B array of laser blanking data or digital triggering data, depending on the application. This blanking data is streamed by correlated DIO (digital IO port on the NI-DAQ,) to port 0, which is not available on all NI-DAQ devices. This allows the digital output stream to coincide (correlate) with the X and Y analog voltage streams. This number should be an integer between 0 and 255, corresponding to assignments to lines 0 through 7 on Port 0. The example shows simple counting of a for loop being assigned to this port so the lines of port 0 will show digital counting.

Point-to-point example:

The point-to-point demonstrates one way to simplify the open-loop control of devices in an application where the user will need to point the laser beam to different sequential points and would like to have a smooth and fast transition between those points. This method uses pre-processed and normalized steps which are stored in files in that same directory. There are three such files, each one for a step of different duration, 0.25ms, 1ms, and 10ms. Each point in the file represents a 10us duration of time due to the present 100000 samples-per-second setting in the .cpp file. Therefore the 0.25ms file

contains a pre-processed (Bessel filtered step function) step of 25 points, each 10us for a total of 250us. The example will ask the user to input a set of normalized co-ordinates to send the device to. Values from 0 to 1 on each axis are valid unless you have a bi-directional device in which case values from -1 to 1 are valid inputs. When a new set of co-ordinates is entered, the program sends a step to the DAQ card which will send the device from Xold, Yold to Xnew, Ynew as follows:

OutputX = Xold + (Xnew-Xold) * Step1ms
OutputY = Yold + (Ynew-Yold) * Step1ms,

where e.g. step1ms is an array of 100 points representing the 1ms filtered step.

If a user enters co-ordinates outside of -1 to 1 limit, program quits.

In the current version the laser blanking b channel is given zeros during the step such that the laser which is modulated with this channel will be turned off in the transition region between the new and old co-ordinates. The laser is than turned on at the very last point of the transition by assigning b[last point]=1. The user should modify this as desired.

Import File example:

Another option is to import a text file with XYM coordinates and repeat the prescribed trajectory in the file infinitely until program is stopped. The data should be formed into three space-delineated columns as shown in the example below (a sample file, xym.txt is included in the *exe* directory of the MirrorcleAO software installation.) The first two columns are normalized trajectories from 0 to 1 for 1-quadrant devices or from -1 to 1 for 4-quadrant devices. These values will be multiplied by maximum voltage setting Vmax which is given in the settings file "mirrorcleao_settings.txt" The third column is the laser modulation (M) or blanking data, 1 for on trace and 0 for off trace.

```
0.51231     0.85026     1.00000
0.51163     0.85054     1.00000
0.51098     0.85083     1.00000
0.51035     0.85114     0.00000
0.50975     0.85144     0.00000
```

Before putting out the voltages, the program will also ask the user for the samples-per-second (SPS) rate. This rate will establish the amount of time between each row being output. So for example, SPS=1000 will have each row output at 1/SPS = 1 ms separation in time. User should be very careful to combine proper trajectories with proper SPS setting so as not to exceed mirror devices' speed capabilities and cause ringing. For the stored example in xym.txt, SPS values from 1000 to 10,000 should be experimented with.

Software Requirements:

The provided dynamic link libraries (DLLs) in Mirrorcle AO are general Windows32 DLLs that can be used in a variety of development environments.

For users with the DAQ-based amplifier, National Instruments MX drivers must be installed.

In order to run the provided examples developed in Visual Studio C++, Microsoft Visual Studio .NET 2003 should be installed.

MirrorcleAO (Analog Output) – Matlab

MirrorcleDraw Matlab-based Software Development Kit. Similar to MirrorcleAO Cpp version, this is a Matlab-based software development kit (SDK) allowing the user the fastest and easiest route to development of micromirror applications.

Matlab-based MirrorcleDraw with a wider range of signal conditioning and signal generation possibilities. For users with Matlab's Filter design toolbox, it is possible to integrate any type of filter in the micromirror demonstration. Spline toolbox will allow the polyline drawing mode to create spline curved vector graphics.

Most of the features in MirrorcleDraw are same as in the executable Cpp-based version described in earlier sections. However, user should note that there is no Correlated-DIO capability in this version, and therefore the laser-modulation digital channel cannot be used as in the executable version.

However, for users with a National Instruments DAQ card which has more than 2 analog outputs, MirrorcleDraw-Matlab will automatically assign the 3rd analog output channel as the modulation output and will provide +5V for laser-on features and 0V for laser off features of the vector drawings.

Matlab-based MirrorcleAO SDK with Matlab functions for quick development of analog output applications. User can use the existing functions to identify an analog output device such as a DAQ card, to create excitation signals and to send them to the output device. The user has to prescribe the excitation signals sample by sample, by forming vectors for the x axis and the y axis.

Software Requirements:

Matlab-based MirrorcleDraw Requires Matlab R14 or later with the Data Acquisition Toolbox and the Signal Processing Toolbox. In addition, Filter Design Toolbox and Spline Toolbox provide additional features but are not required.

Matlab-based MirrorcleAO functions require Matlab R14 or later with the Data Acquisition Toolbox.

MirrorcleAO (Analog Output) – LabView

LabView-based Software Development Kit. Similar to MirrorcleAO Cpp version, this is a LabView-based software development kit (SDK) allowing the user the fastest and easiest route to development of micromirror applications. There are several example vi-s provided here to demonstrate driving of devices through waveform-based signal generation, array-based signal generation, and file importing.

Software Requirements:

LabView-based examples require LabView 8.5 or later.