VirtualCMM Users Guide

Table of Contents

Purpose....3
User Interface....3
Machine Dock....4
Protocol....5
Communication Setup Example....6
Machine Errors....8
ASME B89.4.1 Ballbar With Machine Errors Enabled....8
Override Tip....9
Touch Noise....9
Move Position....9
Jogbox....9
Path Buffer....11
Touch Points Dock....12
Free Points....12
Ballbar Artifact....14
Gaugeblock Artifact....15
Ringgauge Artifact....16
Sphere Artifact....16
Stepgauge Artifact....17
Information Display....17
Message Log....18
Communication....18
Display Readout....18
Protocol Limitations....19
Supported Commands....19
Synchronization of Commands....19
I++ DME Tools....20
Stepgauge 10360-2 Measurement Example....20

Purpose

VirtualCMM is software that emulates the functions of a physical CMM. It was written primarily for testing the measurement routines of *MeasureDirect* when an actual CMM is not available. The virtual machine is animated and can display the recent probe path to help identify problems such as missing moves or inverted touch points.

The *VirtualCMM* includes a functional jogbox and indexable probe head. The default communication protocol used by *VirtualCMM* is a subset of Leitz but DC and I++ Server is also available. It was decided to use the Leitz protocol for communication as a default option for this utility as this may be beneficial for troubleshooting future problems.

User Interface

The user interface of *VirtualCMM* is shown in illustration 1 showing the Touch Points and Machine dock widgets on the left side and the jogbox on the right.

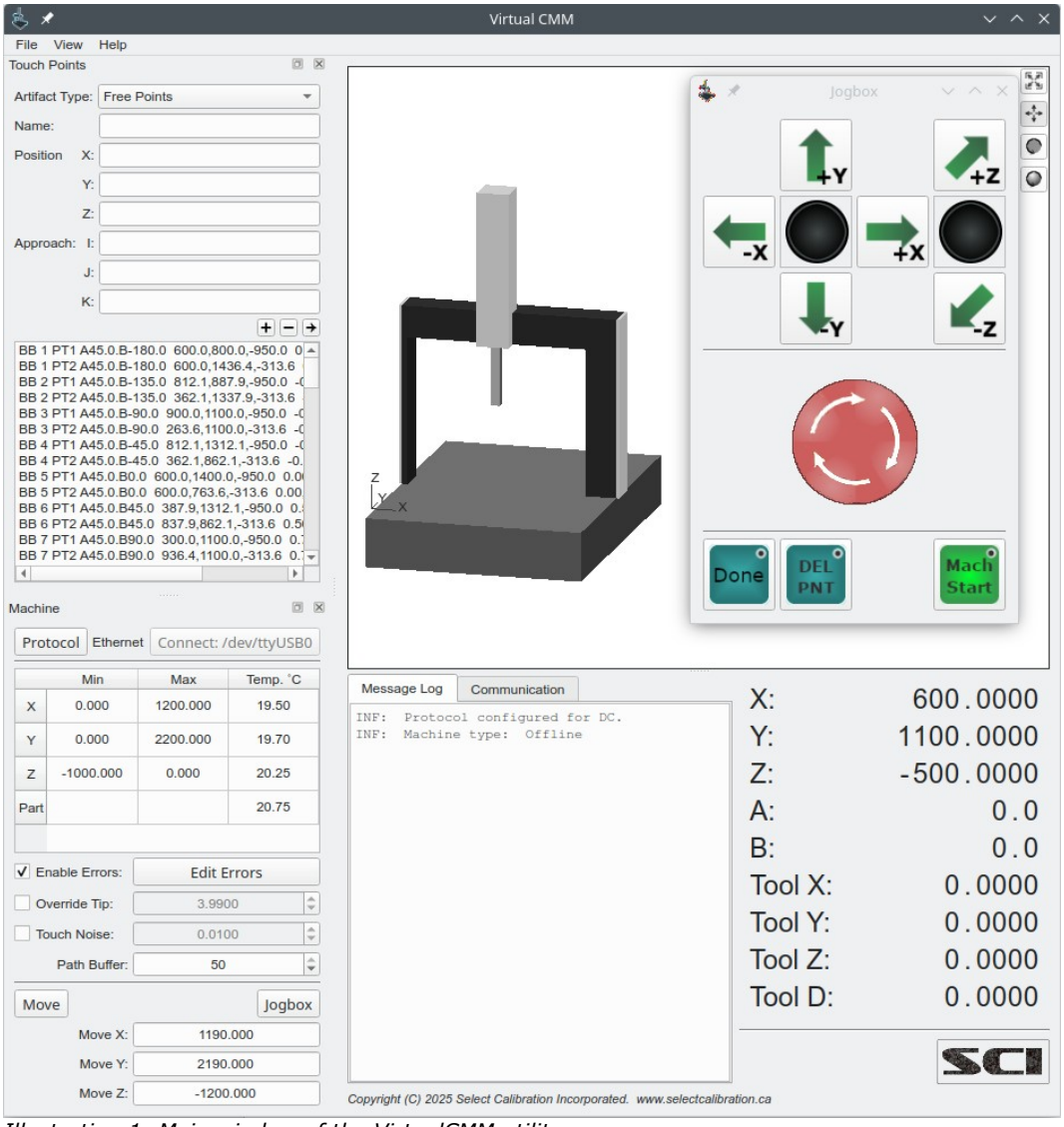


Illustration 1: Main window of the VirtualCMM utility.

Machine Dock

The *Machine Dock* allows configuration of the CMM and method for communication. Illustration 2 shows the machine dock setup for a CMM that has measurement strokes of 1200, 2200, and 1000 mm for the X, Y, and Z axis and configured to use serial communications through comm port */dev/ttyUSB0*.

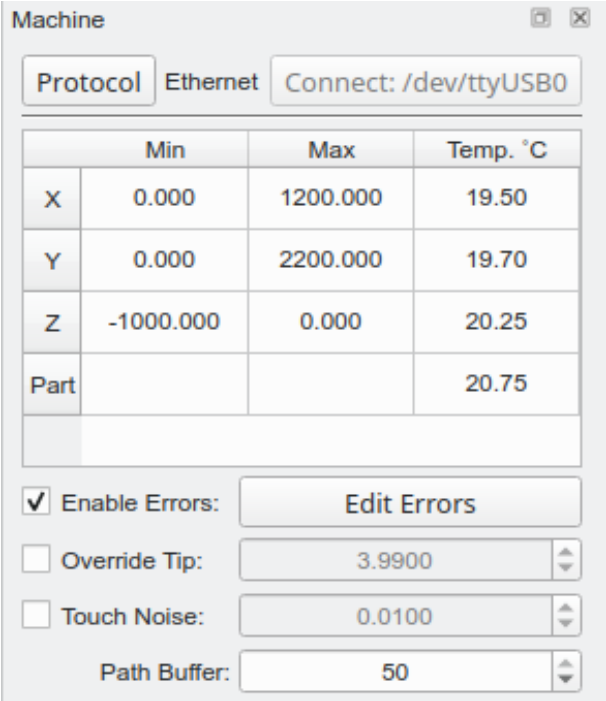


Illustration 2: Machine dock.

Table 1: Machine dock options.

Option	*Description*
Protocol	Setup for the method VirtualCMM will use for communication. See protocol section below for details.
Connect: <name>	When configured for serial clicking this button will connect or disconnect from the named serial port.
Min / Max XYZ	Limits for dimensions of the Virtual CMM.
Temp °C	Temperature inputs for the X, Y, Z, and Part on the simulated CMM.
Machine Errors	Allows simulated machine errors to be added to all measurements. See machine errors section below for details.
Override Tip	Allows the use of a tip diameter that is different from what the inspection software defines.
Touch Noise	Add a random amount of measurement noise to each touch point.
Path Buffer	Size of the buffer containing past measurement points. The data from the buffer is displayed graphically showing touch directions and clearance moves.
Move	Drive the *VirtualCMM* to the specified XYZ coordinate.
Jogbox	Show or hide the virtual jogbox.

Protocol

Clicking the Protocol button will open the communication options dialog as shown in illustration 3

allowing configuration of the controller type and connection method. *VirtualCMM* supports communication using serial or Ethernet.

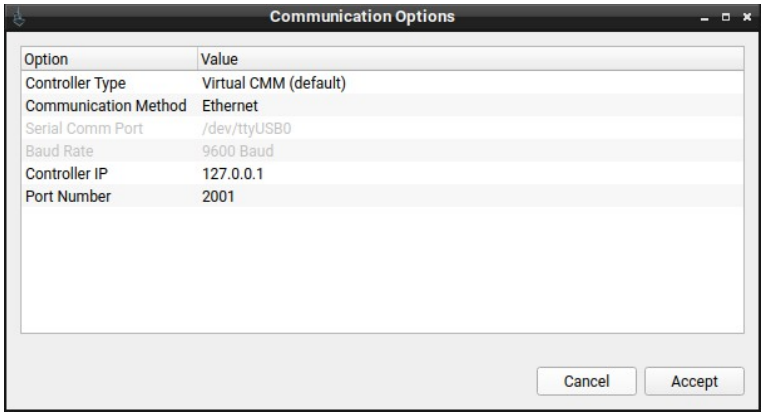


Illustration 3: Communication protocol setup.

Table 2: Protocol options.

Option	*Description*
Controller Type	Type of controller that VirtualCMM will emulate.
Communication Method	Selection of serial or Ethernet.
Serial Comm Port	For serial communications the name of the comm port.
Baud Rate	For serial communications the serial baud rate.
Controller IP	For Ethernet communications the IP address of the connection.
Port Number	For Ethernet communications the IP port address.

When the communication method is set to Ethernet the option to connect using serial is disabled. The name of the serial port is shown in the text of the serial connect button. The default serial port name is based on the operating system and can be changed from the communication protocol setup dialog as shown in illustration 3.

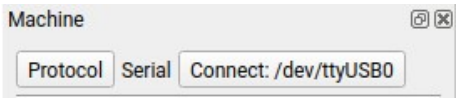


Illustration 4: Serial comm port opened when 'Connect: /dev/ttyUSB0' is clicked.

Communication Setup Example

The following is an example of connecting *MeasureDirect* to the *VirtualCMM* using the DC protocol for communication. Since both utilities will be run on the same computer Ethernet is the easiest option to configure. The *VirtualCMM* controller IP address is set to 127.0.0.1 (localhost) with port number 1234.

Any valid, non-reserved, value can be used for the port number provided both ends are setup the same way.

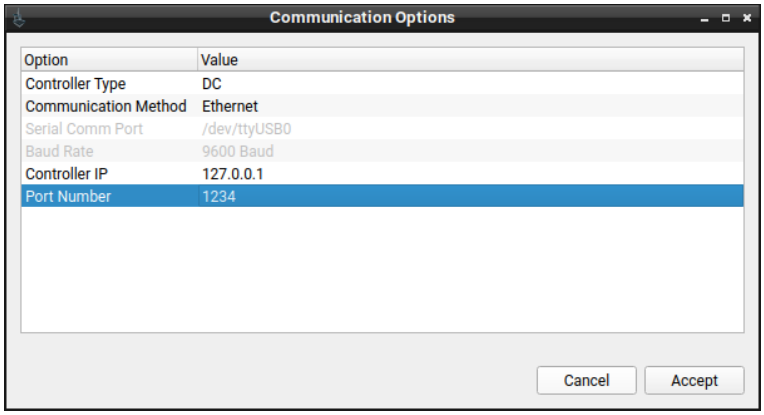


Illustration 5: VirtualCMM protocol setup. The IP address is set to 127.0.0.1 and listening on port 1234 for new connections.

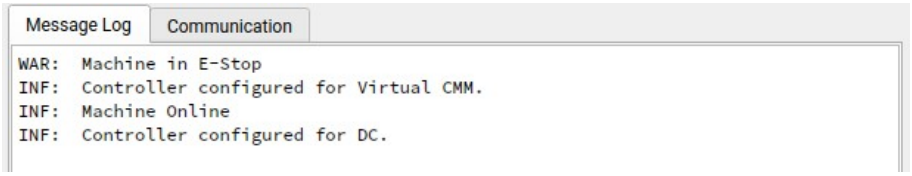


Illustration 6: VirtualCMM information messages showing status changes.

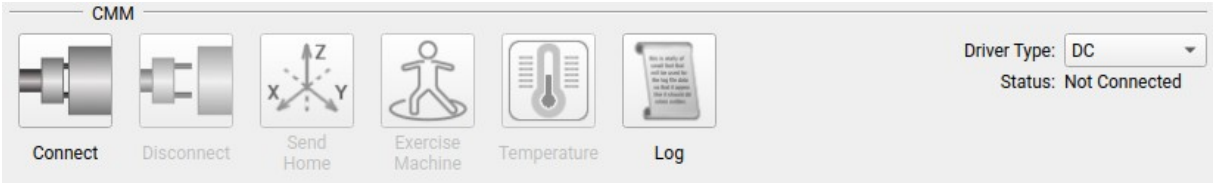


Illustration 7: MeasureDirect CMM configuration prior to clicking Connect. Driver type is set to DC.

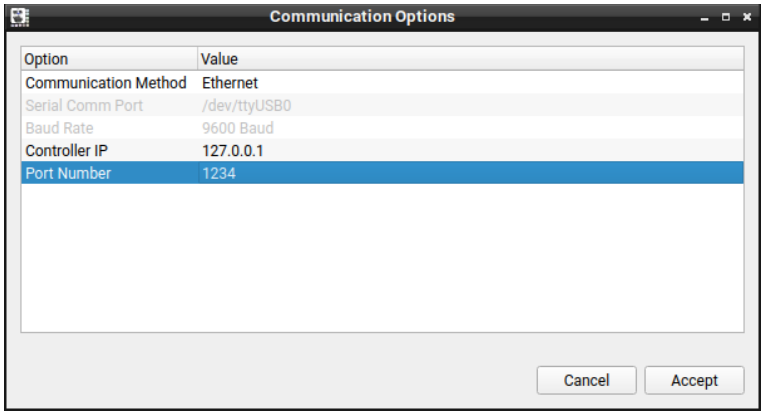


Illustration 8: MeasureDirect connection options that appear when clicking Connect. IP Address is 127.0,0.1 and the port number is set to 1234 matching those of VirtualCMM.

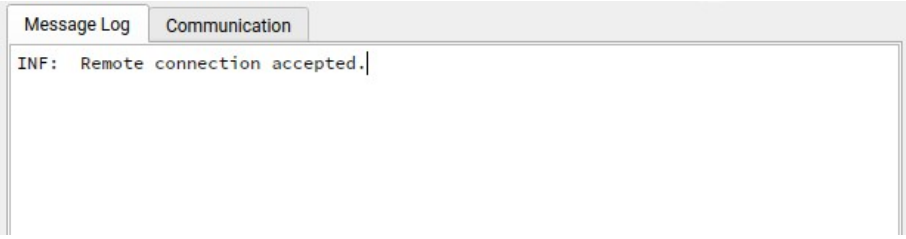


Illustration 9: VirtualCMM message showing the connection is accepted.

Any supported connection method will work provided both ends are setup in a compatible way. Ethernet is the preferred connection method when running both software on the same computer but serial can be used as an alternate method.

Using serial on the same computer will require a loop-back serial cable or equivalent software such as socat to create the virtual serial ports.

Machine Errors

The machine errors option will introduce a simulated scale and squareness error to the simulated CMM.

ASME B89.4.1 Ballbar With Machine Errors Enabled

The machine errors when using an ASME B89.4.1:1997 ballbar will not appear exactly as entered. This is due to how a typical inspection program works when measuring a ballbar. A typical part program will involve the following steps:

1) Manual measurement of a point on each sphere to establish a rough location.
2) DCC measurement of the sphere using a minimal number of points to establish an approximate sphere location. This measurement is based on the manual point taken in step 1.
3) DCC measurement of the sphere using the nominal number of measurement points and preferred measurement pattern. This measurement is based on the approximate sphere location from step 2.

For each step listed above the machine errors are cumulative so without any kind of adjustment the effects of the reported error will be approximately three times larger than what would be expected when using a physical gauge. Effects such as sphere roundness will also contribute to the error but to a lesser extent. One solution is to have a nominal ballbar length entry in the inspection software and have all points measured based on this but this is rare as a typical ballbar does not have a nominal length.

To bring the measured results inline with expected results an option was added when touching the initial manual points of the ballbar spheres in *VirtualCMM* to remove twice the expected error so, in theory, the end result should be close to the expected values. This option is enabled by default can be be disabled if required.

This only affects ASME B89.4.1 ballbar measurements. If proper ballbar data is required that reflect precise machine error effects then the Error Simulator utility should be used instead of VirtualCMM. The final results using VirtualCMM will be close but not exact for an ASME B89.4.1 ballbar measurement.

The problems affecting ASME B89.4.1 ballbar measurements do not affect other length measurement types since the inspection program will target features at nominal locations. For example, if the 10360-2 ballbar measurement is run the results will be exactly as expected as the inspection software targets features based on the nominal lengths and not from prior feature measurements.

Override Tip

The diameter of the active tip is set by the inspection software with an appropriate command but this diameter can be overridden with a user defined value if the option *Override Tip* is checked.

When a tip diameter is used that is different from that of the inspection software a simulated measurement error is introduced and can be useful in some scenarios.

Touch Noise

When measuring points the error can be randomly introduced by enabling this feature and setting a suitable maximum error value. The purpose of this feature is to allow measurement of features with varying levels of error instead of measuring perfect features.

The noise value represents the largest potential error that could be generated for each touch point. The error is generated along the surface normal vector of the touch point.

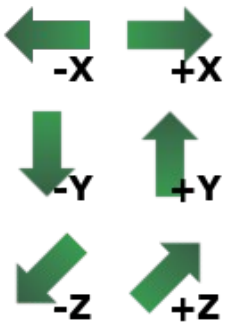
Move Position

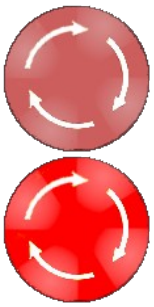
The machine can be driven to an exact location by entering coordinate values in the *Move XYZ* fields then clicking on the *Move* button. If the target position is outside of the measurement volume of the CMM the motion of the machine will stop at the edge of the volume.

Jogbox

The simulated jogbox allows the operator to manually drive the machine within the CMM volume. The visibility of the jogbox can be toggled by pressing the *Jogbox* button in the *Machine dock*.

Table 3: Jogbox functions.

Button	*Description*
 Directional Arrows	Allows manually moving the CMM in the indicated axis by clicking on the associated button. The move distance and speed is fixed. *It is not possible to position the CMM to an exact location using only the jogbox. It is only intended to move the machine by short hops in specified axis directions.*

Button	*Description*
	Emergency Stop button. When an emergency is present the E-Stop button will be shown in bright red otherwise dull red. The emergency is cleared by pressing the Machine Start button.
	Machine start button. When Machine Start is clicked the button will glow green and the CMM will no longer be in an emergency state.
	Done button. Function depends on the inspection software.
	Erase hit. Function depends on the inspection software.

When driving the *VirtualCMM* an error is returned to the inspection software if the machine is asked to move outside of the measurement volume by the inspection software. The format of the message mimics the format of the active communication protocol. An example of this kind of error is shown in illustration 10.

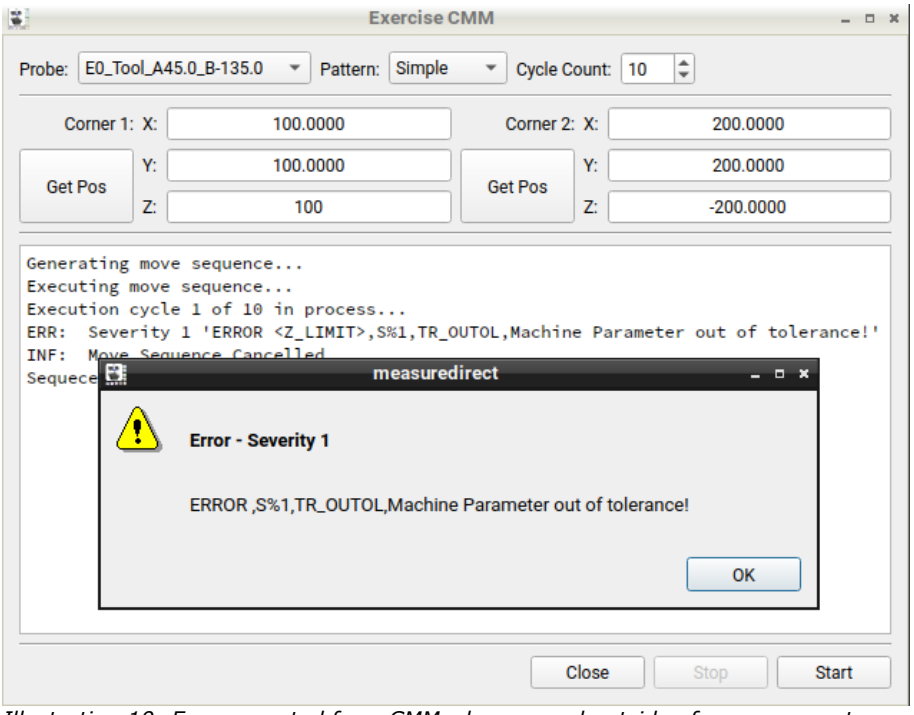


Illustration 10: Error reported from CMM when moved outside of measurement volume.

Path Buffer

The path buffer option allows the motion of the probe and all touch points to be visible when the *VirtualCMM* is driven by the inspection software. The path is useful to see relative location and direction of previous move and touch points. The number of path points displayed is controlled by the *Path Buffer* setting and can be disabled by setting this entry to zero.

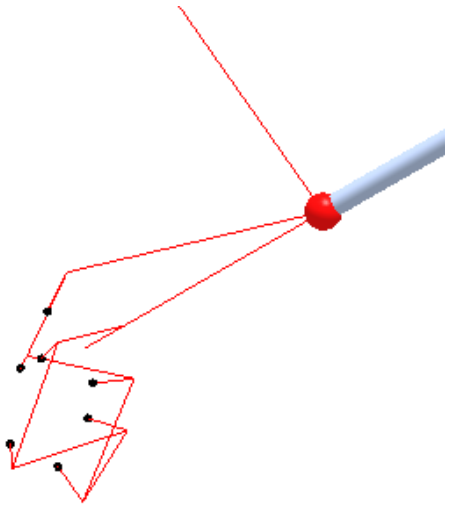


Illustration 11: Measurement of a sphere with the path buffer active.

Touch Points Dock

The *Touch Points dock* is used to create simulated measurement points that can be sent to the inspection software. Starting with version 7 of the *VitualCMM* utility touch points specific to certain types of artifacts can be sent instead of relying on free points. Unlike the free points version the individual touch points are calculated automatically based on the location, orientation, and artifact traits and can be sent to the inspection software by double-clicking any of the artifact points.

Free Points

Free points are individual touches that can be used for any purpose of feature measurement. The naming convention is open and customizable by the end user.

To send a free touch point to the inspection software double click on any of the points shown in the list or enter suitable values in the XYZIJK fields then press the send (->) button. New points can be added to the existing list of touch points by clicking the add (+) button. Selected points can be deleted by pressing on the delete (-) button.

The touch point file is located in the folder '.virtualcmm' and the location of this folder depends on the operating system.

Starting with VirtualCMM version 7.0 automatic point generation has been removed. Existing points are not deleted but a default set of test points for various artifacts are no longer created.

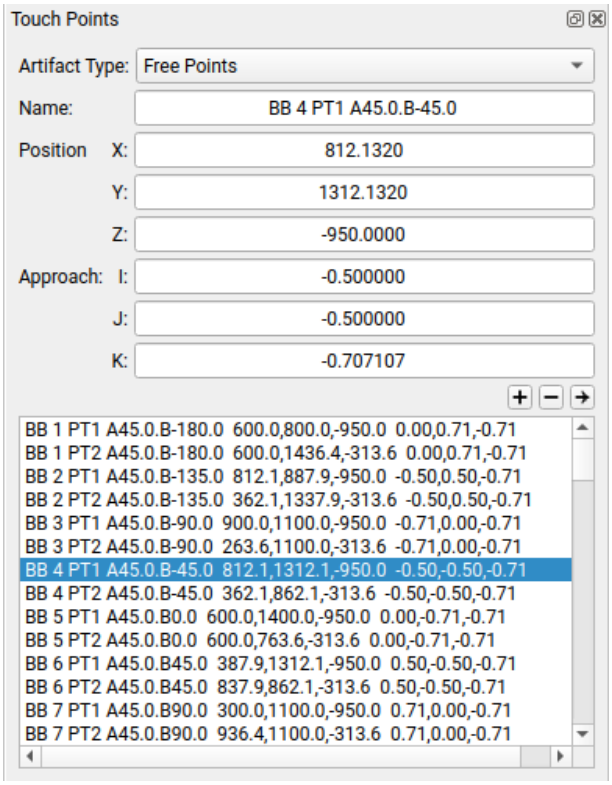


Illustration 12: Touch Points dock.

The naming convention of the legacy free point data for the different artifact types is based on the following guidelines:

<Identification> <Group> <Feature Point> <AB Angles> <XYZIJK>

Table 4: Legacy free point naming convention

Identification	*Description*
BB	Ball Bar
GB1	Gauge block measurement. Alignment using plane and probe axis.
GB2	Gauge block measurement. Alignment using plane, line, and point.
PG	Pin Gauge
RG	Ring Gauge
SG B89	Step Gauge. Measurement following ASME B89.4.1:1997
SG ISO	Step Gauge. Measurement following ASME B89.4.10360.2:2008 or ISO 10360-2:2009

Feature grouping is done using a letter or number. When a letter is used it will be X, Y, or Z representing a 1D measurement in X, Y, or Z axis. Numerical entries will be either 2D or 3D in various orientations of the machine volume.

Feature Point	*Description*
PT<n>	Measurement of a point.
PLN<n>	Measurement for one of the points of a plane. There will be a minimum of three points for this feature; PLN1, PLN2, and PLN3.
LN<n>	Measurement for one of the points of a line. There will be a minimum of two points for this feature; LN1 and LN2.
CIR<n>	Measurement for one of the points of a circle. There will be a minimum of three points for this feature; CIR1, CIR2, and CIR3.

The AB Angle entry describes the ideal probe head orientation to measure the specified artifact. For example, if the AB Angle entry is *A90.0.B90.0* then the ideal probe head orientation for this measurement will be A = 90.0 and B = 90.0.

The XYZIJK entry contains the location and approach direction for the touch point. The XYZ touch shown in the point list is compensated (not at stylus center) and the IJK values are the approach normal direction.

The touch point data sent to the inspection software is adjusted to match the type of controller. The XYZ point will be offset to the center of the probe stylus for all controller types. The IJK vector is inverted if surface normal vectors are expected.

An efficient method to add a new point, based on an existing point, is to select the existing point then update the fields that are different and then press the add (+) button when done. The original point selection remains active and can be deleted by pressing the (-) button if the original point is no longer required.

Points with identical label names and values are allowed.

Ballbar Artifact

This option creates individual touches specific for the measurement of a ballbar. Only two points are expected for a typical ballbar inspection program. Illustration 13 shows an example of the ballbar artifact.

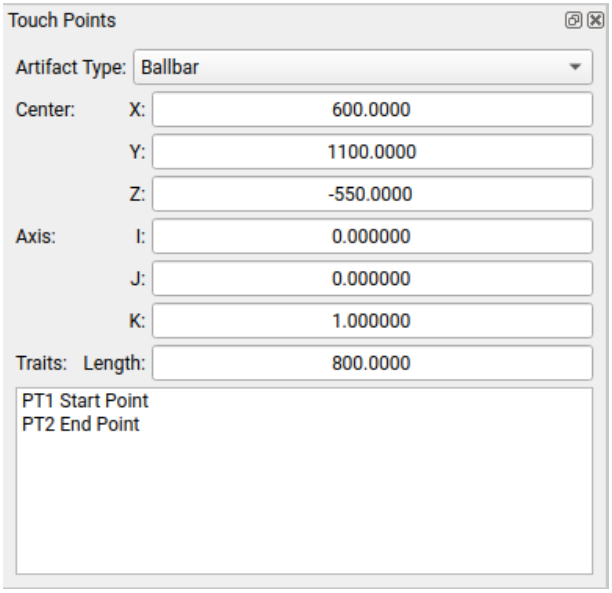


Illustration 13: Ballbar points.

Gaugeblock Artifact

This option creates individual touches specific for the measurement of a gaugeblock. Three points are expected for a typical gaugeblock measurement. Illustration 14 shows an example of the gaugeblock artifact window.

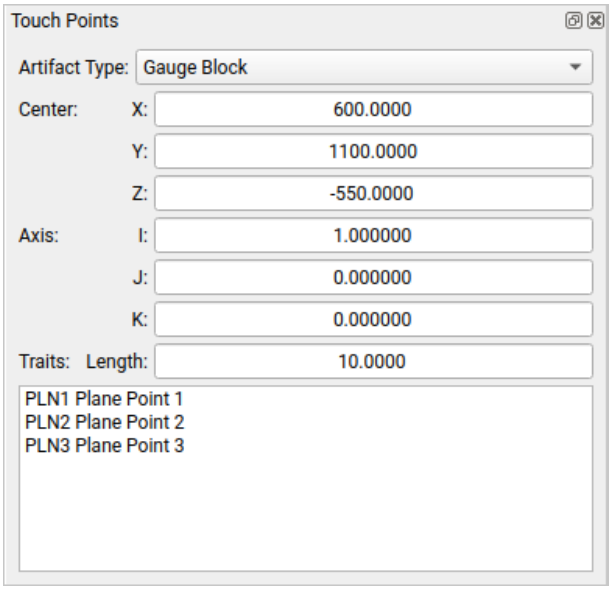


Illustration 14: Gaugeblock measurement.

Ringgauge Artifact

This option creates individual touches specific for the measurement of a ring or pin gauge. Illustration 15 shows an example of the ringgauge artifact window with options for ID or OD.

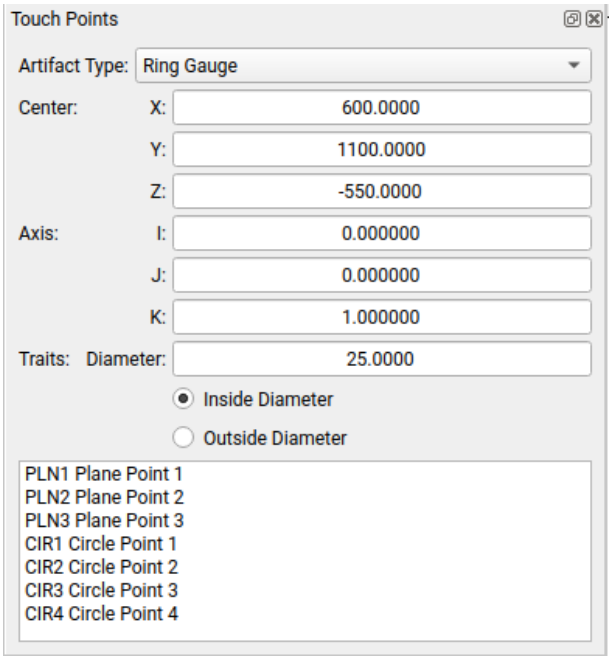


Illustration 15: Ring or Pin gauge measurement.

Sphere Artifact

This option creates individual touches specific for the measurement of a sphere. Illustration 16 shows an example of a sphere artifact window. For measurements that involve only a single touch of a sphere the last point, identified as *Probe Normal*, can be used.

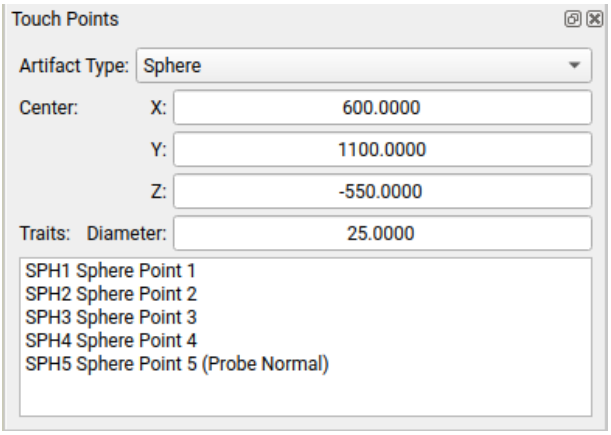


Illustration 16: Sphere measurement.

Stepgauge Artifact

This option creates individual touches specific for the measurement of a stepgauge. Illustration 17 shows an example of a stepgauge artifact window. Stepgauge measurements involve a datum plane, line, and separate start and end points.

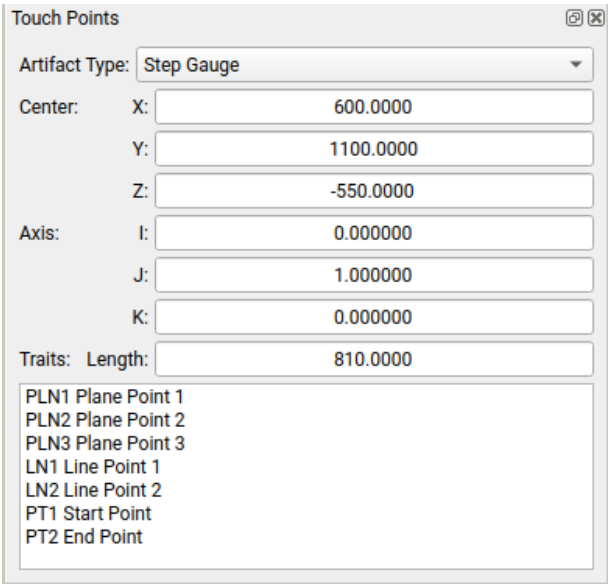


Illustration 17: Stepgauge measurement.

Information Display

The information section of the Virtual CMM utility consists of the *Message Log*, *Communication*, and position display (DRO). illustration 18 shows an example of the information displayed.

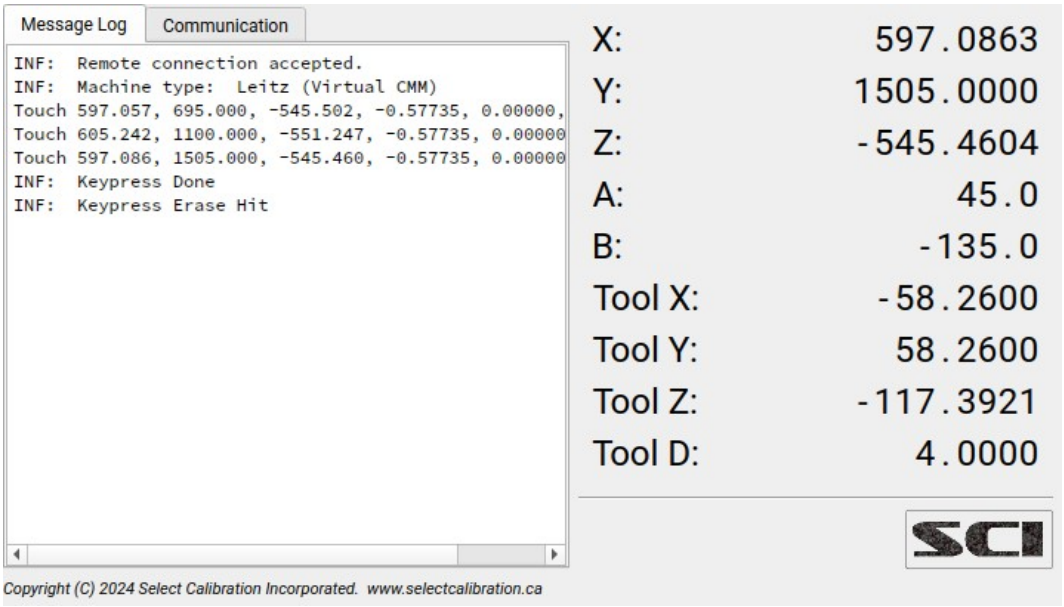


Illustration 18: Virtual CMM information display.

Message Log

The Message Log contains event information from the execution of the *VirtualCMM*. Illustration 19 shows examples of different events shown in the log such as touch points, jogbox key presses, errors, and other information.

```
INF:  Remote connection accepted.
ERR:  Machine in E-Stop
INF:  Machine Online
Touch 601.414, 1100.000, -948.586, -0.70711, 0.00000, -0.70711
Touch 151.000, 651.000, -312.190, -0.50000, -0.50000, -0.70711
INF:  Keypress Done
Touch 601.414, 1100.000, -948.586, -0.70711, 0.00000, -0.70711
INF:  Keypress Erase Hit
Touch 601.000, 1099.000, -948.586, -0.50000, 0.50000, -0.70711
Touch 358.071, 1356.071, -595.032, -0.50000, 0.50000, -0.70711
Touch 101.000, 1599.000, -241.479, -0.50000, 0.50000, -0.70711
Touch 546.586, 1150.586, -882.118, 0.70711, 0.70711, 0.00000
Touch 146.586, 1550.586, -316.432, 0.70711, 0.70711, 0.00000
Touch 601.828, 1103.828, -954.243, -0.50000, 0.50000, 0.70711
Touch 94.828, 1610.828, -237.236, 0.50000, -0.50000, -0.70711
```

Illustration 19: Message log window showing record of different events from the VirtualCMM.

Communication

The Communication section shows the raw data between the inspection software and the *VirtualCMM* utility. The number of displayed lines can be set by adjusting the *Buffer Size* option. Illustration 20 shows an example of the communication data displayed by the virtual CMM.

```
>> [......1676] POS 598.17157, 1096.17157, -957.07107, 0.00000, 0.00000, 0.00000, 0.00000
<< [......1677] MOVABS 607.67157, 1086.67157, -943.63604
>> [......1678] %
<< [......1679] MOVABS 607.67157, 1086.67157, -943.63604
>> [......1680] %
<< [......1681] GETPOS
>> [......1682] POS 607.67157, 1086.67157, -943.63604, 0.00000, 0.00000, 0.00000, 0.00000
<< [......1683] PROBE 595.17157, 1099.17157, -952.82843,, 0.500000,-0.500000,-0.707107
>> [......1684] POINT 596.17157, 1098.17157, -954.24264, 0.00000, 0.50000, -0.50000, -0.707
<< [......1685] GETPOS
>> [......1686] POS 598.17157, 1096.17157, -957.07107, 0.00000, 0.00000, 0.00000, 0.00000
```

Buffer Size: 16

Illustration 20: Communication log showing received and sent data from the inspection software.

The >> symbol is for data sent by VirtualCMM where << is for received data.

Display Readout

The display readout shows the position of the CMM, the AB head angles, the tool offsets, and the tool tip diameter. The CMM position is always the center of the stylus tip and is a combination of the machine position and tool offset.

X:	541.7400
Y:	1041.7400
Z:	-617.3921
A:	45.0
B:	-45.0
Tool X:	-58.2600
Tool Y:	-58.2600
Tool Z:	-117.3921
Tool D:	4.0000

Protocol Limitations

There are limitations when using the *VirtualCMM* as compared to a real CMM. The primary purpose of *VirtualCMM* is for the testing of *MeasureDirect* when a real CMM is not available so, ideally, the simulator is identical to that of a real CMM as much as possible but some variations are acceptable.

Supported Commands

The first limitation is the number of commands supported by *VirtualCMM* where only a subset of commands are available that is necessary for the measurement of the artifacts by *MeasureDirect* and other related functions. When running a version of *VirtualCMM* that has the debug option enabled unsupported commands will be reported in the debug window.

For the I++ DME Server there are published specifications so it was possible to implement most of the required functions. Not all commands are supported and not all state changes are strictly adhered to. The implementation should be enough to function as a practical CMM replacement regardless.

The Leitz and DC protocols are proprietary for Hexagon CMM's and only partially implemented in VirtualCMM. The I++ DME is not technically a standard but a tightly controlled specification by a consortium of manufacturers.

Synchronization of Commands

The second limitation is how commands are synchronized. Commands sent to a real controller are buffered and executed in the order sent where only some commands sent to the *VirtualCMM* are handled asynchronously.

The DC controllers have at least two execution queues allowing some commands to be executed in parallel. When using the secondary queue for commands such as GETPOS the behaviour is closer to that of VirtualCMM.

Starting with version 7.4 of VirtualCMM the GETPOS command for Leitz and DC controllers are buffered and returned in the order received but many other commands are not.

I++ DME Tools

Starting with version 6.0 *VirtualCMM* can be setup as an I++ server. For the most part this addition is not that different than other supported interface types with the exception of the tools where the I++ Server is required to handle all tool functions.

The method used by *VirtualCMM* is to generate a generic list of tools with angles suitable for *MeasureDirect* testing. When calling the function EnumTools() the returned tool list is the default list which is a list of probes at increments of 45 degrees for the A and B axis.

When requesting a tool position with head AB angles that are not part of the pre-defined list then the probe will be dynamically created provided the head AB angles can be extracted from the tool name. The name of the tool must include the AB angles as part of the name in a format that contains Ax.xBy.y somewhere in the name. The AB angle names can be separated by various characters. The following are examples of tool names that are acceptable:

Tool_Name.10.41.A90-B90
Tool_Name.A90B90
Tool_Name.TIPA90B0
T1A90.B90

Interpolated tool positions are not added to the list of tools returned by EnumTools(). In the event the name of the tool does not match existing tools, and the AB angles cannot be extracted from the tool name, a default tool is created at angles A0B0.

Stepgauge 10360-2 Measurement Example

This example shows the measurement of a step gauge using *MeasureDirect* connected to the *VirtualCMM*.

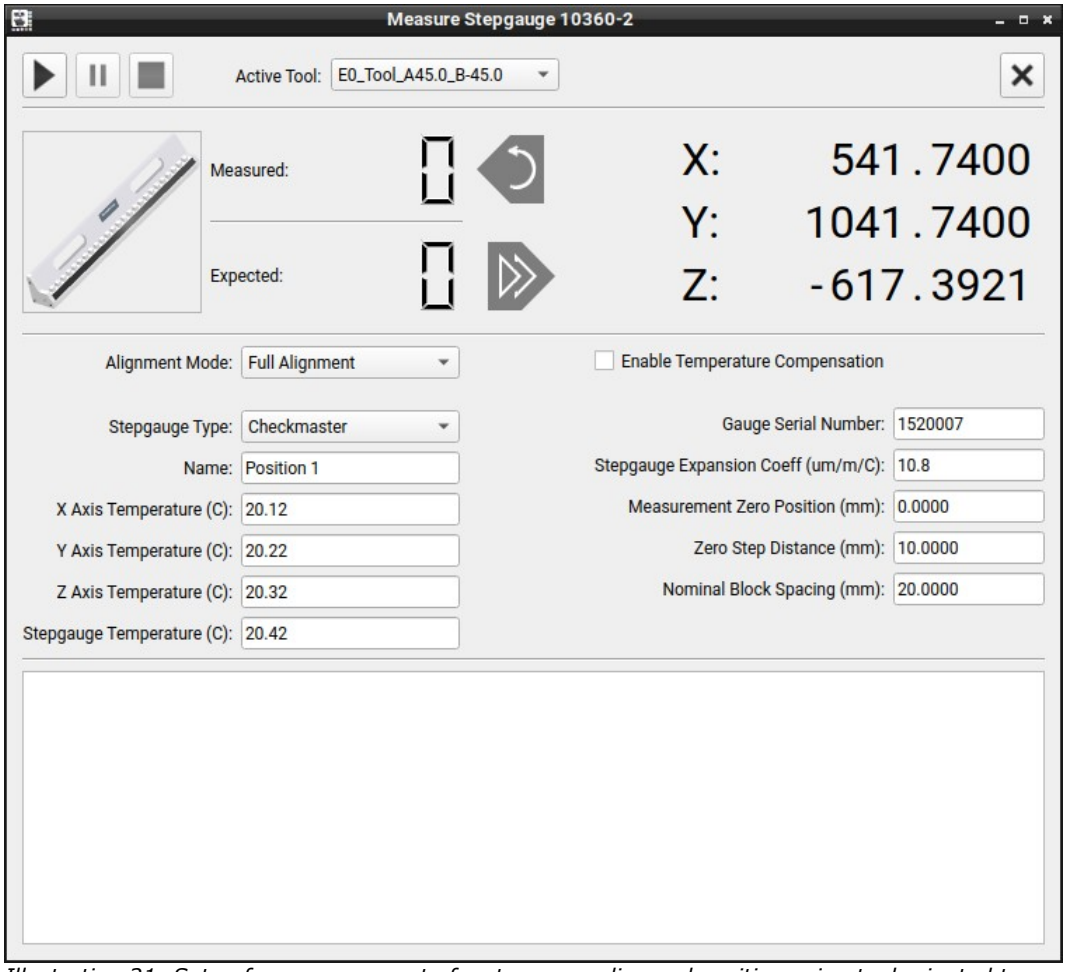


Illustration 21: Setup for measurement of a stepgauge diagonal position using tool oriented to A45, B-45.

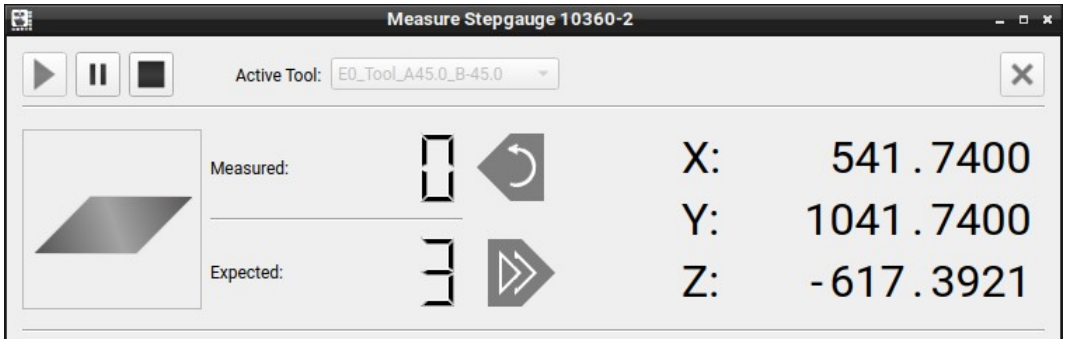


Illustration 22: Waiting for the three plane points on the stepgauge.

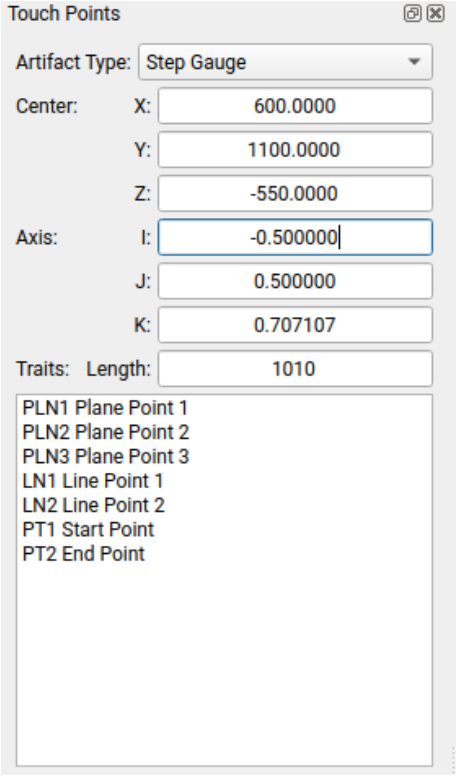


Illustration 23: Selection of points used for the step gauge measurement starting with the plane PLN1, PLN2, and PLN3.

```
Touch 854.222, 843.991, -903.391, -0.86603, -0.28868, -0.40825
Touch 601.774, 1104.674, -552.050, -0.86603, -0.28868, -0.40825
Touch 349.267, 1349.006, -189.192, -0.86603, -0.28868, -0.40825
Touch 852.500, 853.194, -911.115, 0.00000, -0.81650, 0.57735
Touch 347.500, 1358.252, -196.978, 0.00000, -0.81650, 0.57735
Touch 346.512, 1353.488, -191.513, 0.50000, -0.50000, -0.70711
Touch 853.494, 846.506, -908.495, -0.50000, 0.50000, 0.70711
Touch 346.523, 1353.477, -191.529, 0.50000, -0.50000, -0.70711
```

Illustration 24: Record of touch points shown in the message log.

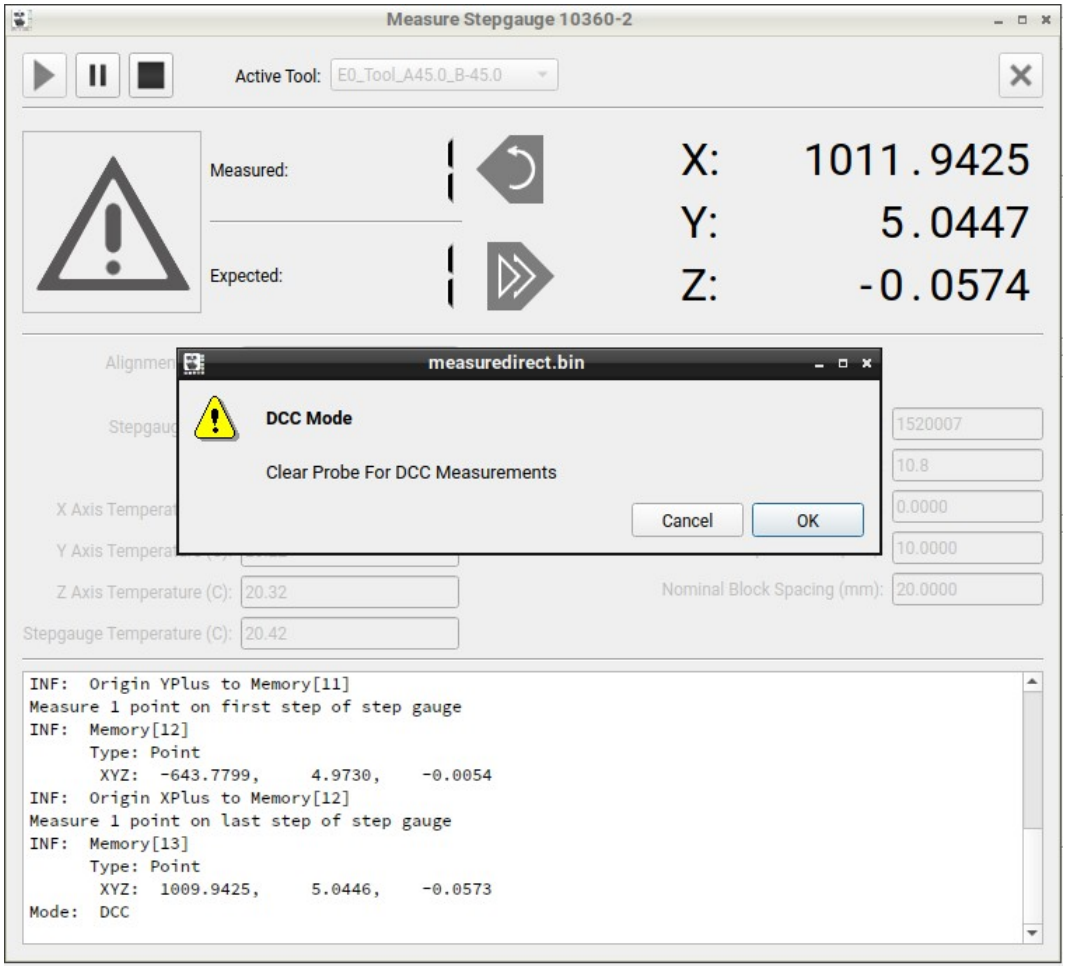


Illustration 25: Following the last measurement point MeasureDirect will begin the automatic sequence.

Illustration 26: Jogbox function buttons.

The jogbox DONE button can be used to acknowledge features and prompts in MeasureDirect. DEL PNT can erase accidental touch points.

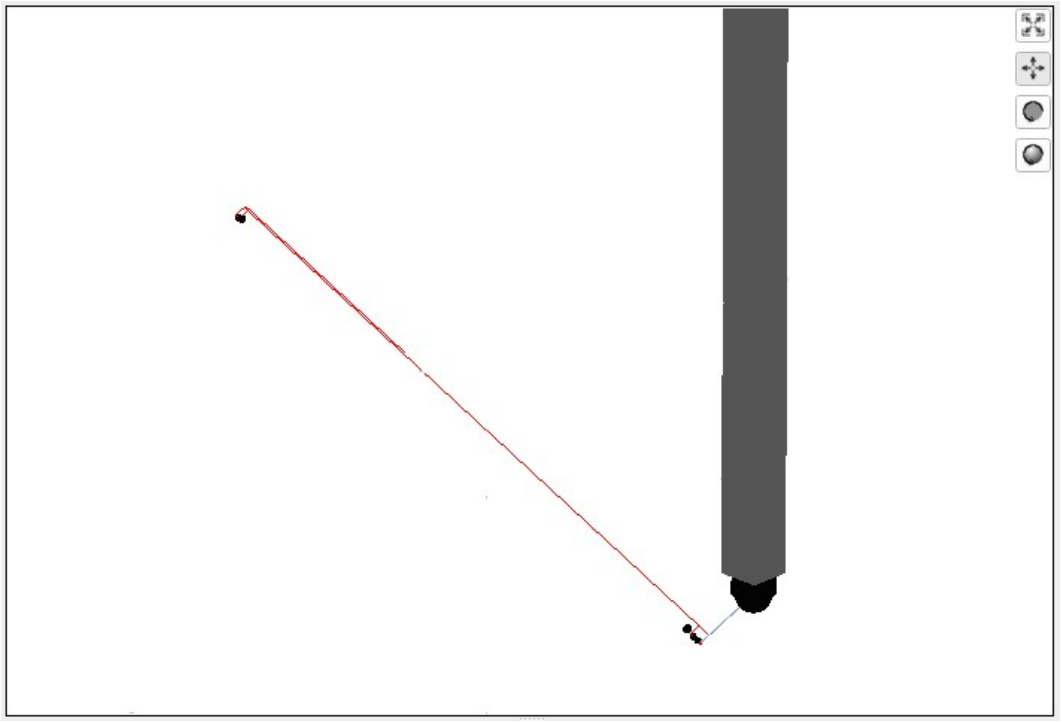


Illustration 27: View of the step gauge measurement sequence.

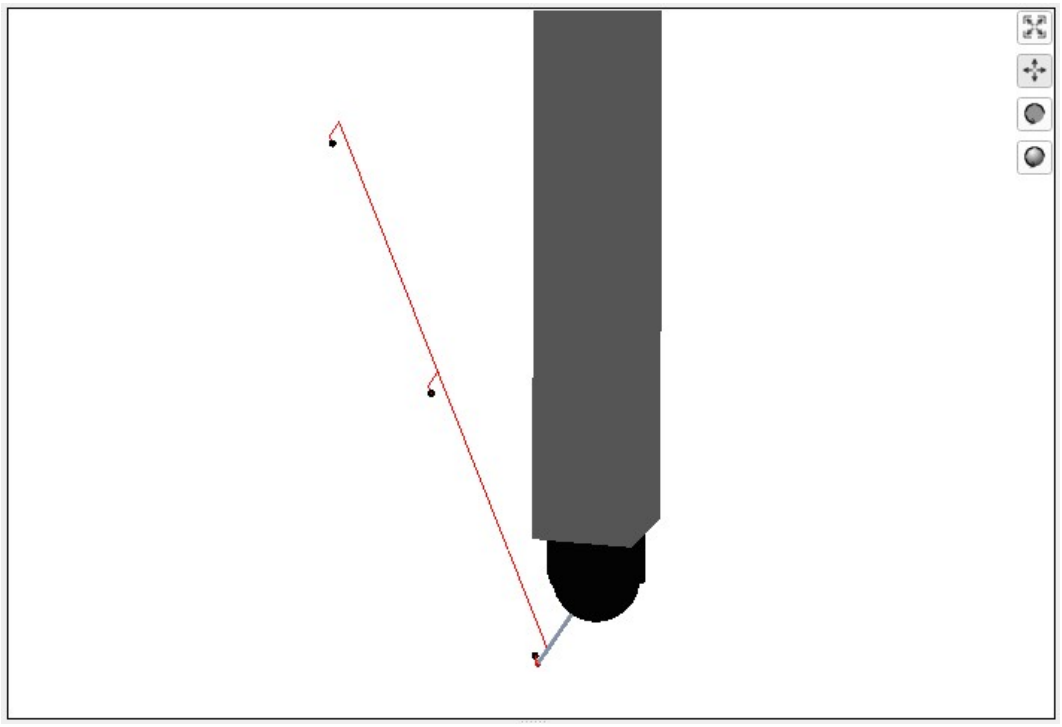


Illustration 28: Another view of the step gauge measurement sequence.

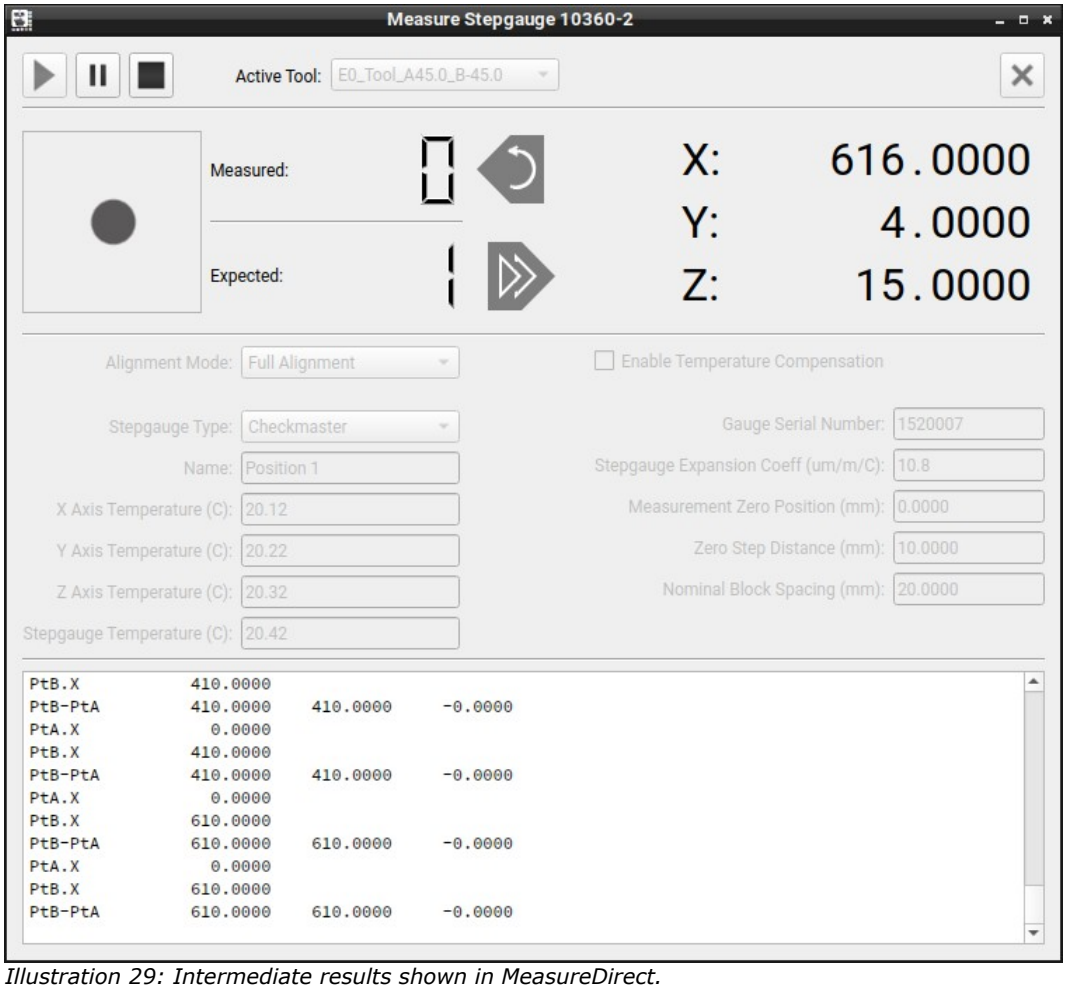


Illustration 29: Intermediate results shown in MeasureDirect.

Revision History		
Date	*Version*	*Changes*
Dec 4, 2021	1.0	New program.
Dec 18, 2021	1.1	[bug fix] DRO not updated when probe offsets changed. [bug fix] Path lines not displayed on program startup. Added direct XYZ move option Update of OpenGL drawing routines. Changed sensitivity of wheel mouse zoom.
Jan 16, 2022	2.0	[bug fix] Make path line buffer end at last shown touch. [bug fix] Min and Max axis limits not validated. [bug fix] Home position not always at front / left / top. [bug fix] SG B+/-90 points does not fit inside the default volume. [bug fix] Not checking E-Stop at start of moves. Increase maximum size of path buffer. Added debug option to turn off logging of position requests. Added option for serial communication. Added option to configure serial or Ethernet IP and port connection. Added option for circular moves. Drive machine to manual touch point locations. Draw entire machine and not just the Z axis. Remove bounding cube.
Apr 3, 2022	2.1	Added debug option for testing DC controllers error data. Added handler for READTP command. Added handler for WKPTCSY command. Accept 3 parameter PRBPIN commands. Improved command matching routine.
July 14, 2022	3.0	Added option to view raw communication data. Changed DRO so that digits are column aligned.
Oct 6, 2022	3.1	Added status indicator for selected protocol. Added commands related to temperature compensation.
Nov 16, 2022	4.0	Added type of controller to communication options.
Nov 20, 2022	4.1	[bug fix] DC controller jogbox Done and Del-Pnt not working.
Jan 2, 2023	5.0	Added probe offset display in coordinate window. Added additional step gauge measurement patterns. Added additional gauge block measurement patterns. Added option to override the tip diameter size. Updated graphics of probe head and machine. Changed test data names to be uniform across all measurements. Added warning if closed while still connected.
Jun 27, 2023	5.1	[bug fix] DC controller show stroke commands missing fields.
July 14, 2023	6.0	[bug fix] Leitz jogbox DelPnt not sending proper sequence. [bug fix] Project file Error_Level 0 not working as intended. Separated interface types into separate classes. Major rewrite.

Revision History		
		Added I++ DME Server as a protocol option. Added option to generate random touch point errors.
July 16, 2023	6.1	[bug fix] Shift in machine position following a probe change. [bug fix] Volume error messages missing quotation marks. [bug fix] Error messages not sending termination sequence.
July 21, 2023	6.2	[bug fix] Noise value written to tip diameter of active machine. [bug fix] I++ GetProp returns a default value instead of error. Added support for controller alignment systems for I++ Server. Added additional minor I++ commands.
July 28, 2023	6.3	[bug fix] I++ Manual touch points not converted to active coordinate system.
Jan 3, 2024	7.0	Addition of artifact measurement types Machine is no longer in E-Stop on startup. Change of terminology. Using Tool instead of Probe for example. Removed option to automatically generate test points. Position of tip after manual touch drawn at touched surface.
May 14, 2024	7.1	Added option to set I++ tool offsets
Aug 22, 2024	7.2	Implemented FindTool / FoundTool I++ functions. Changed I++ to report tool offsets from center of head sphere.
Sep 18, 2024	7.3	[bug fix] Return effective tool offsets and not nominal offsets. Added options for Tool.EOffset and FoundTool.EOffset Added option for tool.ER()
Jan 29, 2025	7.4	Queue of GETPOS command for Leitz and DC controllers. Disable inactive communication options. Added optional MOVABS syntax for Leitz and DC controllers. Added temperature attribute for machine axis and part. Added temperature related functions for all controllers.
Feb 25, 2025	7.5	[bug fix] Ring and Pin gauge axis may not match input axis. [bug fix] Ring and Pin gauge plane point samples based on diameter. [bug fix] Enforce a minimum diameter for pin, ring, and sphere.
Jan 1, 2026	8.0	[bug fix] Depth scaling causes artifacts to be drawn incorrectly. Changed random number generator to return a logarithmic result. Added machine scale and squareness error option. Added internal option to have machine startup un-homed on macos and GNU/Linux.