UM1025
User manual

Getting started with STM-STUDIO

Introduction

STM-STUDIO is a graphical user interface that allows sampling and visualizing in real time of user's variables while the application is running. It is designed to run on PCs with Microsoft® Windows operating systems.

This tool works with STM8 microcontrollers through SWIM (single wire interface module) and with STM32 microcontrollers through JTAG or SWD (serial wire debug) interface.

Contents

List of figures

1 Installing STM-STUDIO

1.1 JRE installation

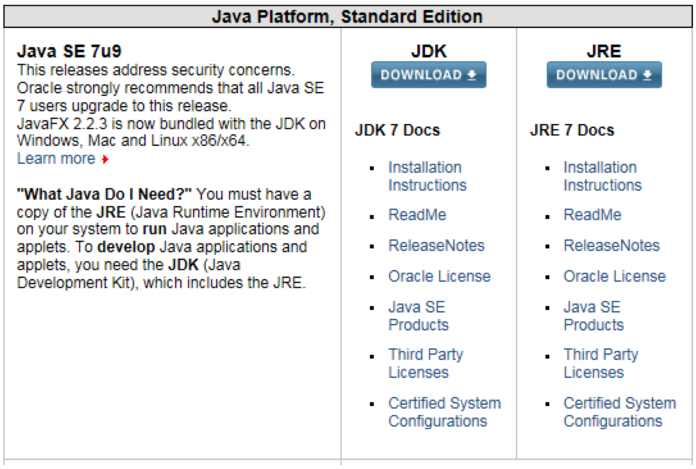
Ensure that the Java Run Time Environment (JRE) is installed on your machine. JRE version 1.7 or more recent is recommended (JRE 7).

To check the installed JRE version, check that Java bin path is already added to PATH environment variable, then open a Windows command dialog and enter java -version.

If you have a JRE version older than 1.7, please download the latest JRE version from http://www.oracle.com/technetwork/java/javase/downloads/index.html.

STM-STUDIO requires the 32-bit version of the JRE to be installed, even on 64-bit versions of Windows.

Figure 1. Java Platform



Java Platform, Standard Edition

Java SE 7u9
This releases address security concerns. Oracle strongly recommends that all Java SE 7 users upgrade to this release.
JavaFX 2.2.3 is now bundled with the JDK on Windows, Mac and Linux x86/x64.
Learn more

"What Java Do I Need?" You must have a copy of the **JRE** (Java Runtime Environment) on your system to **run** Java applications and applets. To **develop** Java applications and applets, you need the **JDK** (Java Development Kit), which includes the JRE.

JDK

JDK 7 Docs

- Installation Instructions
- ReadMe
- ReleaseNotes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations

JRE

JRE 7 Docs

- Installation Instructions
- ReadMe
- ReleaseNotes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations

1.2 STM-STUDIO installation

Once the JRE is installed or updated, run STMStudio_setup.exe.

1.3 Hardware support

STM-STUDIO supports ST-LINK hardware with JTAG, SWD and SWIM protocols. STM-STUDIO also supports RLink (in-circuit debugger and programmer from Raisonance which supports JTAG, ICC and SWIM interfaces), STice advanced emulation system, and ST-TSLink as additional hardware targets.

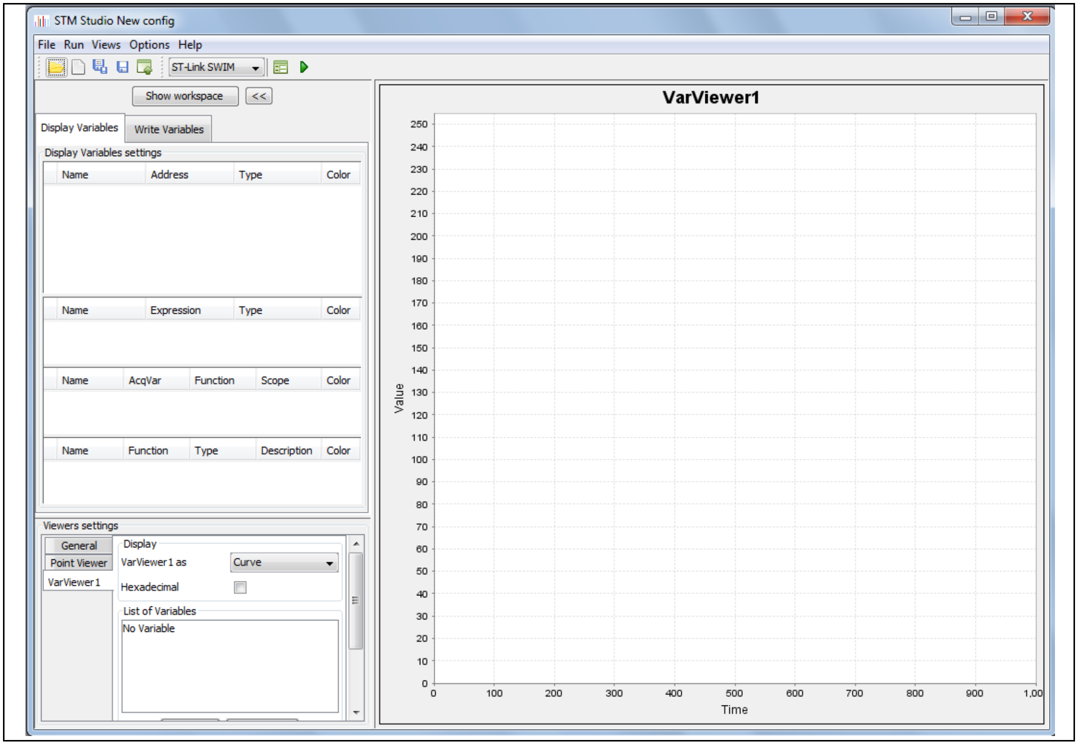
2 Running STM-STUDIO

2.1 Overview

Run STMStudio.exe through the Desktop icon or the Program Folder shortcut that are created during the installation setup. The screen shown in *Figure 2* appears.

- The configuration panes display current settings which change according to the type of display selected.
- The display area provides a visualization of the various settings currently selected.

Figure 2. STM-STUDIO initial screen



2.2 Creating variables

It is necessary to create a list of variables to be inspected during the recording session.

STM-STUDIO manages four kinds of variables:

- Absolute variables that are identified by their physical storage address.
- Statistical variables that can compute values like min, max, average and standard deviation of absolute variables.
- Expression variables that are the result of a mathematical expression evaluation. An expression is the combination of absolute or statistical variables and mathematical operators (+,-,*, /…), for example: (Variable1+Variable2)*Variable3. Note that expression variables are evaluated after statistical variables, thus it is not possible to compute statistics on expressions.
- Plugin variables that contain user-configurable information.

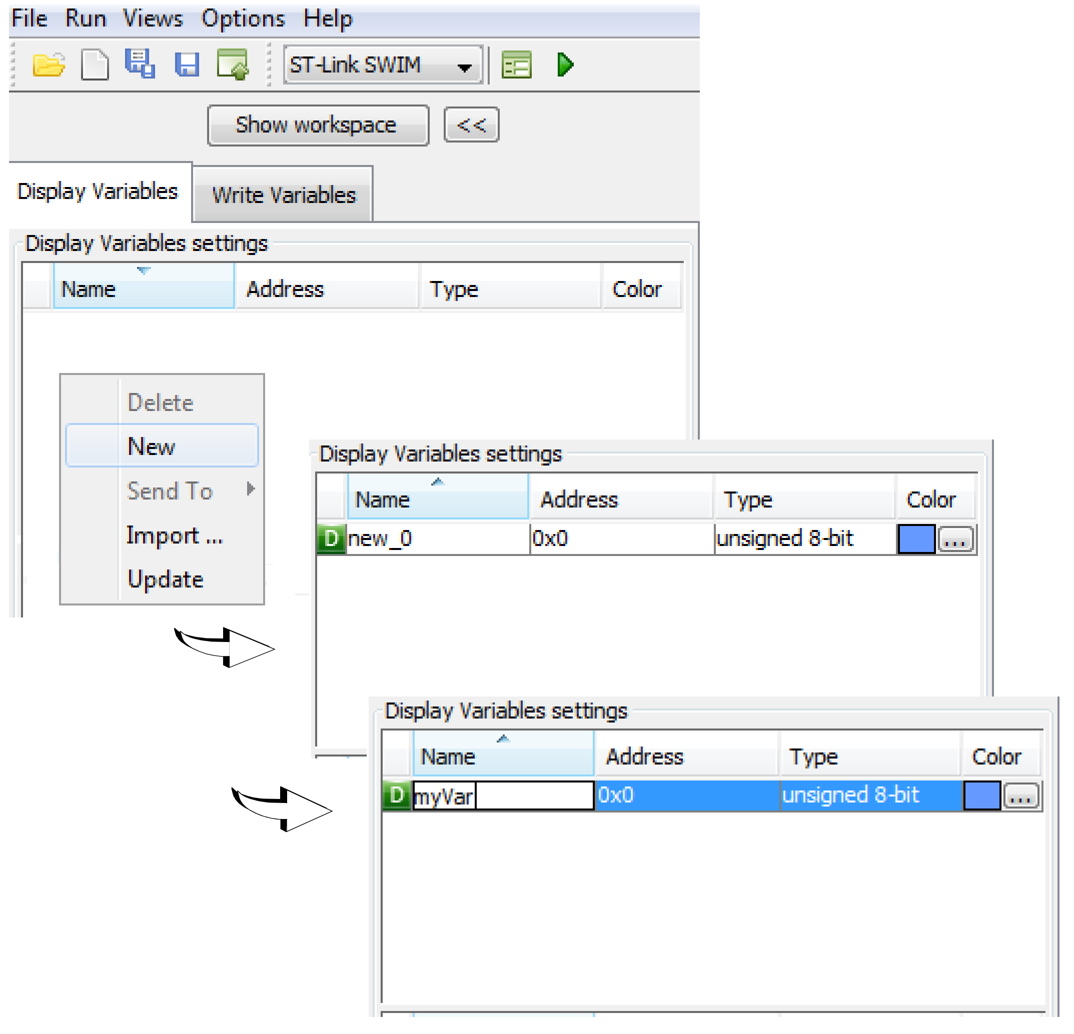
2.2.1 Adding absolute variables

Absolute variables are displayed in the upper pane of the **Variables Settings** pane. They can either be added directly, or be imported from an executable (.elf file).

Directly

Variables can be added directly from the table using the **New** contextual menu (right click on the mouse). The new variable is added with default parameters. It is the user’s responsibility to define the name, the address, the size or optionally the color of the variable by editing each of the table fields.

Figure 3. Adding absolute variable directly from table



Name	Address	Type	Color

	Name	Address	Type	Color
D	new_0	0x0	unsigned 8-bit	...

	Name	Address	Type	Color
D	myVar	0x0	unsigned 8-bit	...

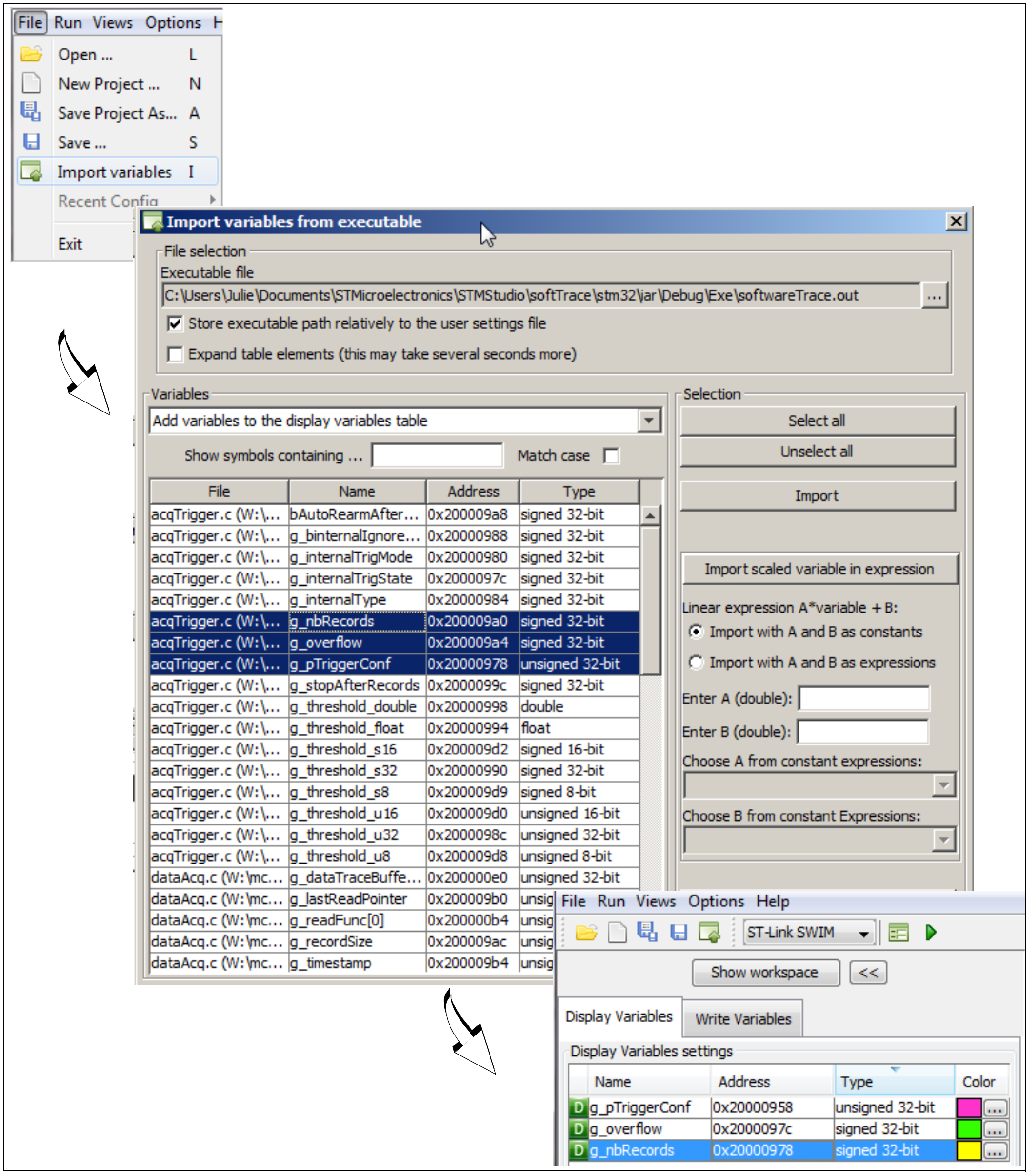
Imported

Variables can be imported from an application executable using **File>Import variables**:

1. Select the file that contains the variables.
   a) Select whether to **Store executable path relatively to the user settings file**.
   Choose this to adapt the STMStudio user settings file to the directory tree policy, which is preferable when copying the project from one PC to another one, storing the path relatively allows the copied project to open without having to browse for the new application location.
   If the application remains at a fixed location, an absolute path is preferable. Note that if the STMStudio settings file and application file are not stored on the same disk, the path is necessarily stored in absolute.
   b) Select whether to **Import scaled variables in expression**:
   It is possible to import variables directly as scaled expressions, instead of importing variables then creating expressions. In this case, the 'A' and 'B' coefficients of the linear expression must be specified before importing. Several variables can be imported with the same coefficients.
   If one or both coefficients may change during the project lifetime, it is recommended to first create constant expressions, before importing scaled variables, and use these constant expressions as 'A' and 'B' coefficients. This allows future changes to the 'A' and 'B' values without modifying the scaled expressions.
   c) Select the **Show symbols containing ...** and **Match case** to filter the symbols displayed in the edit box.
   d) Click on the **column header** to sort the list by column if desired.
   e) Use multiple selection mode if desired:
   – **shift+click** for range selection,
   – **ctrl+click** for single item addition,
   – **shift+ctrl+click** for range addition.
2. Select the variables.
3. Click the **Import** button.

The address, size and name parameters are provided by the executable debug information.

Figure 4. Importing absolute variables



File selection

Executable file

C:\Users\Julie\Documents\STMicroelectronics\STMStudio\softTrace\stm32\iar\Debug\Exe\softwareTrace.out

☑ Store executable path relatively to the user settings file

□ Expand table elements (this may take several seconds more)

Variables

Add variables to the display variables table

Show symbols containing ... Match case □

File	Name	Address	Type
acqTrigger.c (W:\...	bAutoRearmAfter...	0x200009a8	signed 32-bit
acqTrigger.c (W:\...	g_binternalIgnore...	0x20000988	signed 32-bit
acqTrigger.c (W:\...	g_internalTrigMode	0x20000980	signed 32-bit
acqTrigger.c (W:\...	g_internalTrigState	0x2000097c	signed 32-bit
acqTrigger.c (W:\...	g_internalType	0x20000984	signed 32-bit
acqTrigger.c (W:\...	g_nbRecords	0x200009a0	signed 32-bit
acqTrigger.c (W:\...	g_overflow	0x200009a4	signed 32-bit
acqTrigger.c (W:\...	g_pTriggerConf	0x20000978	unsigned 32-bit
acqTrigger.c (W:\...	g_stopAfterRecords	0x2000099c	signed 32-bit
acqTrigger.c (W:\...	g_threshold_double	0x20000998	double
acqTrigger.c (W:\...	g_threshold_float	0x20000994	float
acqTrigger.c (W:\...	g_threshold_s16	0x200009d2	signed 16-bit
acqTrigger.c (W:\...	g_threshold_s32	0x20000990	signed 32-bit
acqTrigger.c (W:\...	g_threshold_s8	0x200009d9	signed 8-bit
acqTrigger.c (W:\...	g_threshold_u16	0x200009d0	unsigned 16-bit
acqTrigger.c (W:\...	g_threshold_u32	0x2000098c	unsigned 32-bit
acqTrigger.c (W:\...	g_threshold_u8	0x200009d8	unsigned 8-bit
dataAcq.c (W:\mc...	g_dataTraceBuffe...	0x200000e0	unsigned 32-bit
dataAcq.c (W:\mc...	g_lastReadPointer	0x200009b0	unsig
dataAcq.c (W:\mc...	g_readFunc[0]	0x200000b4	unsig
dataAcq.c (W:\mc...	g_recordSize	0x200009ac	unsig
dataAcq.c (W:\mc...	g_timestamp	0x200009b4	unsig

Selection

Linear expression A*variable + B:

● Import with A and B as constants

○ Import with A and B as expressions

Enter A (double):

Enter B (double):

Choose A from constant expressions:

Choose B from constant Expressions:

Display Variables settings

	Name	Address	Type	Color
D	g_pTriggerConf	0x20000958	unsigned 32-bit	🟪
D	g_overflow	0x2000097c	signed 32-bit	🟩
D	g_nbRecords	0x20000978	signed 32-bit	🟨

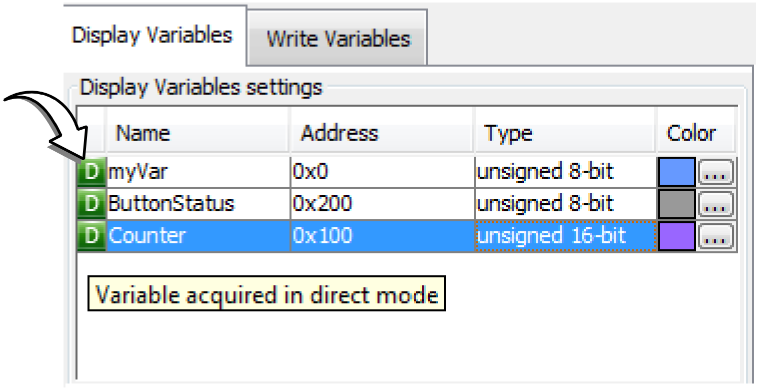
Acquisition mode

Absolute variables may be acquired in 2 modes:

- Direct acquisition mode, which is not application intrusive, but does not give an instantaneous view of the application state.
- Snapshot acquisition mode, which gives an accurate view of the application state, but which requires application instrumentation. Refer to the *Section 2.5: Configuring the acquisition settings* for configuring this mode.

Click on the icon on the left column in order to toggle the acquisition mode of a variable.

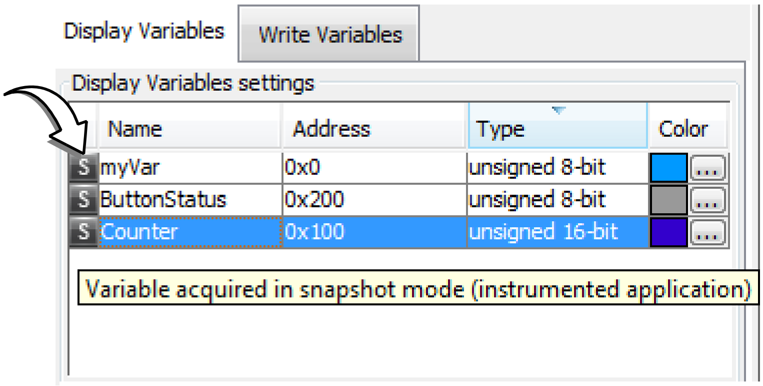
Figure 5. Direct acquisition mode



	Name	Address	Type	Color
D	myVar	0x0	unsigned 8-bit	■ ...
D	ButtonStatus	0x200	unsigned 8-bit	■ ...
D	Counter	0x100	unsigned 16-bit	■ ...

Variable acquired in direct mode

Figure 6. Snapshot acquisition mode



	Name	Address	Type	Color
S	myVar	0x0	unsigned 8-bit	■ ...
S	ButtonStatus	0x200	unsigned 8-bit	■ ...
S	Counter	0x100	unsigned 16-bit	■ ...

Variable acquired in snapshot mode (instrumented application)

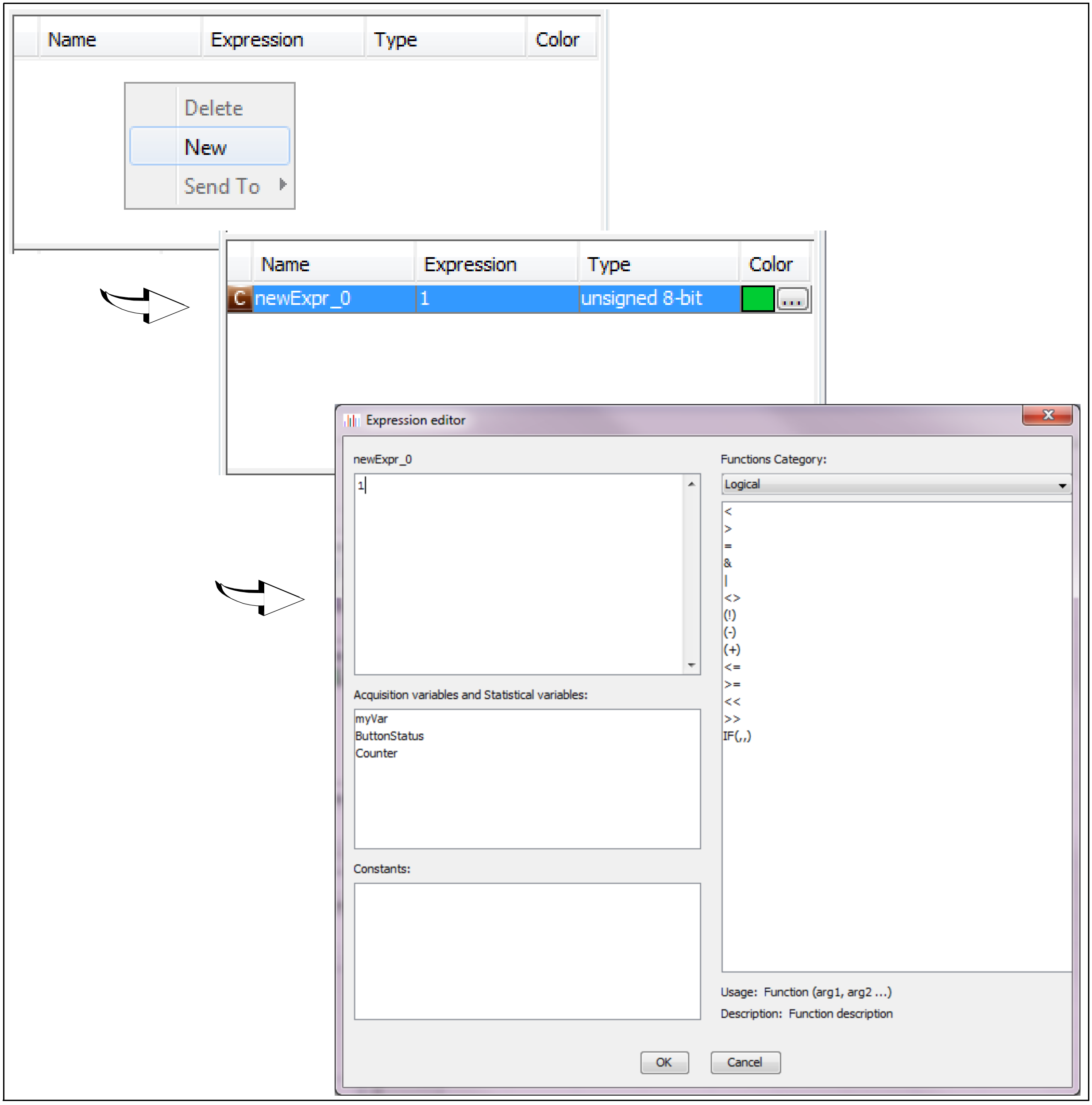
2.2.2 Adding expression variables

Expression variables are displayed in the second leftmost pane in the **Variables settings** pane. Use the **New** contextual menu to add a new variable. For details on the allowed expression syntax, please refer to *Appendix A: Syntax for expression variables on page 38*.

Clicking on the **Expression** column will open the expression editor window, which will assist you writing the expression (providing the list of known symbols and functions).

Expressions can use application variables or statistic variables. They can also be constant (depending on no application or statistic variable). In the general case the result of an expression cannot be re-used in another expression; only constant expressions can be reused in other expressions.

Figure 7. Contextual menu to add expression variable



	Name	Expression	Type	Color
C	newExpr_0	1	unsigned 8-bit	🟩 ...

newExpr_0

1

Functions Category: Logical

<
>
=
&
|
<>
(!)
(-)
(+)
<=
>=
<<
>>
IF(,,)

Acquisition variables and Statistical variables:

myVar
ButtonStatus
Counter

Constants:

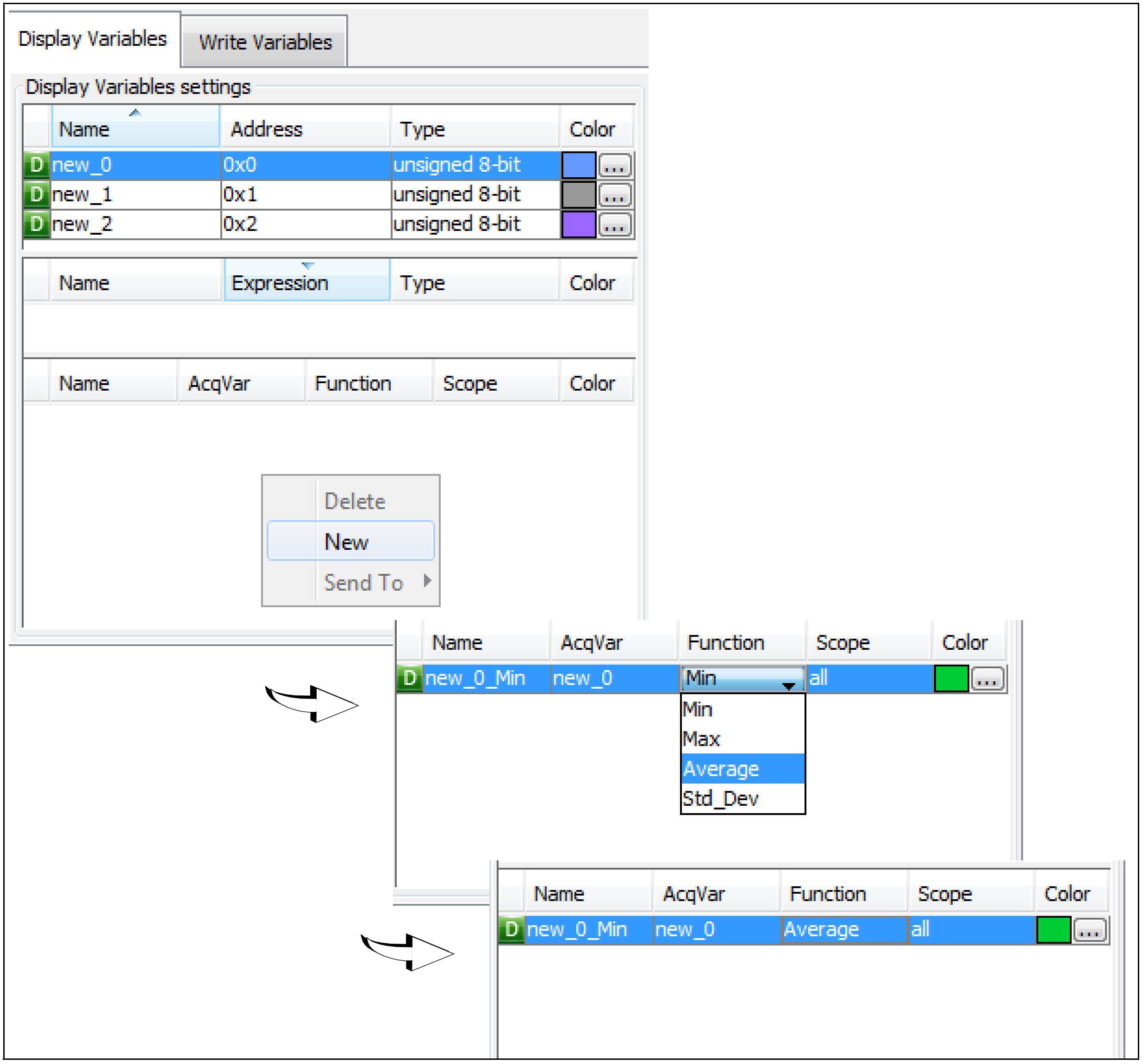
Usage: Function (arg1, arg2 ...)

Description: Function description

2.2.3 Adding statistical variables

Statistical variables are displayed in the second lowest pane of the **Variables settings** pane. Use the **New** contextual menu to add a new variable. Then select the absolute variable and the function to be computed. The scope defines the temporal window of the computing: on all records acquired since the acquisition start, or only on the last ones (number to be specified).

Figure 8. Adding statistical variables



	Name	Address	Type	Color	
D	new_0	0x0	unsigned 8-bit	🟦	...
D	new_1	0x1	unsigned 8-bit	▨	...
D	new_2	0x2	unsigned 8-bit	🟪	...

Name	Expression	Type	Color

Name	AcqVar	Function	Scope	Color

	Name	AcqVar	Function	Scope	Color	
D	new_0_Min	new_0	Min	all	🟩	...

	Name	AcqVar	Function	Scope	Color	
D	new_0_Min	new_0	Average	all	🟩	...

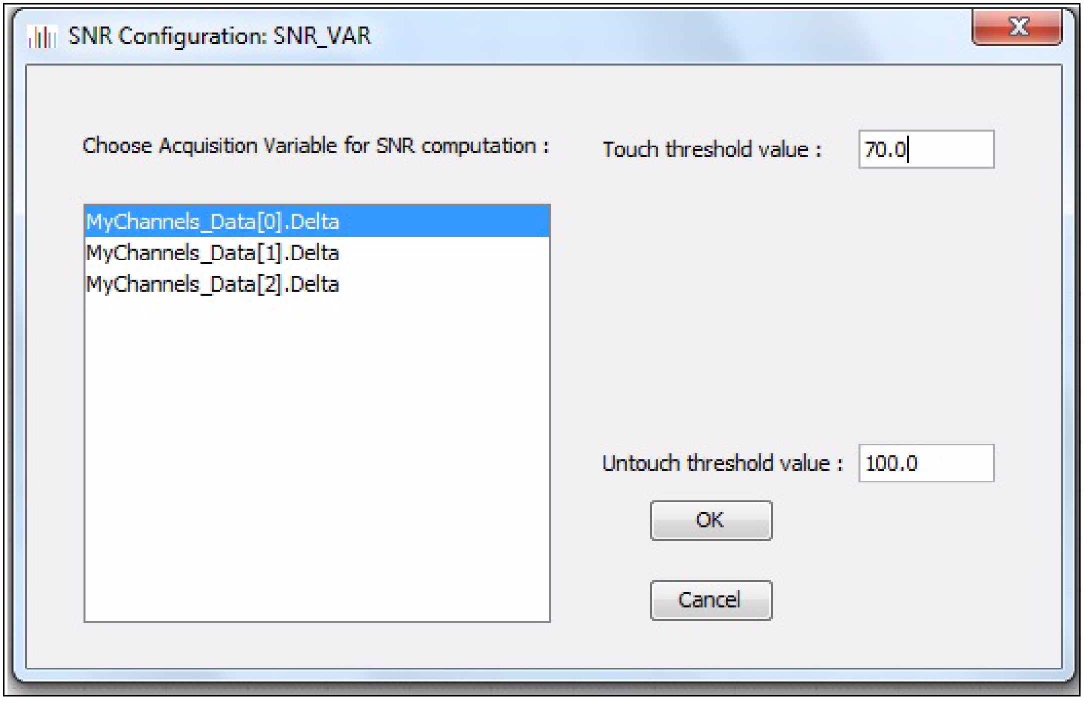
2.2.4 Adding plugin variables

Plugin variables are defined in an external java plugin. An example is provided with source and java documentation for Signal/Noise Ratio computation. Refer to the "plugins/SnrPlugin/javadoc" subdirectory of STMStudio application for details of how to implement your own plugin.

Use the New contextual menu to add a new variable. Then select in the list of available plugins ("Snr" is the first one available). A configuration window for the corresponding variable will open (see example in *Figure 9* for the SNR plugin).

In the case of the SNR plugin, select the variable on which the computation should be made, and select the "touch" and "untouch" thresholds. Please refer to the plugin javadoc or java source files for details on the SNR computation.

Figure 9. SNR plugin configuration



SNR Configuration: SNR_VAR

Choose Acquisition Variable for SNR computation :

MyChannels_Data[0].Delta
MyChannels_Data[1].Delta
MyChannels_Data[2].Delta

Touch threshold value : 70.0

Untouch threshold value : 100.0

2.3 Inspecting variables

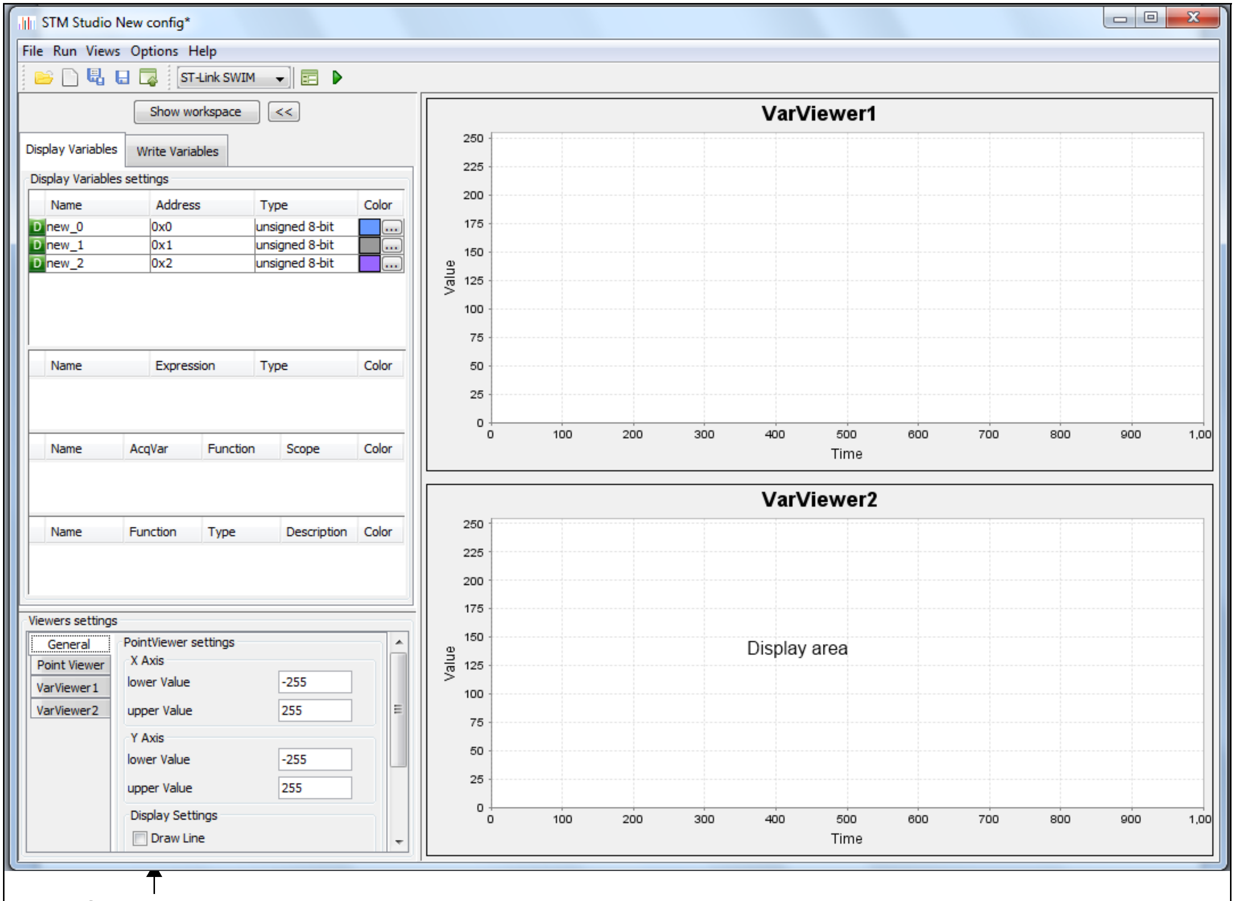
Once variables are created, they can be inserted in appropriate windows called “variable viewers” so that they can be visualized during a session. When creating a new configuration, one default variable viewer is created but the user can add as many variable viewers as necessary.

2.3.1 Adding a variable viewer

In the **Viewers settings** pane, select the **General** tab and click on the **Add Viewer** button; or use the contextual menu within the **Viewers settings** pane and select the **New Var Viewer** item.

A new tab (VarViewer2) is added to the **Viewers settings** pane and a **VarViewer2** window is displayed in the display area.

Figure 10. Creating a variable viewer



Name	Address	Type	Color
D new_0	0x0	unsigned 8-bit	🟦 ...
D new_1	0x1	unsigned 8-bit	▨ ...
D new_2	0x2	unsigned 8-bit	🟪 ...

PointViewer settings

X Axis

lower Value -255

upper Value 255

Y Axis

lower Value -255

upper Value 255

Display Settings

□ Draw Line

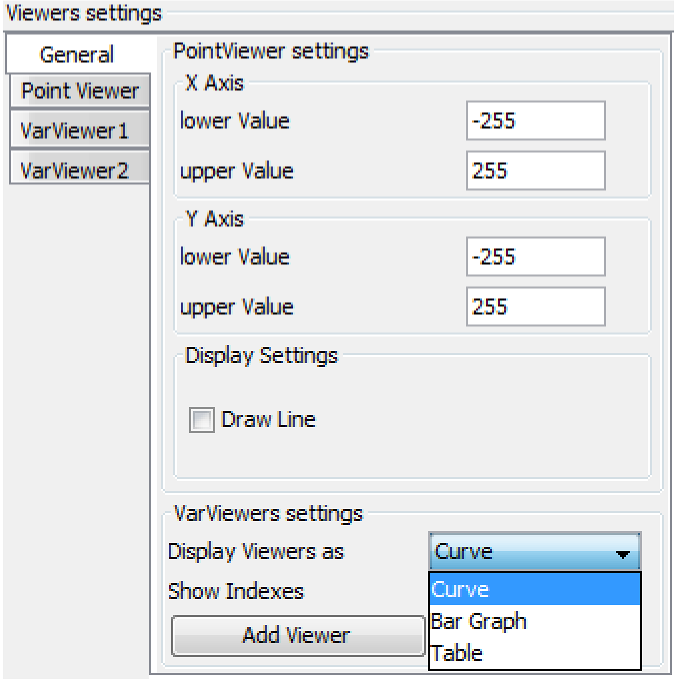
Settings pane

Each time a new viewer is created, a new tab is added to the **Viewers settings** control pane.

2.3.2 Customizing variable viewers

The **General** tab in the **Viewers settings** pane contains settings that apply to all variable viewers.

Figure 11. Customizing variable viewers



Viewers settings

General | Point Viewer | VarViewer1 | VarViewer2

PointViewer settings

X Axis

lower Value: -255

upper Value: 255

Y Axis

lower Value: -255

upper Value: 255

Display Settings

□ Draw Line

VarViewers settings

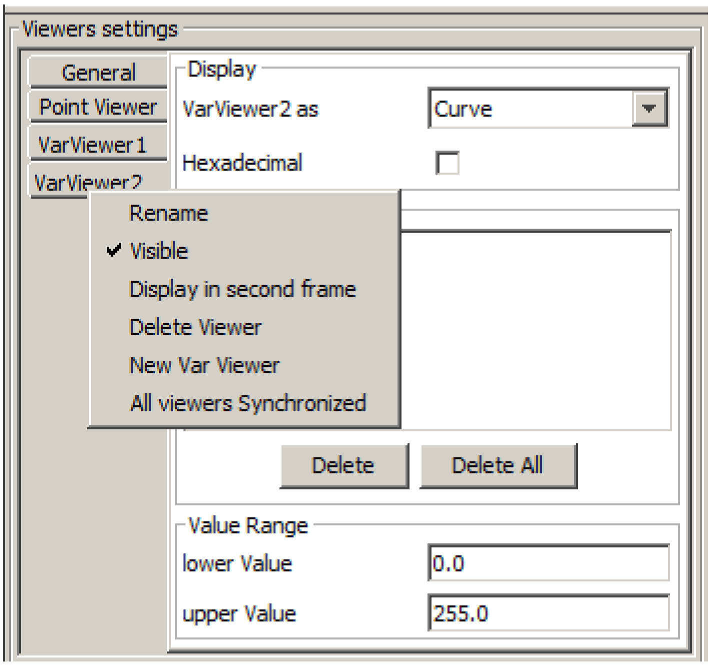
Display Viewers as: Curve

Show Indexes

- **The format of all viewers:** (Curve, Bar Graph or Table) can be changed here in a single action. This will override the local setting of each viewer, that may be changed afterwards.
- **Show Indexes:** show a slider for all variable viewers to display the time in ms.
- **Add Viewer:** create a new variable viewer.

Each viewer supports custom settings that can be modified: select the corresponding tab and change the settings.

Figure 12. Viewers settings



VarViewer2 as: Curve

Hexadecimal □

lower Value: 0.0

upper Value: 255.0

From the **Viewers settings** pane, you can:

- add a **New Var Viewer**, **Rename** the viewer, make the viewer **Visible** or **Delete** the selected viewer: use the contextual menu;
- select the viewer display format: Curve, Bar Graph or Table;
- select the variable display format: decimal or hexadecimal;
- remove a variable from the viewer: select the variable in the list and click **Delete**;
- clear the list of variables: press the **Delete All** button;
- change the vertical axis range of values: set **lower Value** and **upper Value**. Each viewer is associated to a specific range of values. If a variable is outside that range of values, the variable is not displayed. In this situation, it is useful to modify the vertical range to make the variable appear.

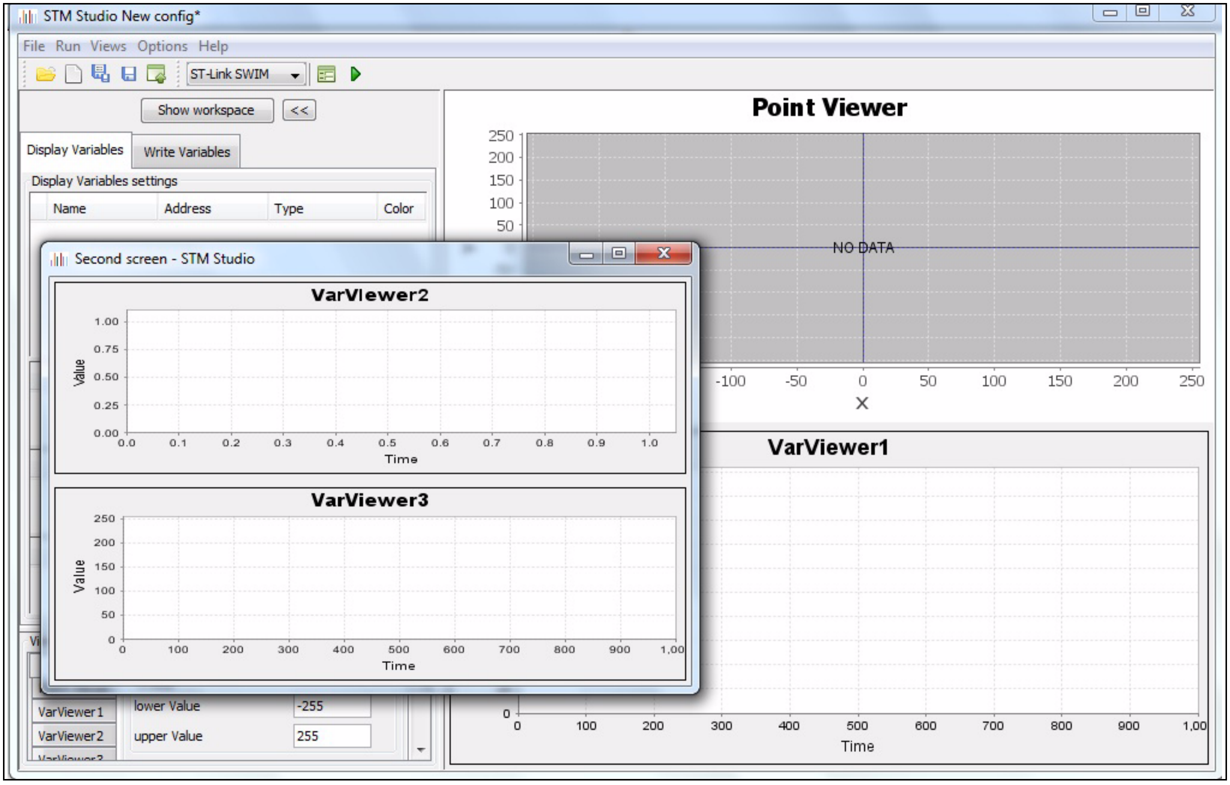
Note: *At least one variable viewer must be defined. Therefore, the **Delete Viewer** menu is displayed only if there are at least two variable viewers.*

The viewers can placed either on the right side of the STMStudio main window, or in a second display. If several viewers are displayed in one area:

- their size can be adjusted by a drag-and-drop action on the horizontal boundary,
- their position can be changed by a drag-and-drop action, initiated out of the display area of the viewer (where such action is associated with the zoom).

It's only possible to drag from an area with several viewers. It is possible to drop to any area (main window or second screen), irrespective of the drag area.

Figure 13. Multiple viewers



lower Value -255

upper Value 255

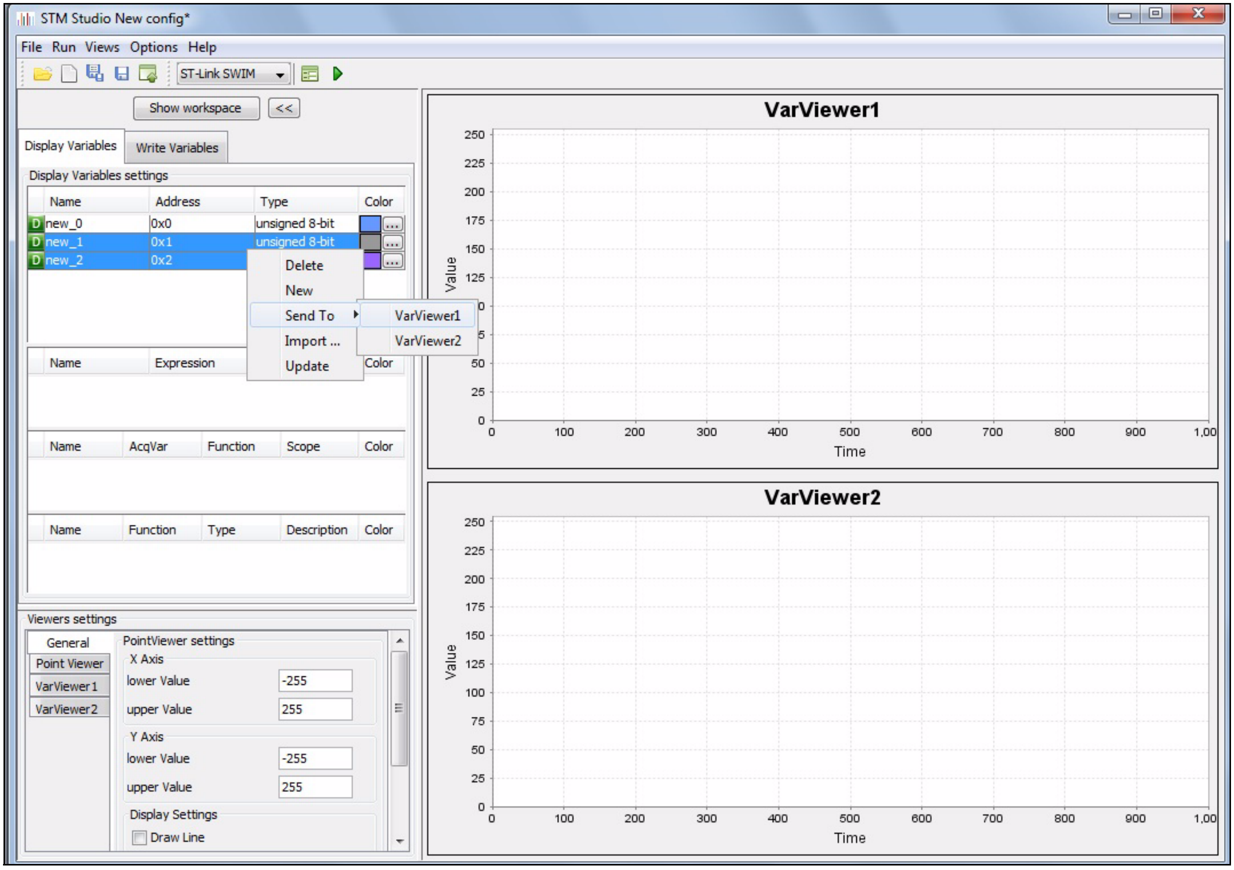
2.3.3 Adding variables to a variable viewer

There are two solutions:

- Drag item from a table of variables and drop it directly either on the variable viewer display or in the list of variables located in the **Viewers settings** pane.
- Use the **Send To** contextual menu in the **Display Variables settings** table.

You can use multiple selections in each of the three variable tables to add several variables at a time in a variable viewer.

Figure 14. Adding variables to a variable viewer



	Name	Address	Type	Color
D	new_0	0x0	unsigned 8-bit	...
D	new_1	0x1	unsigned 8-bit	...
D	new_2	0x2		...

Name	Expression	Color

Name	AcqVar	Function	Scope	Color

Name	Function	Type	Description	Color

X Axis

lower Value: -255

upper Value: 255

Y Axis

lower Value: -255

upper Value: 255

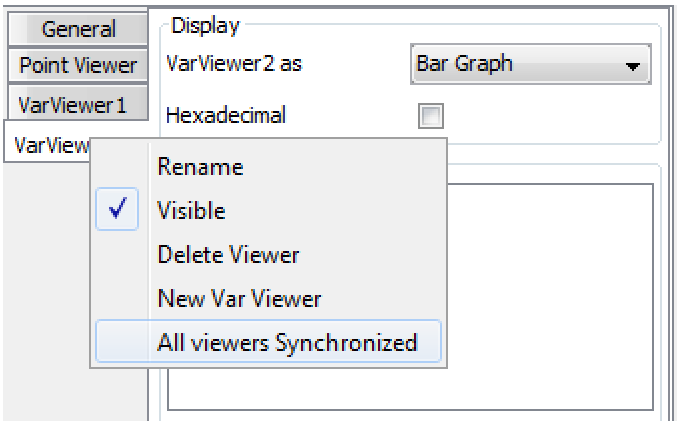
Display Settings

□ Draw Line

2.3.4 Synchronizing viewers

Viewers can be synchronized together by selecting **All viewers Synchronized**: scrolling horizontally in a viewer also affects the other viewers.

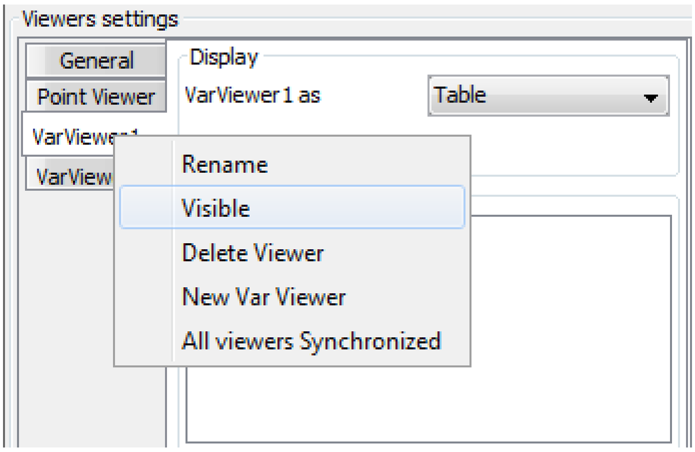
Figure 15. Synchronizing viewers



2.3.5 Hiding viewers

By default, all viewers are visible after creation. It is possible to temporarily hide a viewer (in order to optimize the displayed area) without loosing its configuration: use the contextual menu from the corresponding viewer tab.

Figure 16. Hiding viewers

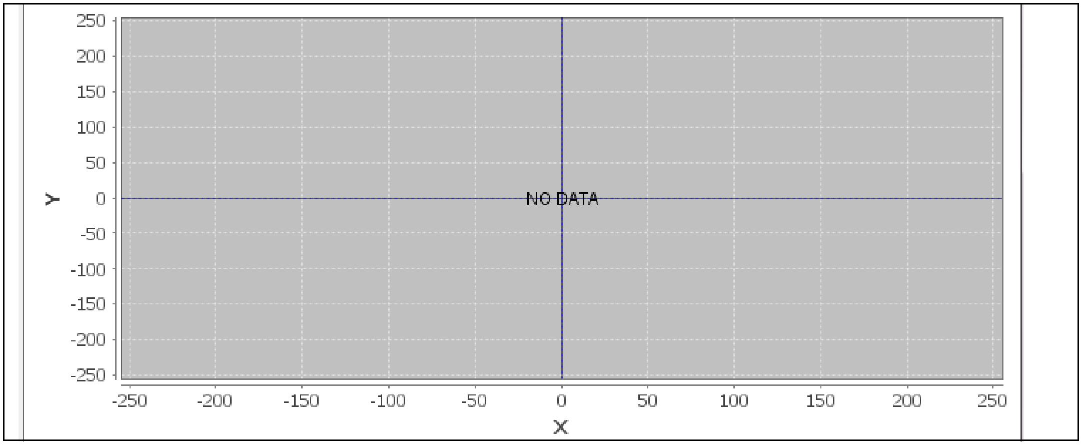


2.4 Using the Point Viewer

The **Point Viewer** is created by default in addition to the default variable viewer. The **Point Viewer** is unique and displays points, not variables. A point is an association of two variables resulting in a screen coordinate: a variable on the X axis and a variable on the Y axis

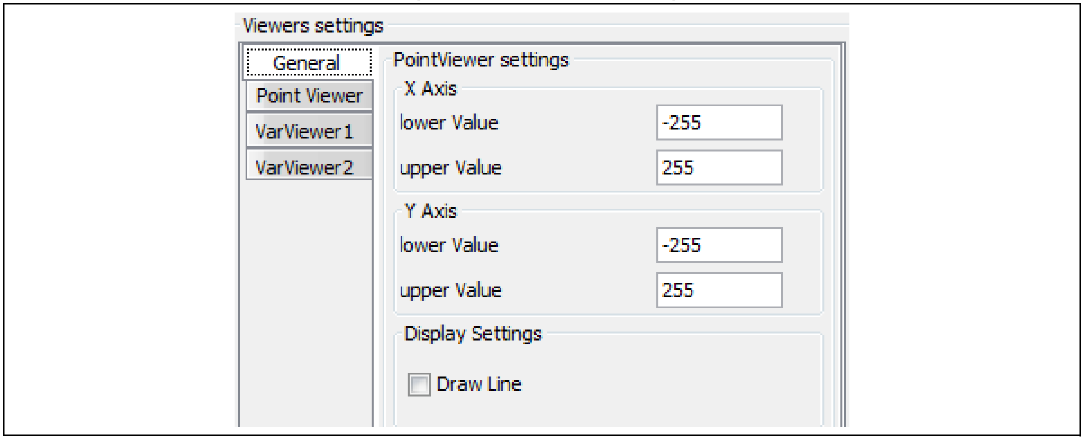
By default, the **Point Viewer** is not displayed. To display it, proceed as for the variable viewers: check the **Point Viewer** item in the **Views** menu or the **Visible** contextual menu in the **Viewers settings** pane.

Figure 17. Point Viewer



2.4.1 Customizing Point Viewer

Figure 18. Point settings



PointViewer settings

X Axis

lower Value -255

upper Value 255

Y Axis

lower Value -255

upper Value 255

Display Settings

□ Draw Line

Use the **General** tab in the **Viewers settings** pane to modify Point Viewer parameters. For both axes, the user can specify the attempted range of values.

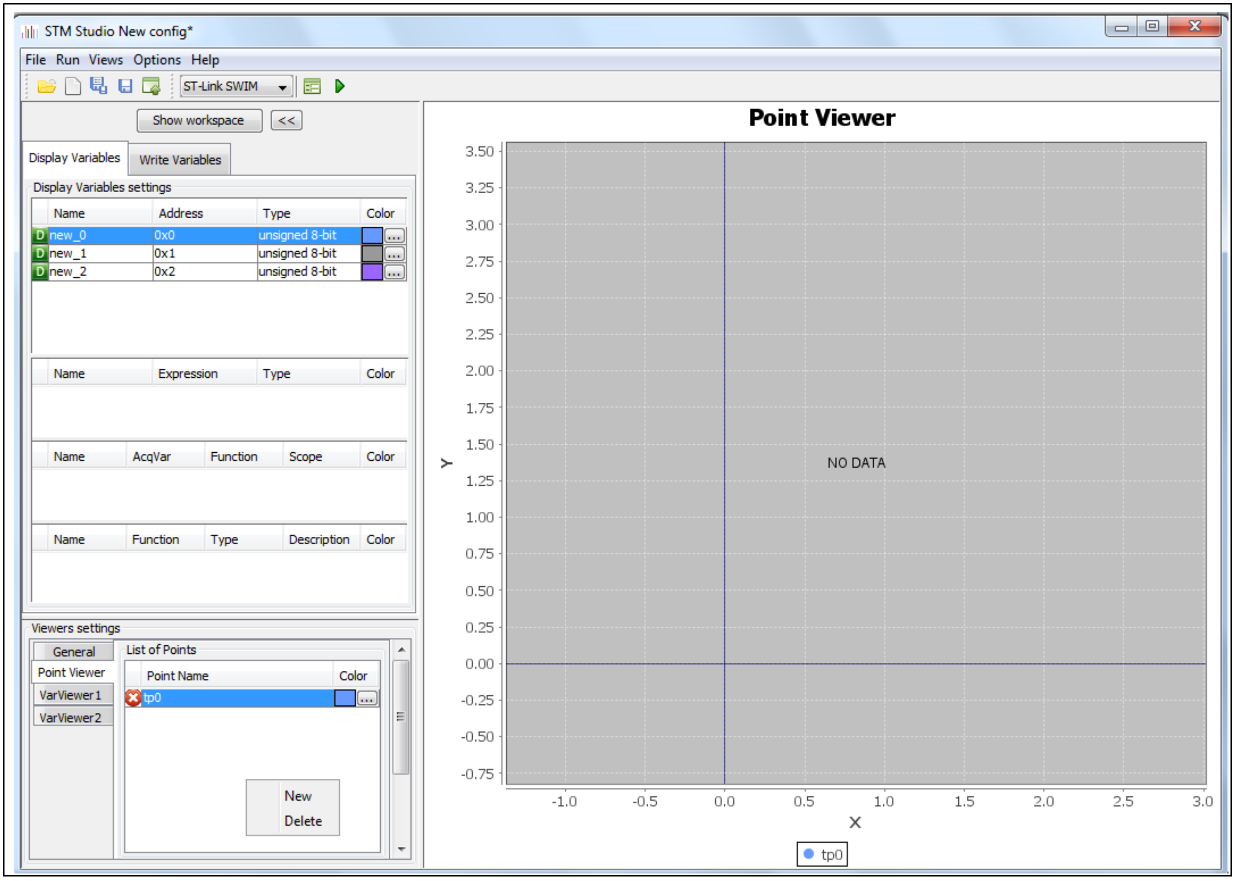
Caution: If the value of the Variable is outside the range of values currently defined for the axis, the point does not appear in the Point Viewer area.

The user can also control the **Draw Line**: when checked, this option draws a curve using all point positions. By default (Draw Line not checked), the points are displayed as screen points.

2.4.2 Adding a new point

To create a new point, select the **Point Viewer** tab in the **Viewers settings** pane, display the contextual menu in the **List of Points** table and click **New**. A new point is added to the list and to the **Point Viewer** legend.

Figure 19. Adding a point



Display Variables settings

Name	Address	Type	Color
new_0	0x0	unsigned 8-bit	...
new_1	0x1	unsigned 8-bit	...
new_2	0x2	unsigned 8-bit	...

Name	Expression	Type	Color

Name	AcqVar	Function	Scope	Color

Name	Function	Type	Description	Color

List of Points

Point Name	Color
☒ tp0	...

The new point has a default name and color, no variable on X and Y (as shown by the red crossed icon visible on *Figure 19*). It can then be customized.

2.4.3 Customizing a point

The point name and color can be modified directly in the list by clicking on the **Name** field and on the **Color** chooser button.

Annotations and variables on X and Y axes can be modified in the **Update Point(s)** pane combo boxes. Click on a point in the list to display the settings for that point; any modification is then registered for the point.

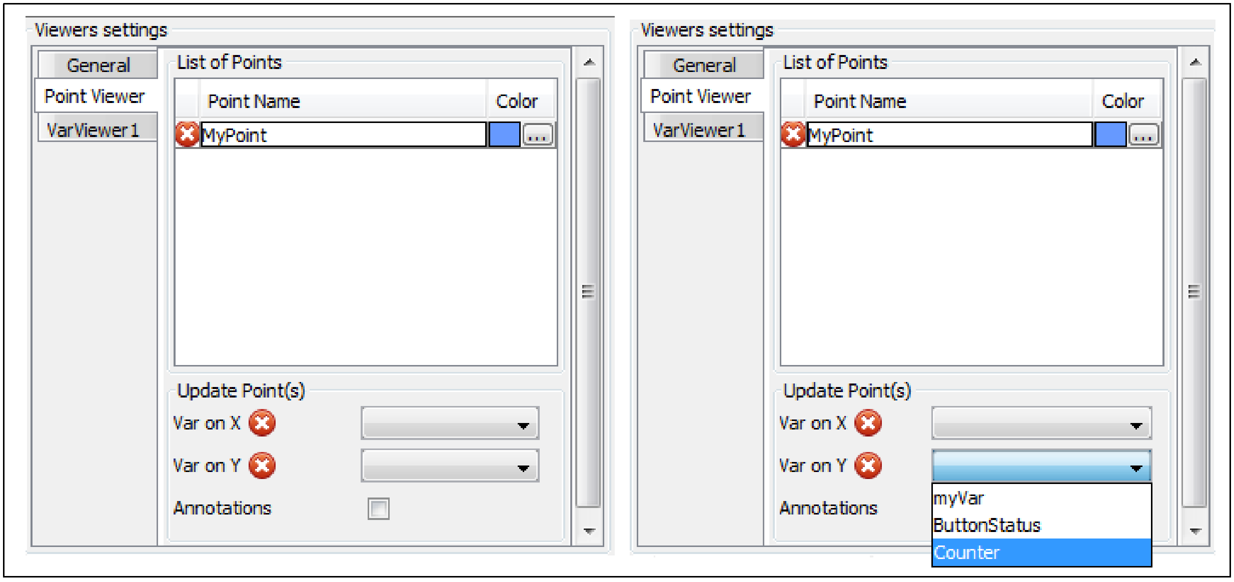
Note: The Point Viewer can display only variables that are acquired in the same mode.

Select **Show Annotations** to display the annotations for the specific point.

Point customizing is mandatory for points newly created, but can be done at any time for existing points either directly in the list for the name and the color or by selecting the point in the list and modifying the combo boxes or the check box.

*Note: When a point has no variable defined on X (or on Y), a red-crossed icon is added to the list (see Figure 20). This is the case after creating a new point, but also after deleting a variable that was referenced by the point. When you select an invalid point, the icon also appears in the **Update Point(s)** pane near the combo box that should be filled. You must then update the point, otherwise it is not displayed during the next visualization session.*

Figure 20. Customizing point settings



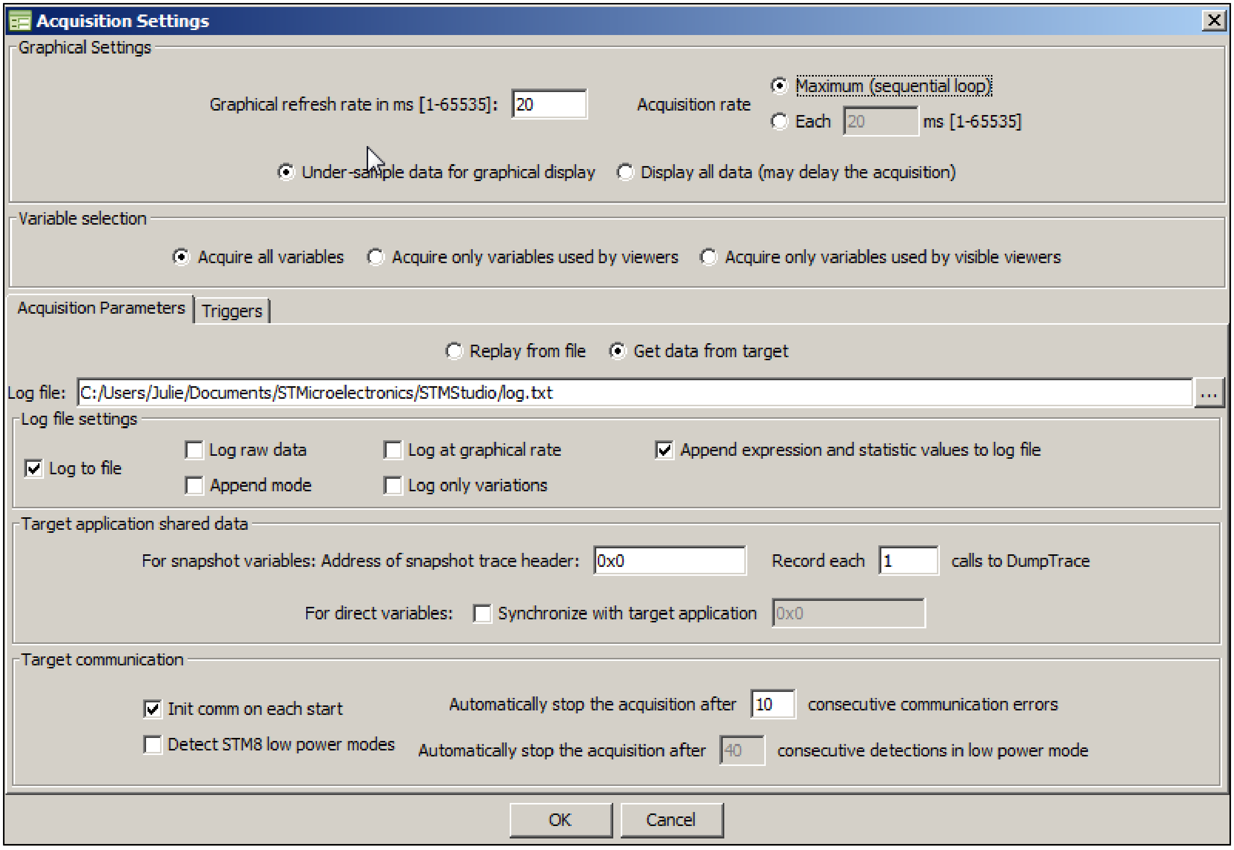
2.5 Configuring the acquisition settings

STM-STUDIO can run in two different modes:

- Replay from file
- Get data from target (Direct or Snapshot acquisition, as defined at the variable's level)

Depending on the selected mode, some other parameters may need to be configured. The acquisition settings are configured through the **Option** > **Acquisition Settings** menu.

Figure 21. Acquisition settings dialog box



Acquisition Settings

Graphical Settings

Graphical refresh rate in ms [1-65535]: 20

Acquisition rate: ● Maximum (sequential loop) ○ Each 20 ms [1-65535]

● Under-sample data for graphical display ○ Display all data (may delay the acquisition)

Variable selection

● Acquire all variables ○ Acquire only variables used by viewers ○ Acquire only variables used by visible viewers

Acquisition Parameters | Triggers

○ Replay from file ● Get data from target

Log file: C:/Users/Julie/Documents/STMicroelectronics/STMStudio/log.txt

Log file settings

☑ Log to file ☐ Log raw data ☐ Log at graphical rate ☑ Append expression and statistic values to log file ☐ Append mode ☐ Log only variations

Target application shared data

For snapshot variables: Address of snapshot trace header: 0x0 Record each 1 calls to DumpTrace

For direct variables: ☐ Synchronize with target application 0x0

Target communication

☑ Init comm on each start Automatically stop the acquisition after 10 consecutive communication errors

☐ Detect STM8 low power modes Automatically stop the acquisition after 40 consecutive detections in low power mode

2.5.1 Replay from file mode

In this mode, data is read from the specified file and no communication with the target is required. The display is not real time: the Graphical refresh rate defines the time lapse between records read from the file, independently from their timestamp saved during the acquisition. Specify the minimum value for the fastest possible replay, or a greater value for a slower replay. For a correct behavior, it is recommended to replay a file in the same configuration as the one used during the acquisition of the data into that file.

The mandatory conditions are:

- **Log raw data** parameter must not be changed between the acquisition and the replay.
- All absolute variables required by the graphical interface during the replay must be present in the log file.
- **Append mode** option must be disabled when saving acquired data into the file.

2.5.2 Get data from target mode

Variable monitoring may be achieved in two different ways, according to the variable configuration:

- Direct acquisition mode - The PC continuously reads data from the connected target. The maximum record size is six frames of 255 bytes. The acquisition is not intrusive for the application, except that the SWIM, JTAG or SWD pins must be reserved for the tool (moreover, on STM8, STM-STUDIO activates the SWIM, which impacts the STM8 behavior after a reset: it remains stalled by the debug module). However, time elapses between the reading of the first variable of a record and the reading of the last one. This can sometimes give a wrong image of the application state. For word variables, it can even lead to unexpected values because MSB and LSB are not read exactly at the same time.
- Snapshot acquisition mode - The application must be instrumented so that variables are sampled on particular application events (user-defined). C source code and project templates are provided in the “softTrace” subdirectory. Simply add DataAcq.h and DataAcq.c files into your project, and call the DumpTrace function where you expect to sample your application state. You may then customize as needed:
  - The maximum number of variables in one record (SNP_TRC_NB_MAX_WORD_ VAR in dataAcq.h): include a safety margin to avoid having to rebuild your application each time you want to trace one more variable. STM-STUDIO computes the actual record size when the acquisition session starts. That number only affects the size of the trace header. STM-STUDIO displays an error message if you try to trace more variables than the application allows.
  - The size of the trace buffer (SNP_TRC_BUFFER_SIZE): this size does not have to be a multiple of SNP_TRC_NB_MAX_WORD_VAR, but it must be able to contain at least two records. A wide buffer is preferred in order to avoid buffer overflow, all the more if your application calls “DumpTrace” frequently.

For correct synchronization with records acquired in direct mode, a timebase must be defined, reflecting the exact time period between 2 consecutive calls to DumpTrace: this is the aim of SNP_TRC_TIMESTAMP_BASE_UNIT and SNP_TRC_TIMESTAMP_VALUE definition in dataAcq.c.

Some parameters can be adjusted for data acquisition from target:

- **Graphical refresh rate**: number of milliseconds between two graphical refreshes. The actual refresh rate depends on computer configuration, system and execution environment (the specified value is used for a system timer). Note that the graphical refresh process runs both concurrently and independently from the acquisition process (the acquisition is in general much faster than the graphical refresh rate).
- **Acquisition rate**: defines the rate at which the host PC reads the target data. When set to the maximum, the acquisition process is a sequential loop limited by the CPU resources given to him by the Operating System. For long acquisition sessions spying slowly changing variables, it might be better to reduce the acquisition rate, thus freeing host CPU resources for other tasks, and also decreasing the log file size (if activated).
- **Under-sample data for graphical display:** in this mode, the acquisition process continues even if the records put at graphical's disposal are still not consumed (displayed). As a result, when the graphical display is slower than the acquisition, some acquired records won't be displayed (but will however be logged to the file if the function is enabled, thus may be displayed later in “replay from file” mode). Conversely,

selecting **Display all data** will make the acquisition process wait for the previous records to be consumed by the graphical interface before going further.

- **Acquire all variables:** in this mode all variables that are described in the Variable Settings control panel are acquired, even if they are not displayed anywhere. This mode may be useful for optimizing the acquisition (to file) with few variables displayed during the acquisition time (all variables may be displayed afterwards in “replay from file” mode).
- **Acquire only variables used by viewers**: acquisition is limited to variables that are displayed in viewers (or used by expressions or statistical variables that are displayed in viewers). In this mode, the acquisition frames will be limited to the strict minimum required for display (but of course, the log file won't contain data for variables not displayed during the acquisition).
- **Acquire only variables used by visible viewers**: only variables that are displayed in the visible viewers are acquired from the target. The list of acquired variables is determined by the viewers' visibility state at the start of the acquisition session, so changing the viewer visibility during the acquisition session has no effect on the list of acquired variables. The change will only be taken into account on the next "start".

Note: If you want to change the visibility of viewers during the acquisition session, select the mode ***Acquire only variables used by viewers****.*
If you also want to add variables to viewers during the acquisition session, choose the mode ***Acquire all variables****.*

- **Log file**: name of the file in which data is recorded (when the Log to file option is selected), or read from (when the Replay from file option is selected).
- **Log to file**: save the data read from the target into the specified log file. Note that the log file may contain more records than displayed during the acquisition session; all records may be displayed afterwards using the Replay from file option.
- **Log raw data**: store raw data as it is read from the target (whole frames, without type computation).
- **Append mode**: when unchecked, the log file is cleared each time the visualization is started.
- **Log at graphical rate**: in general, the acquisition rate is higher than the graphical refresh rate. Check this control to record at the rate used to refresh the display. This may be useful when launching relatively long acquisition sessions, in order to avoid getting too big a file at the end.
- **Log only variations**: only logs records that differ from the previous one.
- **Append expression and statistic values to log file**: when enabled, if any expression or statistic variable is defined in the project, a post-processing phase takes place each time the acquisition session is stopped. During this phase, the log file is completed with values of expression and statistic variables. If disabled, the log file contains only the data read from the application.
- **Init comm on each start**: initialize the communication each time the visualization is started. If selected, the target communication is also closed at the end of each acquisition session (allowing another tool to connect without exiting STM-STUDIO). On STM8, this means that after each “stop” event, the application restarts with SWIM disabled. If the checkbox is disabled, the SWIM remains active on the STM8 target after the end of the STM-STUDIO session.
- **Detect STM8 low power modes:** select this mode in case of STM8 applications spending **lot of time** (several hundreds of milliseconds) in low power mode (HALT, WFE, WFI) **without waking up**. This allows the acquisition process not to try to access

resources (RAM data) that are not available in this mode (and avoid communication errors). In case of applications **often switching** (several times in 100 milliseconds) in low power modes, it is preferable to disable this option and increase the number of allowed consecutive communication errors. As soon as one frame of a record fails, the whole record is skipped (with such kind of applications, it is recommended to regroup all variables to be acquired in the minimal address range, thus increasing the probability to get a full record when the core is awake). Note that it will be all the more difficult to get records as the core is more rarely awake.

- **Address (in user application) of snapshot trace header**: required only if at least one variable is acquired in snapshot mode. Specify the start address of “g_traceHeader” variable in the application (address of “g_traceHeader.startMark[0]”). You can find the address fin the application map file, or obtain it by parsing the application elf file (menu **File>Import variables**).
- **Synchronize with target application**: synchronizes the acquisition from the host with the target application. This acquisition mode requires an instrumented code in the target application (examples for STM8 are in <My Documents>\STMicroelectronics\STMStudio\synchrAcq). The user must:
  - Add enterLock() and exitLock() functions in the application.
  - Specify the address of the stm_studio_flag[0] variable in the **Acquisition settings** window.

  Compared with the snapshot acquisition mode, the synchronized acquisition mode uses less RAM and Flash resources, but is much more time intrusive on the application side (possible waiting loops).

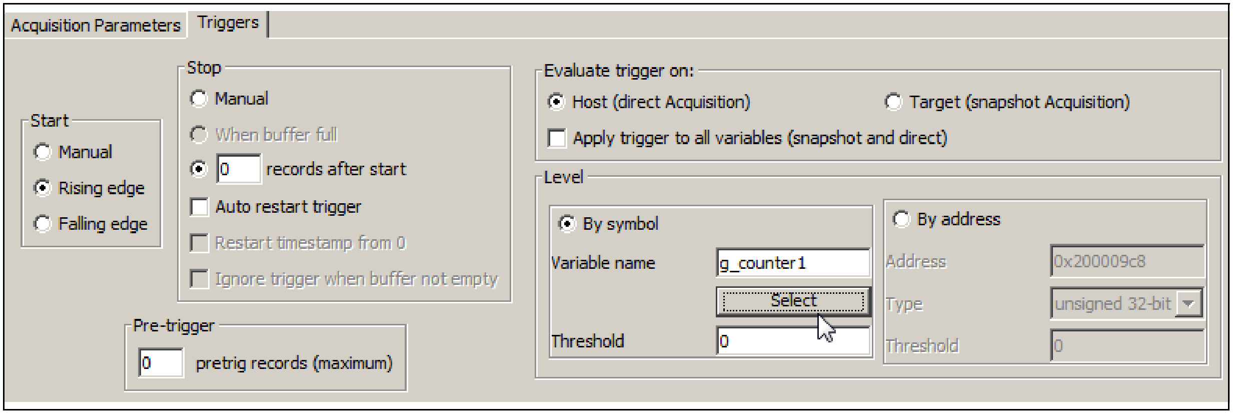
Note: Snapshot mode is preferable for targets that have enough RAM and Flash resources.

- **Record each N calls to DumpTrace**: allows to sub-sample the acquisition in Snapshot mode. N=1 by default means that a record is stored each time dumpTrace is called. Other positive values allow to reduce the acquisition rate (in order to avoid buffer overflows for instance) without rebuilding the application.

2.5.3 Trigger configuration

The STM-STUDIO provides the possibility of triggering the acquisition (in Snapshot mode) or the display (in Direct acquisition mode) on one application event (when getting data from target). It is also possible to define an end-of-acquisition condition.

Figure 22. Trigger parameters



Start
○ Manual
● Rising edge
○ Falling edge

Stop
○ Manual
○ When buffer full
● 0 records after start
□ Auto restart trigger
□ Restart timestamp from 0
□ Ignore trigger when buffer not empty

Pre-trigger
0 pretrig records (maximum)

Evaluate trigger on:
● Host (direct Acquisition) ○ Target (snapshot Acquisition)
□ Apply trigger to all variables (snapshot and direct)

Level
● By symbol
Variable name: g_counter1
Threshold: 0

○ By address
Address: 0x200009c8
Type: unsigned 32-bit
Threshold: 0

Trigger acquisition mode

By default the trigger event (start and/or stop) applies to the mode where it is evaluated. In the case of mixed mode displays, the other mode is managed manually (start/stop on user actions).

Selecting the option **Apply trigger to all variables** overrides this default behavior and synchronizes both modes to the trigger evaluation (no data is displayed before the start event, and no data is displayed after the stop event). Note however that because acquisition modes are not synchronous, there might be a delay between the records acquired in different modes. It might also be possible, for instance, that no record is displayed in direct mode because the trigger set in snapshot mode started and stopped in the same acquisition loop.

Start condition

The acquisition behavior depends on the start condition:

- **Manual**: The acquisition immediately starts after starting the recording session.
- **Rising edge**: the acquisition starts when a sequence of two consecutive records is found where:
  - In the first record, the trigger value is strictly below a specified threshold.
  - In the second record, the trigger value is equal or greater than the specified threshold.
- **Falling edge**: the acquisition starts when a sequence of two consecutive records is found where:
  - In the first record, the trigger value is strictly greater than a specified threshold.
  - In the second record, the trigger value is equal or lower than the specified threshold.

Note: *Because the tool sampling rate may be much slower than the application's variable variation rate, it is possible for the tool to miss a condition.*

Stop condition

- **Manual**: The acquisition stops on user's request (stop recording session).
- **When buffer full**: Available only for trigger defined in Snapshot acquisition. In this case, the recording stops once the trace buffer allocated in the application is full (no buffer overwrite).
- **N records after start**: Specify the number of records to acquire after the trigger condition was hit (N=1 will display only the record where the trigger condition was met).
- **Auto restart trigger**: Available only when start and stop conditions are different from “manual”, and pre-triggering is disabled. In this case, the trigger is rearmed after the stop condition, and the start condition will be evaluated again.
- **Restart timestamp from 0**: Available only for trigger defined in snapshot acquisition, and when “auto restart trigger” is enabled. If enabled, the display is refreshed each time a new trigger event is hit, and the trigger event record will be assigned with timestamp=0. If disabled, the timestamp continues to increment between consecutive trigger events, and several trigger events may be displayed during the same acquisition session.
- **Ignore trigger when buffer not empty**: Available only for trigger defined in Snapshot acquisition, and when “auto restart trigger” is enabled. If enabled, the trigger is not immediately rearmed after the stop condition, but only when the trace buffer allocated

in the application is completely empty (acquired records have been displayed). This allows to avoid having a buffer overflow too soon after the start condition. If disabled, the trigger is immediately rearmed, which means that the trace recording may start again (because start condition met again) before the trace buffer of the previous acquisition is flushed. As a result, a buffer overflow (suspending the recording) may occur quickly after the trigger event, reducing the acquisition interest.

Pre-triggering

The pre-triggering allows to keep few records before the start condition is met.

- **N pretrig records**: Available only when the start condition is different from “manual”. The specified value is a maximum, which means that it is possible for the start condition to be met before all pretrig records have been acquired. In that case, the start condition is not ignored but less pretrig records are displayed. In snapshot acquisition mode, N must not exceed the size of the embedded trace buffer (in number of records) minus 2. In Direct acquisition mode, it must not exceed 200.

Note: Pre-triggering is not compatible with “auto-restart” function. Specifying a value different from 0 pretrig records will disable the “auto restart” function even if selected in the graphical interface.

Level

The trigger threshold may be defined by either of the following actions:

- Using a symbol from the application executable file (or linear expression based on application variable)
- Specifying the address location and access type of the trigger variable to be used.

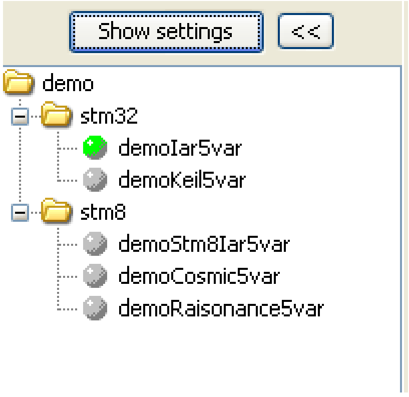
2.6 Project and configurations

STM-STUDIO enables you to save and reload configurations, so that you do not have to reconfigure the entire environment each time STM-STUDIO is launched.

Configurations may be grouped into project, allowing to organize them and easily switch thanks to the workspace panel.

At any time, you can save a configuration, or load or create a new one.

Figure 23. Workspace panel

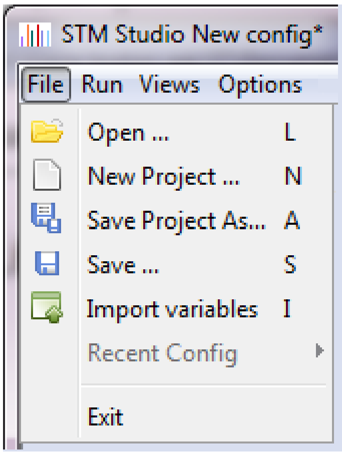


A configuration includes:

- the hardware that was in use the last time the configuration was saved
- the logging settings
- the list of variables with their settings
- the list of variable viewers with their settings and the list of variables they are in charge of displaying
- the settings for the Point Viewer and the list of points the viewer declares, with their settings

All configuration operations are located in the **File** menu.

Figure 24. Configuration operations in File menu



- **Open**: open a project (tsp) or configuration (tsc) file.
- **New Project**: close the current project and open a default project with an empty configuration.
- **Save Project As**: save the current project under a new name (new tsp file).
- **Save**: save the current project (tsp file) along with its active configuration (tsc).
- **Recent**: load a recently used project or configuration.

The STM-STUDIO caption displays the active configuration. When STM-STUDIO is run, the configuration is called “New config” by default. After saving or loading the configuration, the caption title is updated to display the new active configuration.

As soon as a setting is modified, a “*” is added to the active configuration name to signify that the configuration has been modified and that the user will be asked later to save the changes.

From the Workspace panel, it is possible to organize configurations into virtual folders (fully independent from physical directories on disk).

Available actions from folders are:

- **Add new folder**: add a new sub-folder to the currently selected one.
- **Add new configuration**: add a new (empty) configuration item to the currently selected folder. This configuration will have to be selected before being modified and saved.
- **Remove from project**: remove the selected folder, all its sub-folders and configurations. Note that configuration files are not deleted from the disk; only their

references are removed from the project file. It is not possible to remove the project root folder.

- **Import existing configuration**: add a configuration item in the currently selected folder, making reference to an already existing configuration file (tsc).
- **Import configurations from the recent list**: import, in the currently selected folder, all configurations listed in the recent list. Note that this does not import projects (tsp files).

Available actions from inactive (grey LED) configuration are:

- **Remove from project**: remove, from the current folder, the reference to this configuration. The file (tsc), if existing, is not deleted from the disk.
- **Select**: set this configuration as active. Equivalent to a double-click. Before acting, STM-STUDIO will ask for saving the previous active configuration if it has been modified.

Available actions from active (green LED) configuration are:

1. As long as no file from the disk is associated with this configuration item:

- **Save As**: create a configuration (tsc) file on disk and save the configuration settings. The first time, the filename is used as the name for the configuration item. The configuration item may be renamed (without affecting the associated filename) with the “F2” key.
- **Remove from project**: remove the configuration item from the folder.
- **Import existing config**: associate an existing tsc file on disk to this configuration item. This will affect the configuration item name, that may be renamed afterwards with the “F2” key.

2. Once this configuration item has been saved on disk:

- **Save**: save all settings in the configuration file on disk (will ask for overwrite confirmation).
- **Save As**: save all settings in another configuration file on disk (create if not existing, otherwise ask for overwrite confirmation). Note that the previous configuration file on disk will remain, in its last saved state.

You can use the mouse to move folders and configuration items (drag and drop), and to change the active configuration (double click).

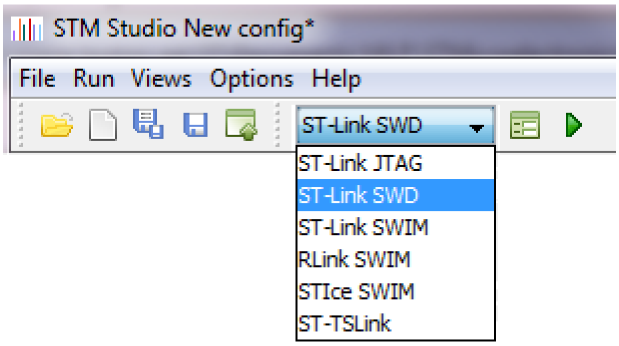
2.7 Running a visualization session

To illustrate a visualization session, a predefined configuration is used with the **Point Viewer** and four variable viewers (Slide, Wheel, Keys) and several variables (slide, wheel, k1, k2 and others).

2.7.1 Starting a session

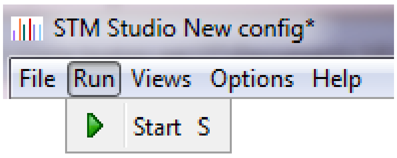
1. First use the toolbar combo box to select the hardware and the protocol to be used for the data acquisition.

Figure 25. Selecting hardware and protocol



2. Then start the visualization using the **Run>Start** menu.

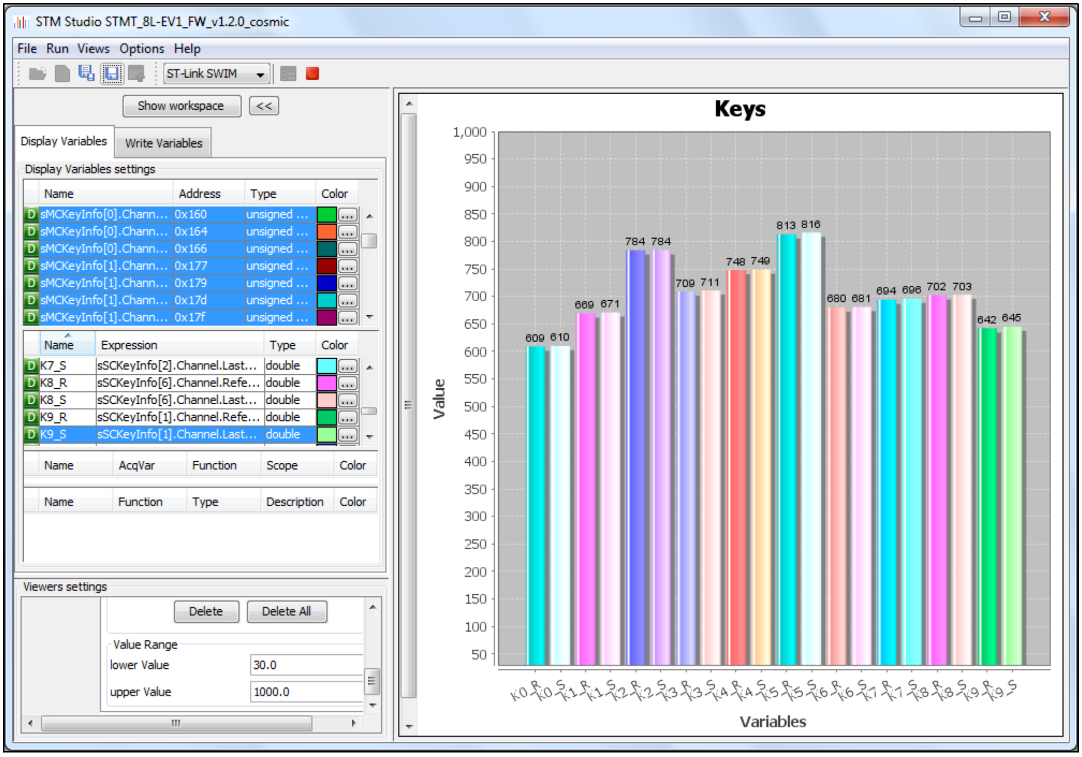
Figure 26. Start menu



2.7.2 Variable visualization

Figure 27 shows an example screen with a demo running.

Figure 27. Variable visualization



STM Studio STMT_8L-EV1_FW_v1.2.0_cosmic

Display Variables settings

	Name	Address	Type	Color
D	sMCKeyInfo[0].Chann...	0x160	unsigned ...	🟩 ...
D	sMCKeyInfo[0].Chann...	0x164	unsigned ...	🟧 ...
D	sMCKeyInfo[0].Chann...	0x166	unsigned ...	🟦 ...
D	sMCKeyInfo[1].Chann...	0x177	unsigned ...	🟥 ...
D	sMCKeyInfo[1].Chann...	0x179	unsigned ...	🟦 ...
D	sMCKeyInfo[1].Chann...	0x17d	unsigned ...	🟦 ...
D	sMCKeyInfo[1].Chann...	0x17f	unsigned ...	🟪 ...

	Name	Expression	Type	Color
D	K7_S	sSCKeyInfo[2].Channel.Last...	double	🟦 ...
D	K8_R	sSCKeyInfo[6].Channel.Refe...	double	🟪 ...
D	K8_S	sSCKeyInfo[6].Channel.Last...	double	🟥 ...
D	K9_R	sSCKeyInfo[1].Channel.Refe...	double	🟩 ...
D	K9_S	sSCKeyInfo[1].Channel.Last...	double	🟩 ...

	Name	AcqVar	Function	Scope	Color

	Name	Function	Type	Description	Color

Viewers settings

Value Range

lower Value: 30.0

upper Value: 1000.0

During the visualization, you may:

- Visualize another viewer from the **Views** menu (see *Figure 28*).
- Show all viewers using the **All visible** menu from the viewer settings **General** tab (see *Figure 29*).
- Change the display of a given viewer select the **Keys** tab and changing the **Display Mode** in the combo box.
- Add an existing variable to a viewer, for example by dragging the k3 variable from the table and dropping it over the wheel viewer in the display area.
- Add new variable viewers and fill them with variables that are already in acquisition, or based on variables already in acquisition (which is restrictive when in acquisition mode “Only used variables in acquisition”).

Figure 28. Visualizing different viewers

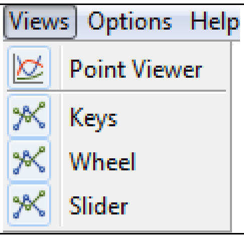
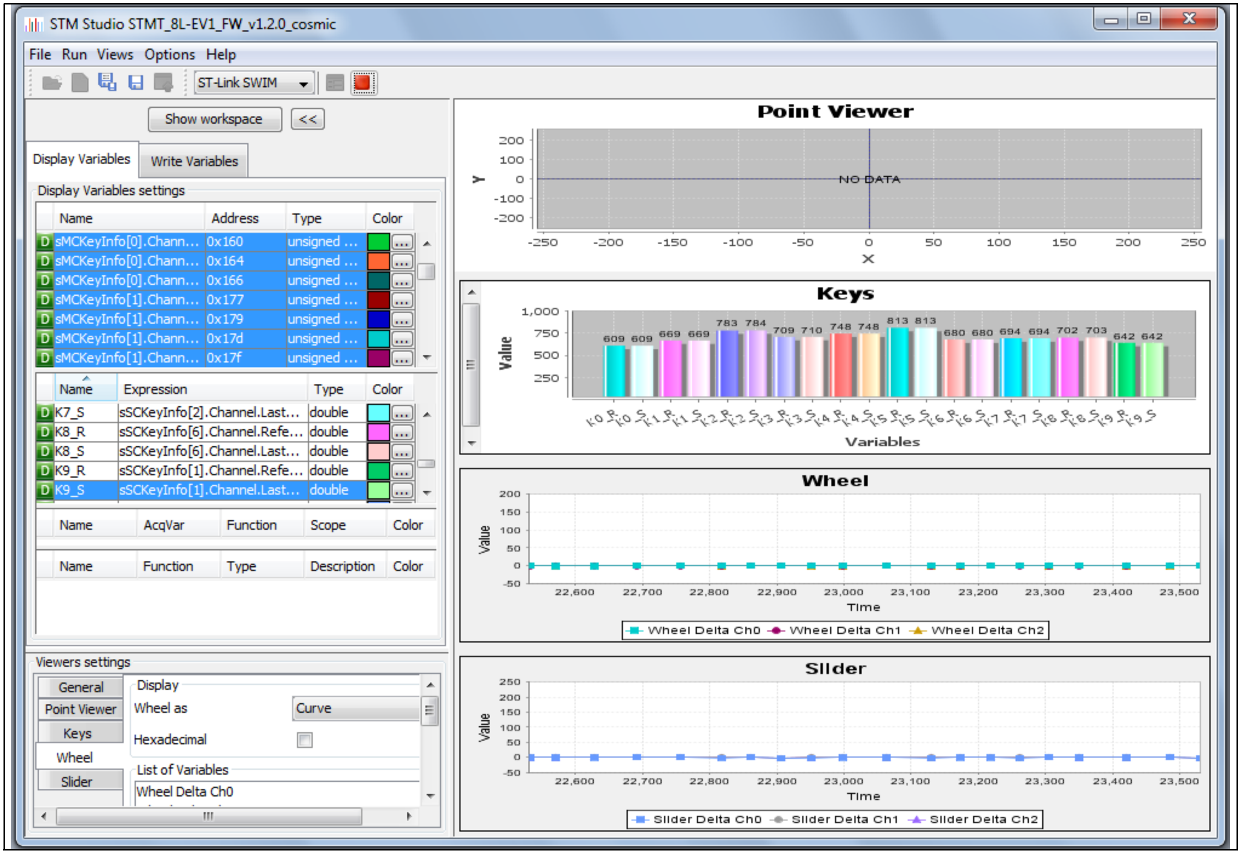


Figure 29. Showing all viewers



STM Studio STMT_8L-EV1_FW_v1.2.0_cosmic

File Run Views Options Help

ST-Link SWIM

Show workspace <<

Display Variables | Write Variables

Display Variables settings

	Name	Address	Type	Color
D	sMCKeyInfo[0].Chann…	0x160	unsigned …	🟩
D	sMCKeyInfo[0].Chann…	0x164	unsigned …	🟧
D	sMCKeyInfo[0].Chann…	0x166	unsigned …	🟩
D	sMCKeyInfo[1].Chann…	0x177	unsigned …	🟥
D	sMCKeyInfo[1].Chann…	0x179	unsigned …	🟦
D	sMCKeyInfo[1].Chann…	0x17d	unsigned …	🟦
D	sMCKeyInfo[1].Chann…	0x17f	unsigned …	🟪

	Name	Expression	Type	Color
D	K7_S	sSCKeyInfo[2].Channel.Last…	double	🟦
D	K8_R	sSCKeyInfo[6].Channel.Refe…	double	🟪
D	K8_S	sSCKeyInfo[6].Channel.Last…	double	🟥
D	K9_R	sSCKeyInfo[1].Channel.Refe…	double	🟩
D	K9_S	sSCKeyInfo[1].Channel.Last…	double	🟩

Name	AcqVar	Function	Scope	Color

Name	Function	Type	Description	Color

Viewers settings

General / Point Viewer / Keys / Wheel / Slider

Display

Wheel as: Curve

Hexadecimal ☐

List of Variables

Wheel Delta Ch0

Some settings can also be changed during the visualization:

- Variables: color
- Variable viewers
  - Value range
  - Bar graph/curve display
  - Maximizing
- Point Viewer
  - Add points
  - Switch between draw mode and point mode
  - Resize axes

During visualization, you may not:

- For variables:
  - Create a new variable
  - Change the address of a variable
  - Change the type of a variable
  - Change the name of a variable
- For a point:
  - Change variables on X and Y

2.7.3 Writing variables on the fly

During the visualization session, you may write data on the fly. STM-STUDIO provides two ways of doing this, depending if the variable to be written into is also read (in acquisition) or not.

For "write only" variables

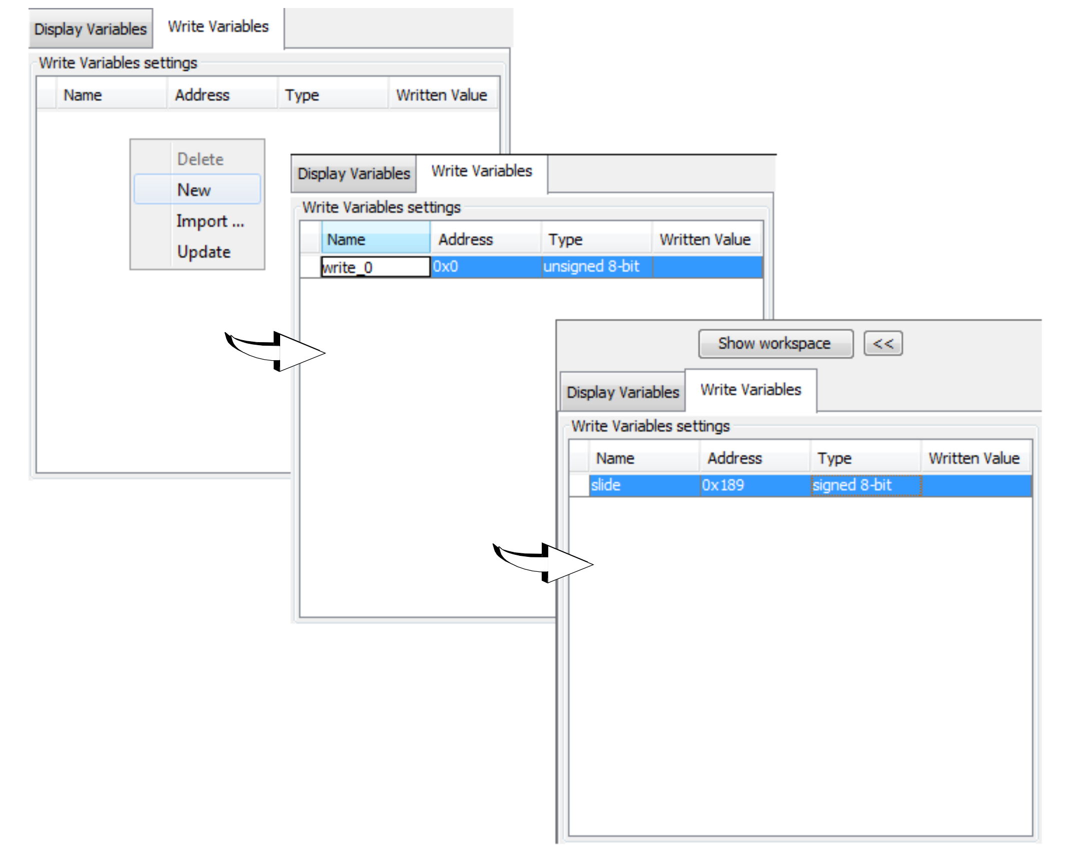
For "write only" variables, prefer using the **Write Variables** tab. Proceed as follows:

First create a write variable in the **Write Variables** tab in the configuration pane. Either one of three methods can be used:

- Drag an absolute variable from the **Display variable** tab to the **Write Variables** tab.
- Use the **Write Variable** tab contextual menus (New or Import) in the same way as for adding absolute variables (*Figure 30*).
- Select “Import variables from executable” then the “Add variables to the write variables table” mode.

It is also possible to define as a Write Variable a linear expression based on an absolute variable (from the application). In this case, the STM-STUDIO reverses the expression before affecting the absolute variable in the application. The drag-and-drop from **Display Variables** tab to **Write Variables** tab is the only way to add an expression to the Write variable table.

Figure 30. Adding a write variable

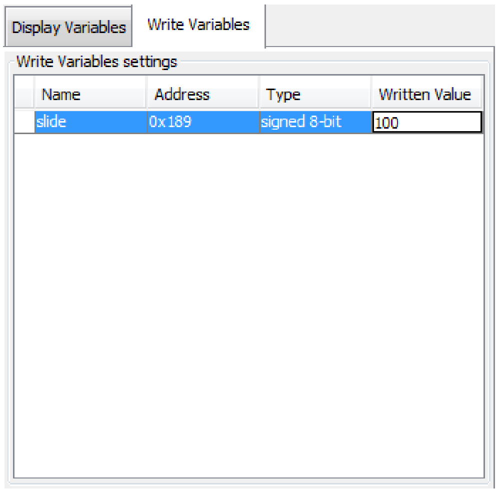


Name	Address	Type	Written Value
write_0	0x0	unsigned 8-bit	

Name	Address	Type	Written Value
slide	0x189	signed 8-bit	

Then write a value at the variable address: click on the **Write Value** field, enter the value, press Return (see *Figure 31*). Note that a shorter way is to put the mouse pointer on the **Write Value** field, to enter the value and to press Return.

Figure 31. Writing on the fly



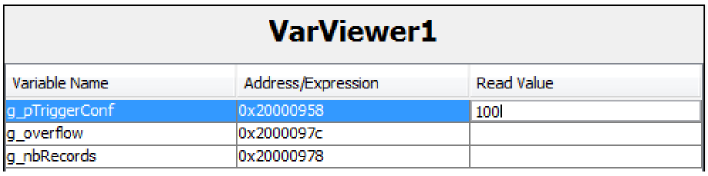
Name	Address	Type	Written Value
slide	0x189	signed 8-bit	100

The last written value remains displayed, even if the value changes on the application side (there is no read back from the write table).

For "read/write" variables

For "read/write" variables, it may be more simple to use a VarViewer in the **Table** display mode. In this mode, the **Read Value** column may be edited and a write command is sent when pressing <Enter> or when the edited cell loses the focus. Press <Esc> in order to cancel the edition without initiating a write command.

Figure 32. Editing a value from the Read Value column



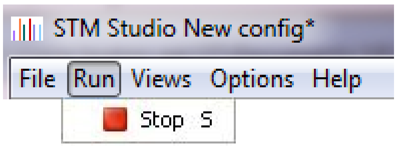
Variable Name	Address/Expression	Read Value
g_pTriggerConf	0x20000958	100
g_overflow	0x2000097c	
g_nbRecords	0x20000978	

Writing from the VarViewer table works for application variables as well as for linear expressions based on a single application variables In that case, the STM-STUDIO reverses the linear expressions before writing to the application variable. It is not possible to write to expressions that are not linearly-dependent on a single application variable.

2.7.4 Post-visualization analysis

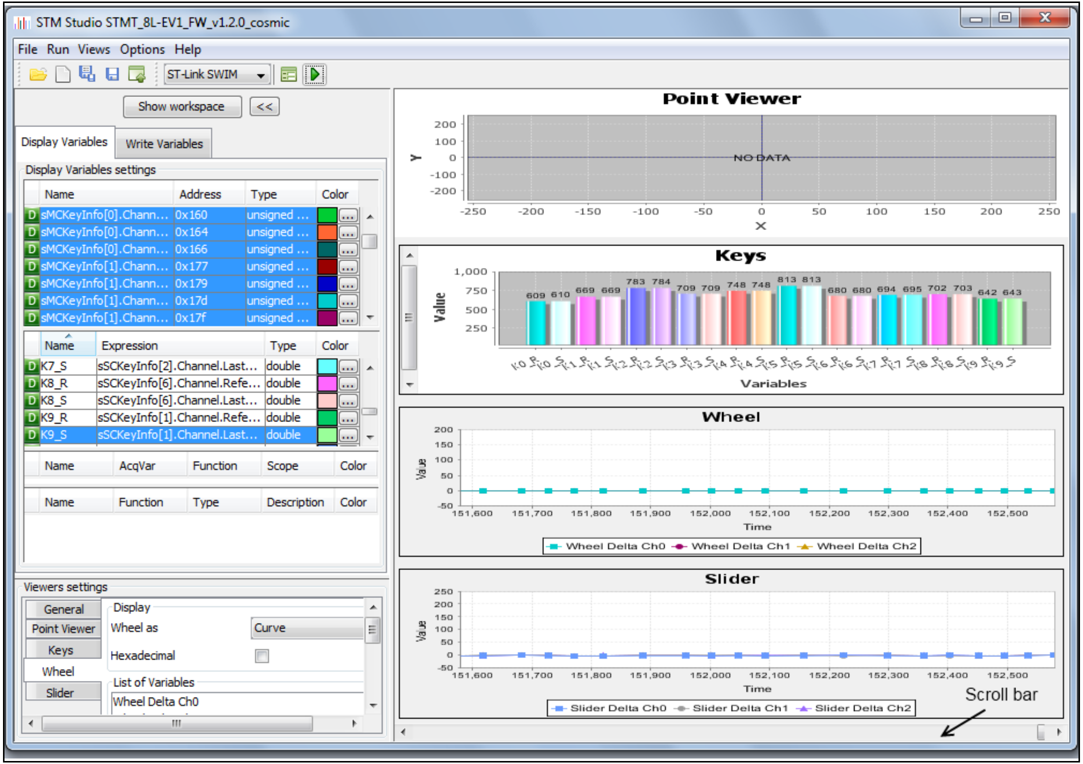
To stop the visualization: click on the **Run/Stop** menu (*Figure 33*).

Figure 33. Stop menu



- Horizontal scroll bars are added to enable the user to replay the whole visualization on both variable viewers (*Figure 34*) and Point Viewer.

Figure 34. Replaying variable viewer visualization



	Name	Address	Type	Color
D	sMCKeyInfo[0].Chann…	0x160	unsigned …	🟩
D	sMCKeyInfo[0].Chann…	0x164	unsigned …	🟧
D	sMCKeyInfo[0].Chann…	0x166	unsigned …	🟩
D	sMCKeyInfo[1].Chann…	0x177	unsigned …	🟥
D	sMCKeyInfo[1].Chann…	0x179	unsigned …	🟦
D	sMCKeyInfo[1].Chann…	0x17d	unsigned …	🟦
D	sMCKeyInfo[1].Chann…	0x17f	unsigned …	🟪

	Name	Expression	Type	Color
D	K7_S	sSCKeyInfo[2].Channel.Last…	double	🟦
D	K8_R	sSCKeyInfo[6].Channel.Refe…	double	🟪
D	K8_S	sSCKeyInfo[6].Channel.Last…	double	🟥
D	K9_R	sSCKeyInfo[1].Channel.Refe…	double	🟩
D	K9_S	sSCKeyInfo[1].Channel.Last…	double	🟩

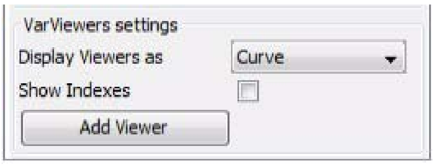
Name	AcqVar	Function	Scope	Color

Name	Function	Type	Description	Color

Wheel as: Curve
Hexadecimal □

- To provide a better Time axis reading on variable viewers, sliders can be added to popup the time value (select the VarViewers settings **General** tab and set **Show Indexes**: *Figure 35*).

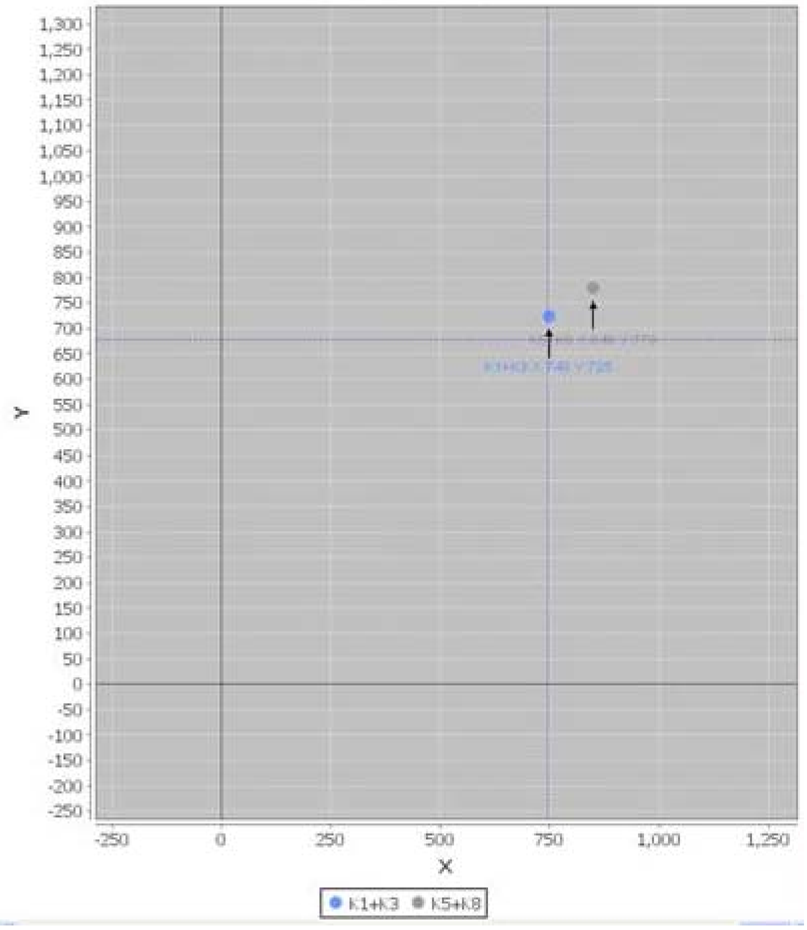
Figure 35. Showing indexes checkbox



Display Viewers as: Curve
Show Indexes □

- For Point Viewer in Point mode, click **Show Annotations** to display coordinates (*Figure 36*).

Figure 36. Showing annotations



Appendix A Syntax for expression variables

The parser supports basic mathematical expressions using acquisition variables. Constants may be expressed in decimal, or hexadecimal with '0x' prefix. Variables are promoted to doubles before computing. The result of the expression evaluation is also a double that may be used in any VarViewer.

Supported operators are:

+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Power
%	Integer division
=	Equals
<>	Not equals
&	Bitwise AND
|	Bitwise OR
!	Logical NOT
<	Less than
>	Greater than
<=	Less than or equals
>=	Greater than or equals
>>	Right shift
<<	Left shift

In addition to the basic operators, the parser supports few mathematical functions:

SQR	Square function
SIN	Sinus of an angle expressed in radians
COS	Cosinus of an angle expressed in radians
TAN	Tangent of an angle expressed in radians
ATAN	ArcTangent
SINH	Sinus Hyperbolic
COSH	Cosinus Hyperbolic
COTAN	Cotangent
EXP	Exponent

LN	Natural log
LOG	10 based log
SQRT	Square root
ABS	Absolute value
SIGN	SIGN(X) returns -1 if X<0; +1 if X>0, 0 if X=0.
TRUNC	Discards the fractional part of a number. e.g. TRUNC(-3.2) is -3, TRUNC(3.2) is 3.
CEIL	CEIL(-3.2) = -3, CEIL(3.2) = 4.
FLOOR	FLOOR(-3.2) = -4, FLOOR(3.2) = 3.
RANDOM(X)	Generates a random floating point number such that 0 <= Result < X. If X is negative, then result is X < Result <= 0.
RND	RND(X) generates a random INTEGER number such that 0 <= Result < int(X). If X is negative, then result is int(X) < Result <= 0.
INTPOW	The INTPOW function raises Base to an integral power. INTPOW(2, 3) = 8. Note that result of INTPOW(2, 3.4) = 8 as well.
POW	The Power function raises Base to any power. For fractional exponents or exponents greater than MaxInt, Base must be greater than 0.
LOGN	The LogN function returns the log base N of X. Example LOGN(10, 100) = 2.
MIN	MIN(2, 3) is 2.
MAX	MAX(2, 3) is 3.
MOD	MOD(x,y) function implements the Java % (modulus) operator.
IF	The IF(b, case1, case2) function provides branching capability. If b is not 0, then it returns case1, else it returns case2. If b==0 then case1 will not be evaluated, and vice versa.

Revision history

Table 1. Document revision history

Date	Revision	Changes
12-Nov-2010	1	Initial release.
18-May-2011	2	Updated: All screenshots *Section 1.1: JRE installation* *Section 2.1: Overview* *Section 2.2: Creating variables* *Section 2.2.4: Adding plugin variables* *Section 2.4: Using the Point Viewer* *Section 2.6: Project and configurations* *Section 2.7: Running a visualization session* Added: *Section 2.5: Configuring the acquisition settings*
12-Mar-2012	3	Updated: Screenshots updated *Introduction* *Section 2.2: Creating variables* *Section 2.2.4: Adding plugin variables* *Section 2.4: Using the Point Viewer* *Section 2.5: Configuring the acquisition settings* *Section 2.6: Project and configurations* *Section 2.7: Running a visualization session* Added: *Appendix A: Syntax for expression variables* Document reformatted as per new die description template
11-Jun-2012	4	Updated: *Section 1.1: JRE installation* *Section 2.2.2: Adding expression variables* *Section 2.2.4: Adding plugin variables* *Section 2.5: Configuring the acquisition settings* *Section 2.7.3: Writing variables on the fly* *Figure 3: Adding absolute variable directly from table* *Figure 4: Importing absolute variables* *Figure 8: Adding statistical variables* *Figure 21: Acquisition settings dialog box* Added: *Section 2.5.3: Trigger configuration* *Figure 5: Direct acquisition mode* *Figure 6: Snapshot acquisition mode* *Figure 22: Trigger parameters*.

Table 1. Document revision history (continued)

Date	Revision	Changes
15-Jan-2013	5	Updated: *Section 1.1: JRE installation* *Section 1.3: Hardware support* *Section 2.2.2: Adding expression variables* *Section 2.3.3: Adding variables to a variable viewer* *Section 2.3.4: Synchronizing viewers* *Section 2.5.2: Get data from target mode* *Section 2.7.3: Writing variables on the fly* All the figures.
26-Oct-2013	6	Updated: *Section 2.2: Creating variables* *Section 2.2.1: Adding absolute variables* *Section 2.2.4: Adding plugin variables* *Section 2.3.2: Customizing variable viewers* *Section 2.5: Configuring the acquisition settings* *Section 2.5.3: Trigger configuration*

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -
Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com