UM1049

User manual

Unico GUI

Introduction

Unico is a cross-platform graphical user interface (GUI) running on Windows, Linux and Mac OS for the demonstration boards of MEMS sensors such as accelerometers, gyroscopes, magnetometers and environmental sensors available in the STMicroelectronics portfolio.

Unico interacts with all the MEMS demonstration boards supported by the STEVAL-MKI109V3 (professional MEMS tool) motherboard and allows a quick and easy setup of the sensors, as well as the complete configuration of all the registers and the advanced features embedded in the digital output devices. The software visualizes the output of the sensors in both graphical and numeric format and allows the user to save or generally manage data coming from the device.

This user manual describes all the functions of the Unico GUI. For details regarding the features of each sensor, refer to the related device datasheet.

1 PC system requirements

Unico software has been designed to operate with Microsoft® Windows platforms, Linux platforms, and Mac OS X platforms.

1.1 Windows platforms

The package “Microsoft Visual C++ 2010 Redistributable Package (x86)” needs to be installed to be able to run Unico on Windows; it can be downloaded from the Microsoft Download Center.

To install the Unico GUI, launch the “Setup_Unico.exe” file included in the package and follow the instructions which appear on the screen. When the software is installed, you can run it from: “Start > STMicroelectronics > Unico > Unico.exe”

1.2 Linux platforms

For Linux Debian-based distributions, a .deb package is provided in the package.

On Ubuntu, you can install the .deb package by opening a terminal and writing the following command:

sudo dpkg -i unico.deb

If the installation fails for any missing dependent library, it is necessary to install the missing library before proceeding with the Unico installation:

sudo apt-get install <missing_library_name>

sudo dpkg -i unico.deb

After the installation has been completed, the executable file (“unico”) is stored in the /usr/local/bin folder.

To run the Unico software, just type:

/usr/local/bin/unico/unico.sh or click on the icon in the Application launcher menu

Note: The current version of Unico is designed to work on Ubuntu 18.04 LTS, although it should be possible to install and run it on other Debian-based distributions after having installed all the dependencies required.

1.3 Mac OS X platforms

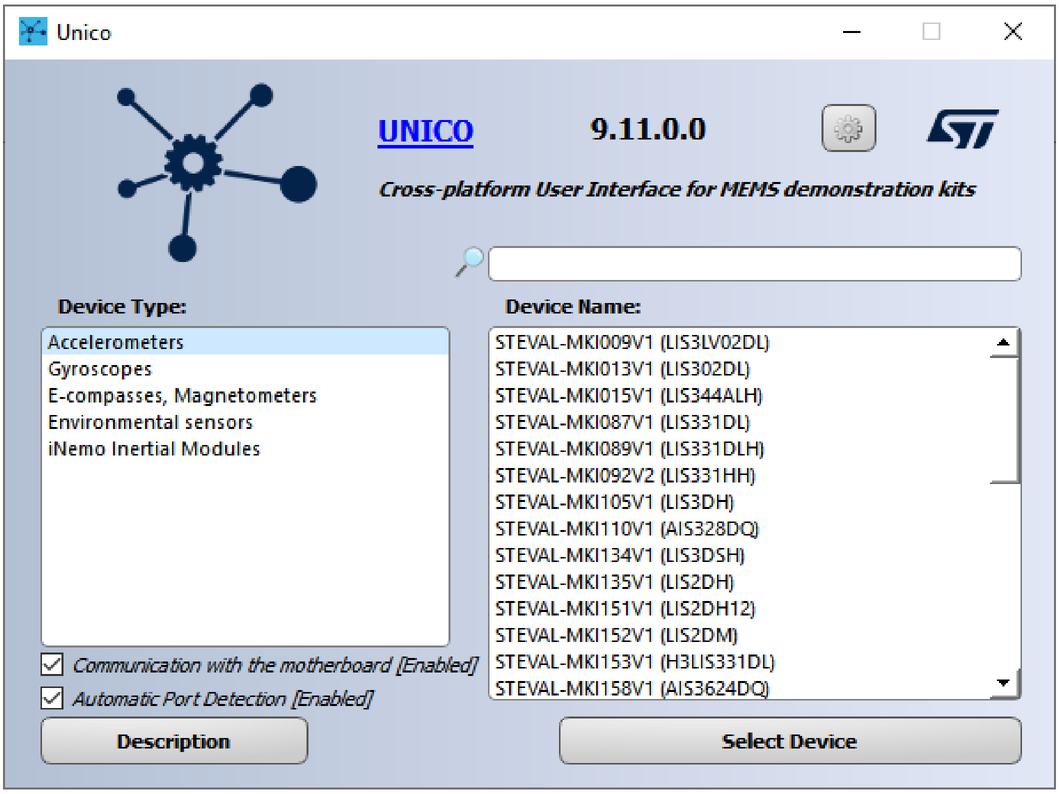
For Mac OS, a PKG file (unico.pkg) is provided in the package.

To install Unico GUI, launch the unico.pkg file and follow the instructions which appear on the screen. When the software is installed, you can run it from [**Applications > unico**].

2 Unico graphical user interface

After launching the Unico GUI, a launcher window appears, as shown in Figure 1. Unico launcher. The GUI shows the list of adapter boards supported by the current release, grouped according to the device type.

Figure 1. Unico launcher



Choose the board currently in use from the list and then click on the button [**Select Device**]. The main window appears after a few seconds (Section 2.1).

In the launcher window, it is also possible to get a brief description of the selected sensor by clicking on the [**Description**] button.

The GUI can automatically detect the port where the board is connected. However, if the user has no administrator rights, or in case of Bluetooth connection, the port selection must be done manually, unchecking the [**Automatic Port Detection**] checkbox. For further information on how to detect the port manually refer to Section 4 Port detection.

On Linux, permissions on the serial port could be required to establish the connection. If so, the GUI suggests the command to be executed (for instance: *sudo chmod 666 ttyACM0*) and opens a terminal to run the command.

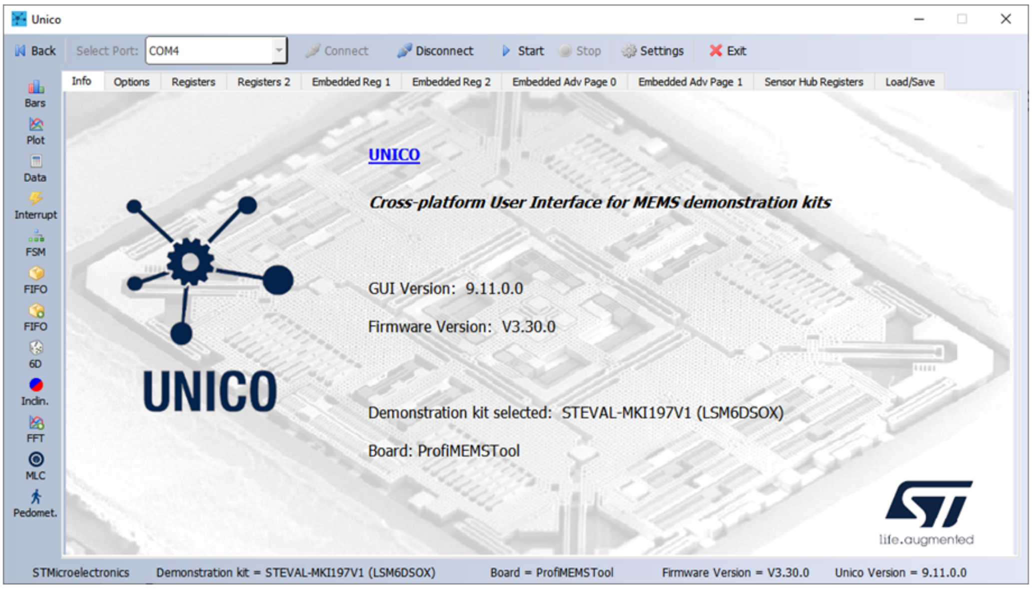
Unchecking the [**Communication with the motherboard**] checkbox, it is possible to run Unico offline. In this case, the GUI cannot communicate with the sensor, so its functionality is limited to the features which do not require interaction with the motherboard.

2.1 Tools available

The Unico main window can be divided in three parts:

1. Main control (Figure 2, tool bar at the top of the window) - connects/disconnects the board, starts/stops data acquisition, exits/returns back for closing the GUI or returning to the launcher.
2. Tab selector (Figure 2, tabs under main control) - is used to toggle between the tabs [**Info**], [**Options**], [**Registers**] and [**Load/Save**]. These tabs are available for all the devices and are used to configure the sensors.
3. Tools (Figure 2, tool bar at the left side of the window) - contains all the tools available for the device in use.

Figure 2. **Unico main window**



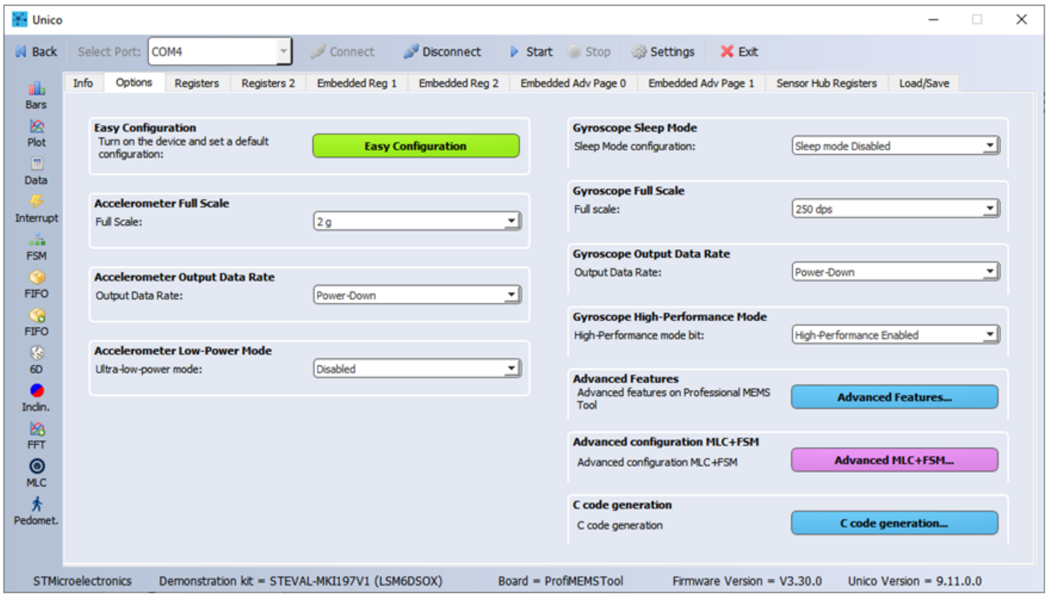
In the following sections all the functions available in Unico GUI are described.

2.2 Options tab

The [**Options**] tab allows the user to control the main parameters of the selected sensor. The content of the tab depends on the sensor chosen. The following parameters refer to a 6-axis IMU device (accelerometer + gyroscope):

1. [**Easy Configuration**]: turns on the device and sets a default configuration
2. [**Accelerometer Full Scale**]: sets the maximum acceleration value measurable by the accelerometer
3. [**Accelerometer Output Data Rate**]: sets the accelerometer output data rate
4. [**Accelerometer Low-Power Mode**]: sets low-power or ultra-low-power modes
5. [**Gyroscope Sleep Mode**]: allows enabling sleep mode for the gyroscope
6. [**Gyroscope Full Scale**]: sets the maximum value measurable by the gyroscope
7. [**Gyroscope Output Data Rate**]: sets the gyroscope output data rate
8. [**Gyroscope High-Performance mode**]: allows enabling high-performance mode for the gyroscope
9. [**Advanced Features**]: opens a page containing advanced features available on the professional MEMS tool board (for example, current measurement, power supply settings, communication protocol settings)
10. [**Advanced configuration MLC+FSM**]: allows merging MLC and FSM configurations into a single .ucf file
11. [**C code generation**]: allows generating C-header files (.h) from configuration files (.ucf)

Figure 3. **[Options] tab**

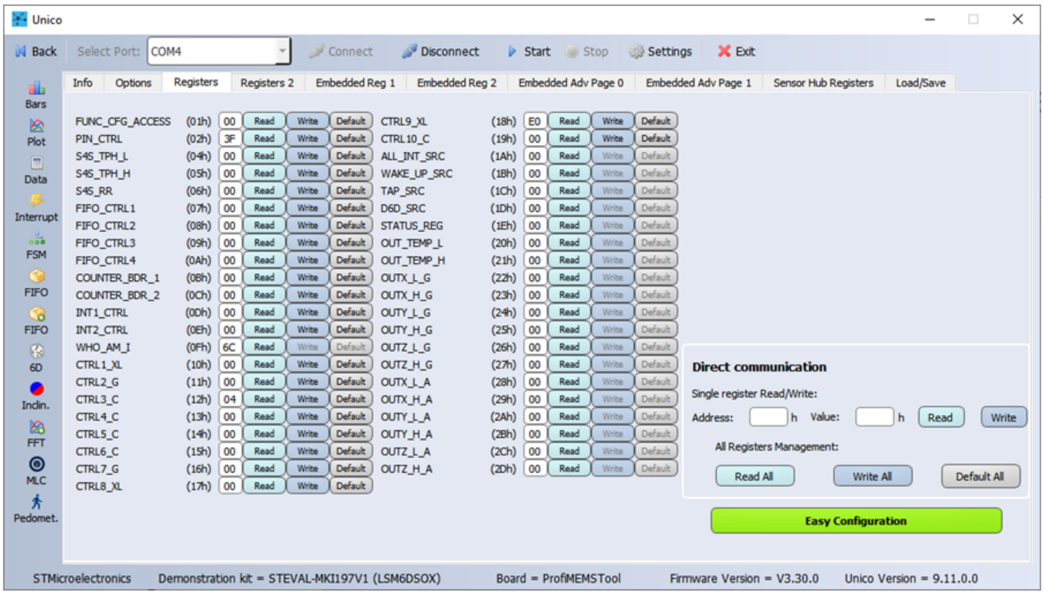


2.3 Registers setup tab

The [**Registers**] setup tab shown in Figure 4. [**Registers**] setup tab allows reading and writing the content of the registers embedded in the MEMS sensor mounted on the demonstration kit. The tab is divided in three sections:

1. General (Figure 4. [**Registers**] setup tab, main page) - provides access to the registers which control the main settings of the device. This section contains control registers, registers to manage the interrupt generation, and so on. It is possible to read and write the contents of each register. To read the default value for a given register, press the [**Default**] button (in this case no data is written in the register, to do this click the [**Write**] button).
2. [**Direct communication**] (Figure 4. [**Registers**] setup tab, group box in the bottom right corner) - provides access to any register in the device. To read a generic register, insert the address value in the [**Register Address**] text box, then click on the [**Read**] button. The retrieved content of the register is displayed in the [**Register Value**] field. As with writing to a register, the user must specify the address and the data to be written inside the fields marked [**Register Address**] and [**Register Value**] respectively, then press the [**Write**] button. [**Read All**], [**Write All**], and [**Default All**] perform the same functions but for all the registers at the same time.
3. [**Easy Configuration**] - this button provides the user the possibility to choose a default configuration, allowing an easy start. When pressed, the sensor register is automatically configured with the default configuration (Figure 4. [**Registers**] setup tab, green button in the bottom right corner).

Figure 4. [Registers] setup tab

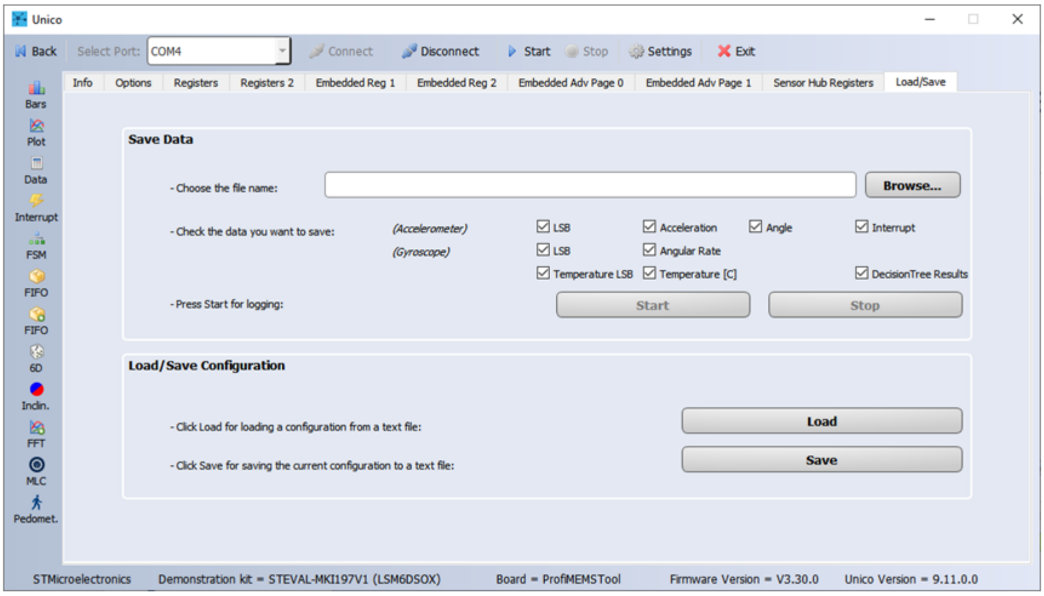


2.4 Load/Save tab

This tab allows the user to save a stream of sensor output data in a text file, available for possible post-processing (Figure 5. [**Load/Save**] tab, [**Save Data**] section). It is possible to select which data must be stored. The [**Browse**] button is used to select the folder and insert the file name, then the [**Start**] and [**Stop**] buttons define the acquisition period.

It is also possible to save the current register configuration by clicking on the [**Save**] button. The configuration saved can be loaded at any time by clicking on the [**Load**] button (Figure 5. [**Load/Save**] tab, [**Load/Save Configuration**] section).

Figure 5. [Load/Save] tab



2.5 Bars tool

The [**Bars**] tool (Figure 6. [**Bars**] tool) displays the data measured by the sensor in a bar chart format. For instance, in the case of a 6-axis module, the accelerations along the X, Y, and Z axes correspond respectively to the red, green, and blue bars. The same colors are used to represent the magnetic values along X, Y, and Z axes.

The height of each bar is determined by the amplitude of the signal measured by the sensor along the related axis. The full scale of the graph depends on the configuration and can be changed through both the option tab (Figure 3. [**Options**] tab) and the register tab (Figure 4. [**Registers**] setup tab).

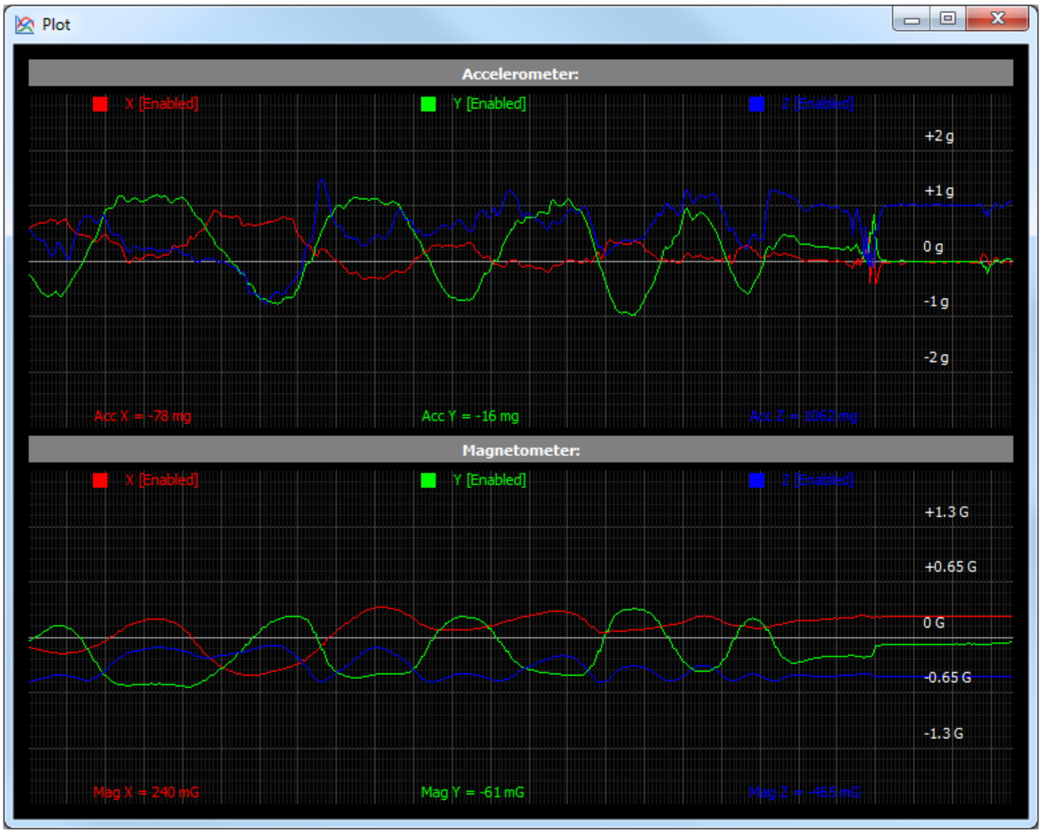
Figure 6. [Bars] tool

2.6 Plot tool

The [**Plot**] tool shows the evolution of the output over time. Figure 7. [**Plot**] tool shows the sequence of the accelerometer and magnetometer samples which have been measured by the 6-axis module mounted on the demonstration kit.

If the selected device contains just the accelerometer, the magnetic part is hidden. In the case of gyroscopes, the plot shows the angular rates.

Figure 7. [Plot] tool



2.7 Scatter plot tool

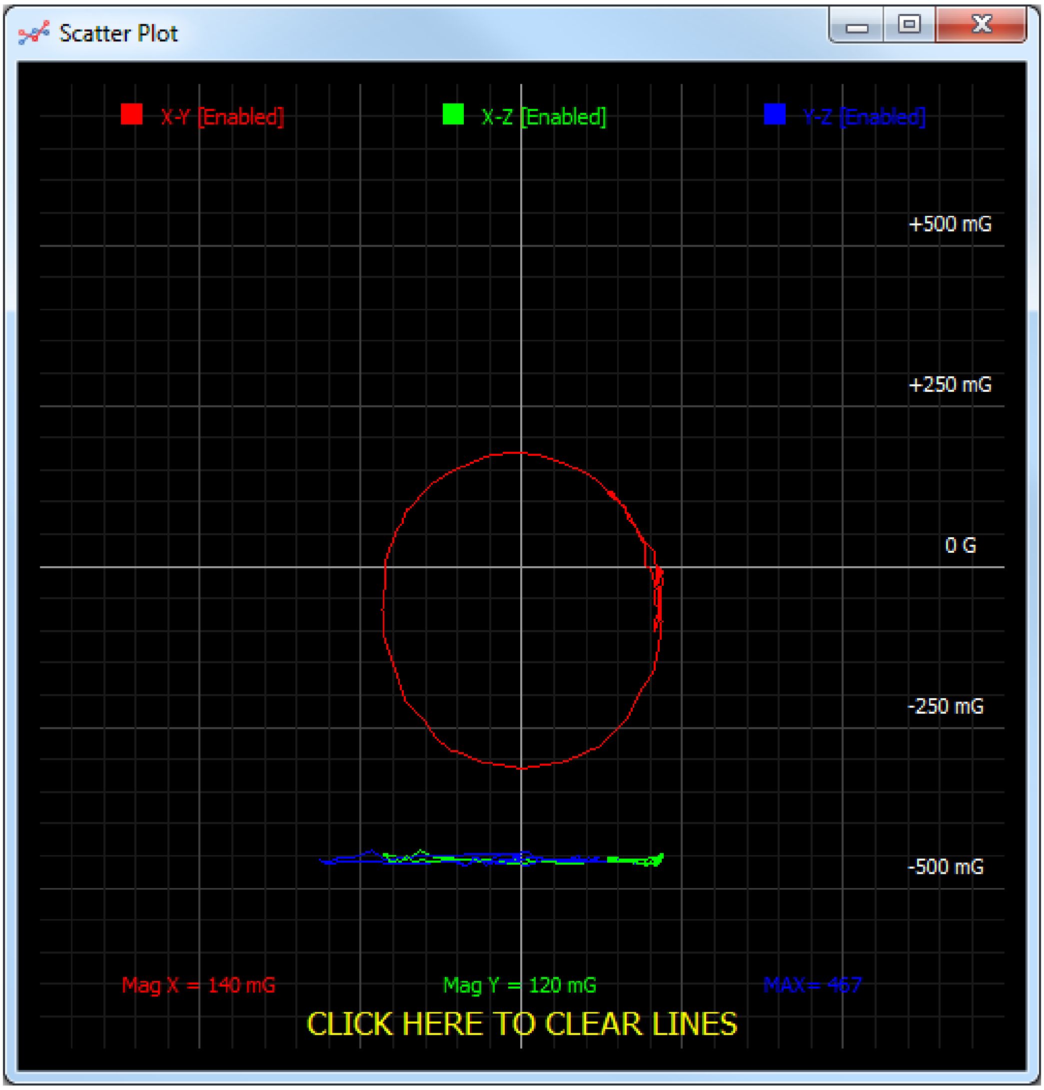
The [**Scatter Plot**] tool shows the graphical representation of the magnetometer data, used to evaluate the quality of the magnetometer calibration.

The plot shows three lines:

- the red line represents the magnetometer X data on the X-axis, and the magnetometer Y-data on the Y-axis.
- the green line represents the magnetometer X data on the X-axis, and the magnetometer Z-data on the Y-axis.
- the blue line represents the magnetometer Y data on the X-axis, and the magnetometer Z-data on the Y-axis.

The three lines can be enabled and disabled independently by clicking on the corresponding text at the top of the window. Finally, clicking on the text [**CLICK HERE TO CLEAR LINES**] resets all the data in the plot.

Figure 8. [Scatter Plot] tool

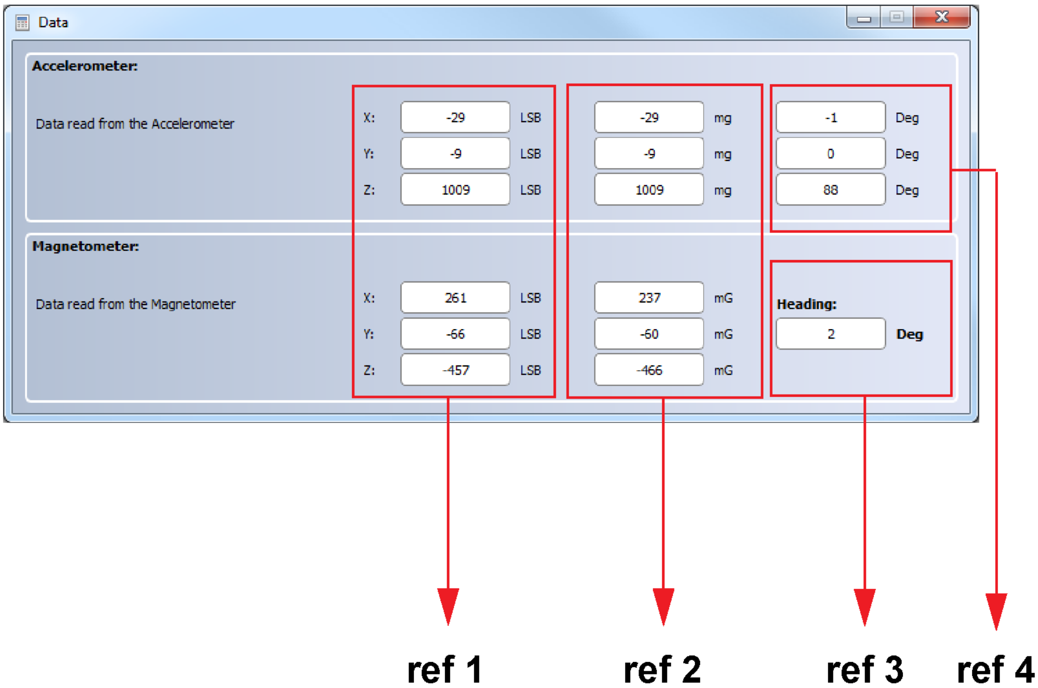


2.8 Data tool

The [**Data**] tool (Figure 9. [**Data**] tool) shows the output values measured by the sensor connected to the demonstration board. For a 6-axis module, the data is divided in the following sections:

1. [**LSB Data**] (Figure 9. [**Data**] tool, ref 1) - displays acceleration and magnetic data provided by the sensor in LSB (no sensitivity is applied).
2. [**Physical Data**] (Figure 9. [**Data**] tool, ref 2) - represents the acceleration/magnetic data measured by the sensor, expressed in the related unit of measurements (taking account of the sensitivity).
3. [**Azimuth**] (Figure 9. [**Data**] tool, ref 3) - displays the azimuth calculated using the magnetic field data.
4. [**Angle**] (Figure 9. [**Data**] tool, ref 4) - returns the tilt angle, 7 degrees.

Figure 9. **[Data] tool**



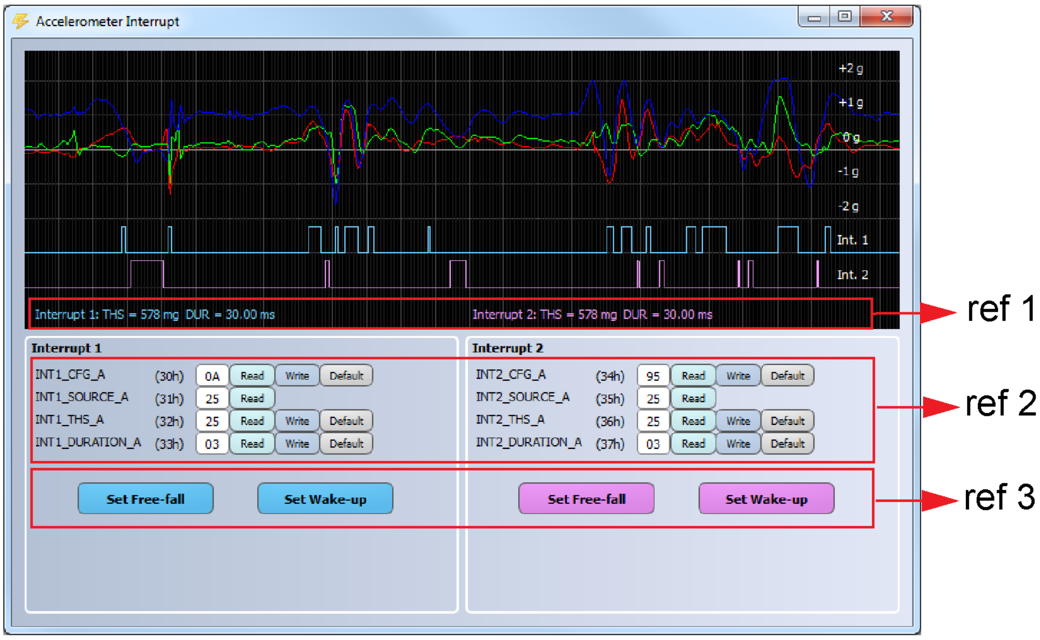
2.9 Interrupt tool

The [**Interrupt**] tool (Figure 10. [**Interrupt**] tool) allows evaluating the interrupt generation features of the MEMS sensor. In this window it is possible to configure the characteristics of the inertial events that must be recognized by the device, and to visualize (in real time) the evolution of the two interrupt lines.

The GUI provides the access to the interrupt registers (such as: INT_CFG, INT_SRC, THS and duration) which allow the configuration of the two independent interrupt sources of the device (Figure 10. [**Interrupt**] tool, ref 2). The labels located at the bottom of the graph (Figure 10. [**Interrupt**] tool, ref 1) show threshold and duration values.

Finally, two buttons per interrupt line are available in the tool, in order to easily set the recommended configuration for free-fall and wake-up detection (Figure 10. [**Interrupt**] tool, ref 3).

Figure 10. [Interrupt] tool



2.10 Compass tool

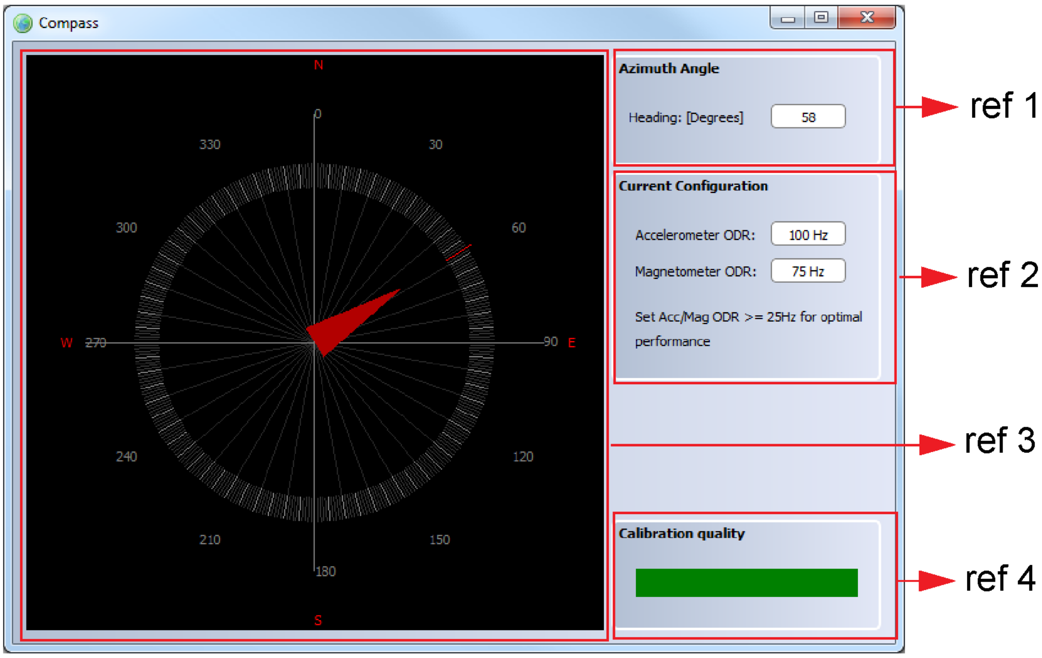
The [**Compass**] tool shows an example of compass application (Figure 11. [**Compass**] tool, ref 3), which can be implemented using a 6-axis module (3-axis accelerometer and 3-axis magnetometer).

The algorithm uses the magnetometer data to measure the Earth's magnetic field, and the accelerometer data to compensate the board inclination. Rotating the board, the GUI shows the heading of the compass (Figure 11. [**Compass**] tool, ref 1).

The performance of the compass is related to the configuration used. So, the GUI shows the current configuration and the recommended configuration (accelerometer and magnetometer ODR, Figure 11. [**Compass**] tool, ref 2).

Before using the compass demo, the system must be calibrated by moving the board randomly for a few seconds; the quality of the calibration step is indicated by a colored bar (Figure 11. [**Compass**] tool, ref 4). A green colored bar means that the quality of the calibration is optimal.

Figure 11. [Compass] tool



2.11 FIFO tool

The [**FIFO**] tool can be used to test the first-in first-out data buffer embedded in the device, when this feature is supported by the sensor (see the device datasheet for more details).

By using the buttons available in the window (Figure 12. [**FIFO**] tool, ref 1), the FIFO can be configured in all the supported modes (for example, bypass, FIFO, stream, stream-to-FIFO).

The GUI shows the values of the X, Y, Z data stored in the 32-byte deep FIFO buffer, indicating both numerical data (Figure 12. [**FIFO**] tool, ref 4) and the corresponding graph (Figure 12. [**FIFO**] tool, ref 2).

Finally, it allows users to save the data in a text file, which can be used for post-processing (Figure 12. [**FIFO**] tool, ref 3).

Figure 12. [FIFO] tool

2.12 State machine tool

The [**Finite State Machine**] tool allows the user to configure the state machines and test their functionality.

In the top part of the [**Finite State Machine**] tool main window, the user can select which state machine is selected (the selection is applied in both the [**Configuration**] tab and the [**Debug**] tab). It is also possible to configure the FSM ODR (data rate) and the long counter parameters. Finally, a converter from float32 to float16 format and viceversa is available. The converter is used to generate the values to be set in the threshold resources in the [**Variable Data Section**].

Three different tabs are available for this tool:

- [**Configuration**] (Section 2.12.1 Configuration tab) - allows setting a configuration for the state machines;
- [**Interrupt**] (Section 2.12.2 Interrupt tab) - shows a plot with accelerometer and gyroscope XYZ data in [*g*] and [dps], interrupt lines and state machine output register information;
- [**Debug**] (Section 2.12.3 Debug tab) - injects log file data in the device in order to check the functionality of the configured programs

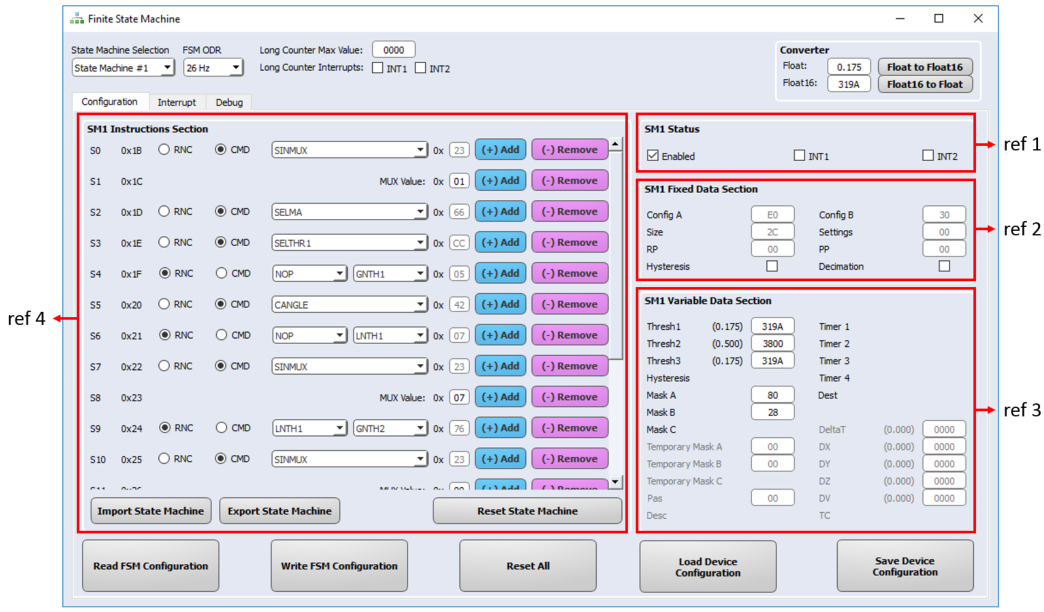
2.12.1 Configuration tab

The [**Configuration**] tab allows visualizing the current configuration and setting a new configuration for the state machines. The user has to implement the program logic in the Instruction section and to set the values of the used resources in the [**Variable Data Section**]. All the needed resources in the [**SMx Variable Data Section**] are automatically shown or hidden depending on the instructions that compose the [**SMx Instruction Section**]: the user has just to set the values of the shown resources.

The tool abstracts the finite state machine structure as shown in Figure 13. [**Configuration**] tab:

1. [**SMx Status**]
2. [**SMx Fixed Data Section**]
3. [**SMx Variable Data Section**]
4. [**SMx Instructions Section**]

Figure 13. [**Configuration**] tab



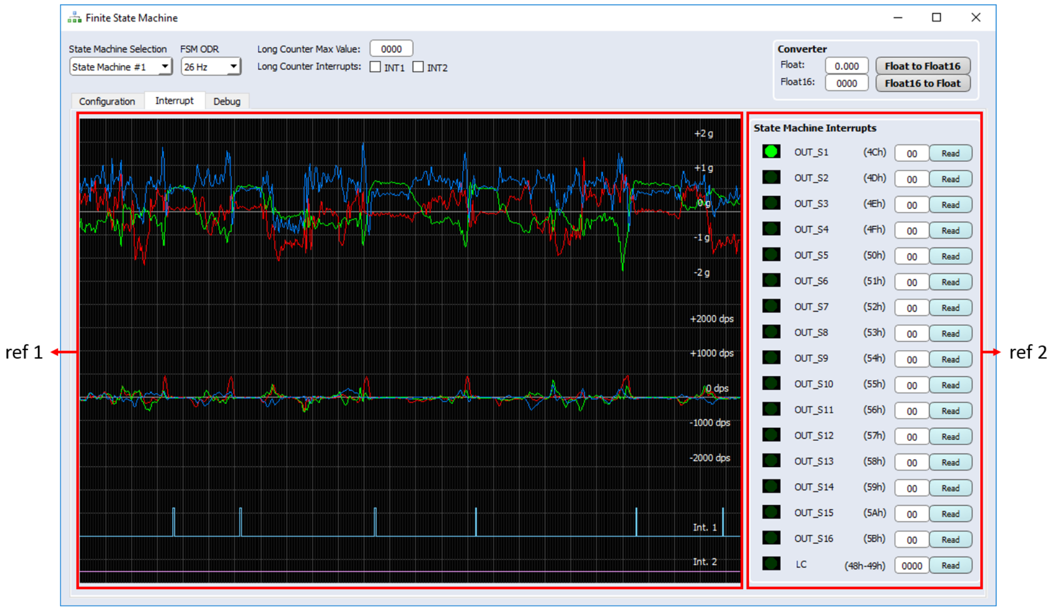
2.12.2 Interrupt tab

The [**Interrupt**] tab shows the accelerometer and gyroscope data in [*g*] and [dps] format, the evolution of the interrupt lines and the state machine interrupts information. It helps the user to check the program functionalities.

The [**Interrupt**] tab is divided in two parts as shown in Figure 14. Interrupt tab:

1. Signal plots
2. [**State Machine Interrupts**] status

Figure 14. Interrupt tab



In the [**State Machine Interrupts**] group box, a graphic green LED is turned on for ~300 msec each time the corresponding state machine interrupt source bit is set to 1. It is also possible to read the corresponding state machine OUT_S_x register by clicking on the corresponding [**Read**] button.

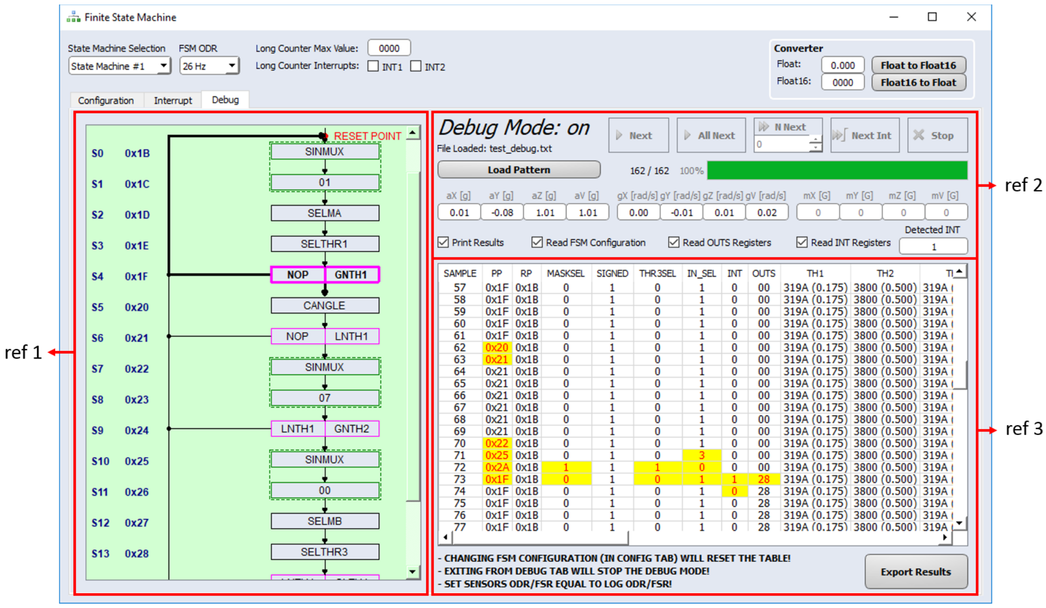
2.12.3 Debug tab

The [**Debug**] tab shows a graphic configuration of the current program set in the selected state machine and allows debugging, sample by sample, after loading an input data pattern. It is used to check the state machine functionality in order to help the user to validate the program. It is composed of three parts:

1. State machine flow
2. Debug commands
3. Output results

When the debug mode is enabled, the current state is highlighted and it is automatically updated based on the injected sample and program behavior. The tool provides the possibility to inject one or more samples at a time: each time a sample is injected, a new row containing the updated values of the state machine resources is added in the results of the output table.

Figure 15. [Debug] tab



2.13 Machine Learning Core tool

The [**Machine Learning Core (MLC)**] tool allows the user to configure a machine learning core which is embedded in some devices. Two different tabs are available for this tool:

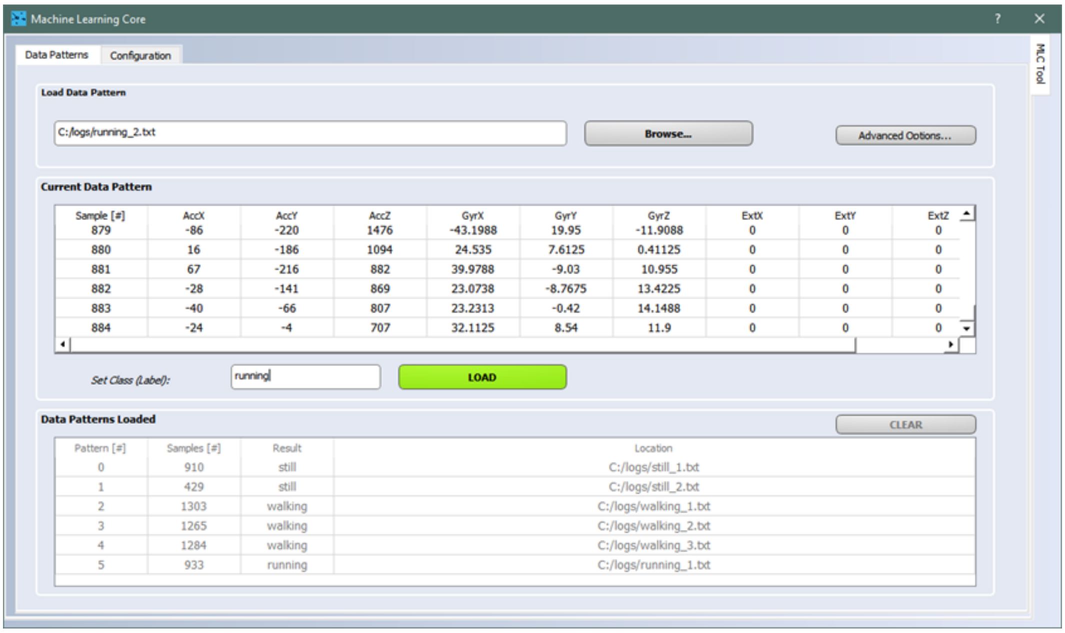
- [**Data Patterns**] (Figure 16. MLC [**Data Patterns**] tab) – allows managing the data patterns to be used and assigning a label to each data pattern loaded.
- [**Configuration**] (Figure 17. MLC [**Configuration**] tab) – allows setting a configuration for the machine learning core.

2.13.1 Data Patterns tab

The [**Data Patterns**] tab allows managing the data patterns to be used for the machine learning processing. The data patterns which are possible to load must have the same data format as the log files generated by Unico in the [**Load/Save**] tab. The unit of measurement for the data are 'm*g*' for the accelerometer and 'dps' for the gyroscope.

When a new data pattern is loaded, an expected result must be assigned, which is the label for the machine learning processing.

Figure 16. MLC [Data Patterns] tab

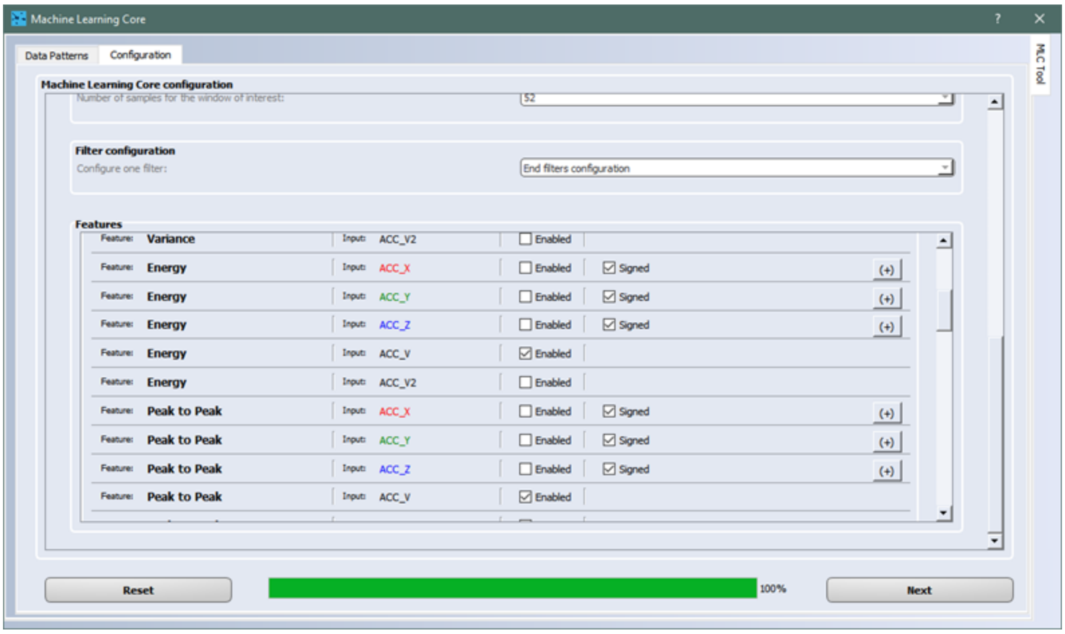


2.13.2 Configuration tab

The [**Configuration**] tab allows configuring the machine learning core by asking the user a series of inputs including the device settings (ODR, full scale) and the machine learning core settings (window for features computation, filters and features, decision trees, results, and so forth).

When the configuration procedure has been completed, the tool generates a configuration file (.ucf) containing the device configuration to enable the machine learning core running the configured algorithm.

Figure 17. **MLC [Configuration] tab**



2.14 Click tool

The [**Click**] tool (Figure 18. [**Click**] tool) allows evaluating the click or double-click function of the MEMS sensor. In this window it is possible to configure the device for single-click and double-click detection and visualize (in real time) the evolution of the interrupt line.

This tool is available only for the devices which integrates the click/double-click function.

By clicking the buttons [**Set Single Click**] (Figure 18. [**Click**] tool, ref 1) or [**Set Double Click**] (Figure 18. [**Click**] tool, ref 2), a default configuration for single-click or double-click detection is loaded. After loading the configuration, a green light appears when the single or double click has been recognized.

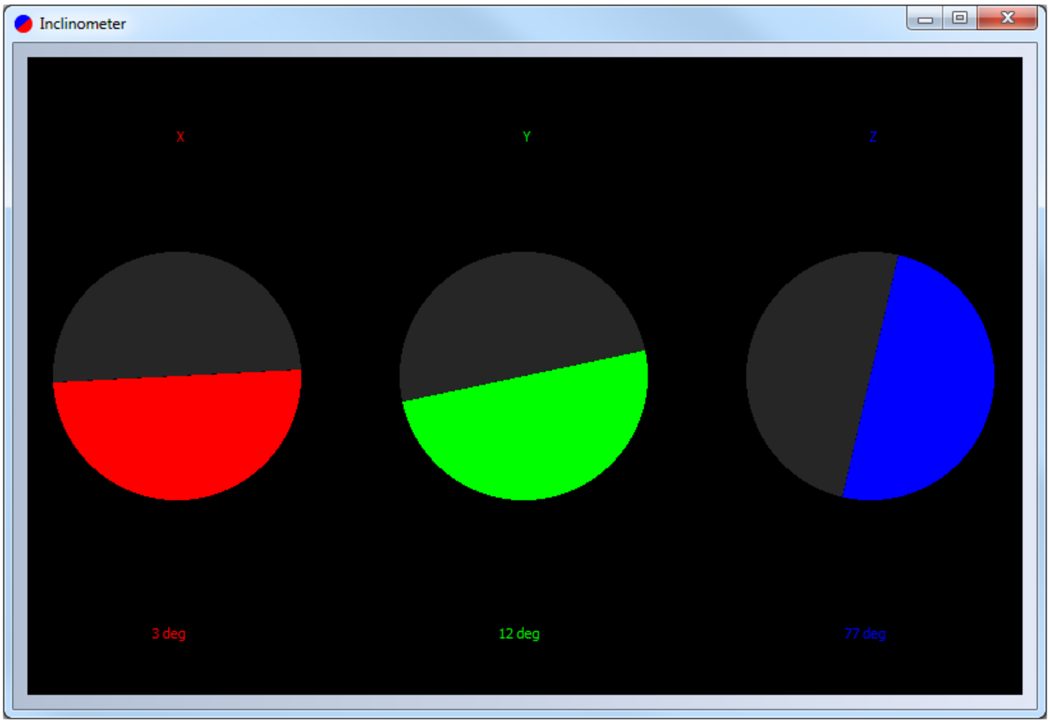
It is also possible to change the register configuration by setting different values in the interrupt registers shown in the tool (Figure 18. [**Click**] tool, ref 3).

Figure 18. [Click] tool

2.15 Inclinometer tool

The [**Inclinometer**] tool (Figure 19. [**Inclinometer**] tool) represents the angle between the accelerometer axis and the horizontal plane. This tool is available if the sensor in use integrates an accelerometer, otherwise it is hidden.

Figure 19. [Inclinometer] tool



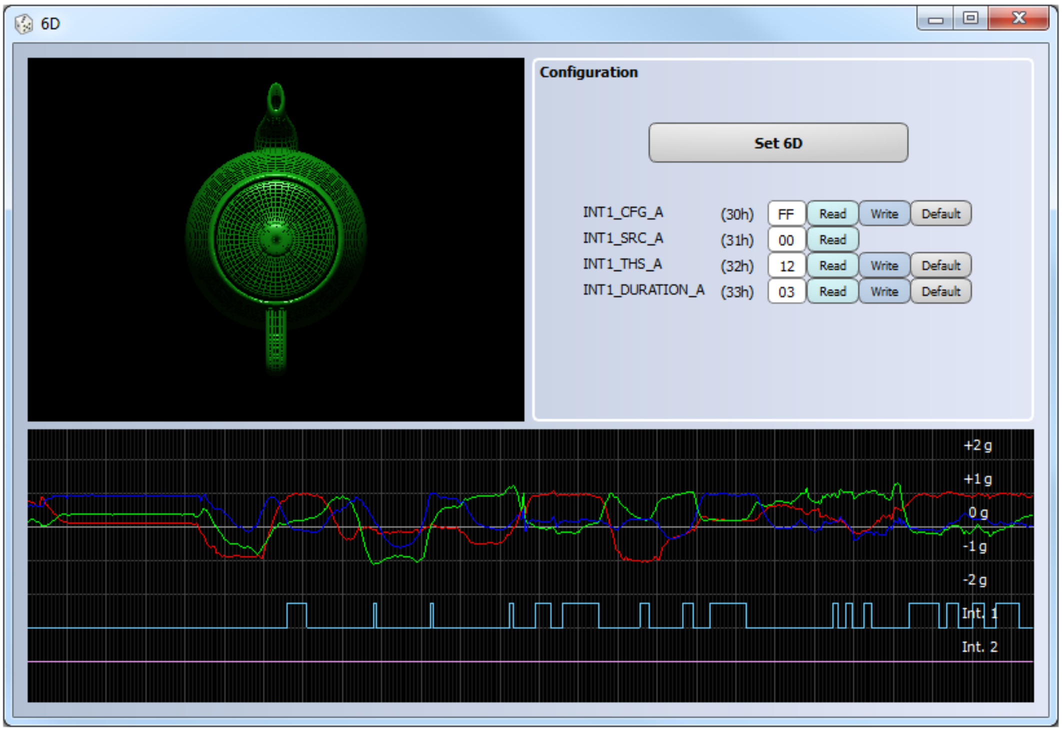
2.16 6D tool

The [**6D**] tool (Figure 20. [**6D**] tool) provides an example of how to use the [**6D position**] function.

In this tool it is possible to configure the interrupt for 6D recognition by clicking the button [**Set 6D**] (Figure 20. [**6D**] tool, ref 1). After loading the configuration, the teapot is oriented depending on the 6D position detected (Figure 20. [**6D**] tool, ref 2). The window also shows (in real time) the evolution of the interrupt line.

It is also possible to change the register configuration by setting different values in the interrupt registers shown in the tool (Figure 20. [**6D**] tool, ref 2).

Figure 20. **[6D] tool**



2.17 FFT tool

The [**FFT**] tool (Figure 21. [**FFT**] tool) shows the fast Fourier transform of the output data.

The window shows the time-domain plot and the frequency-domain plot for each axis.

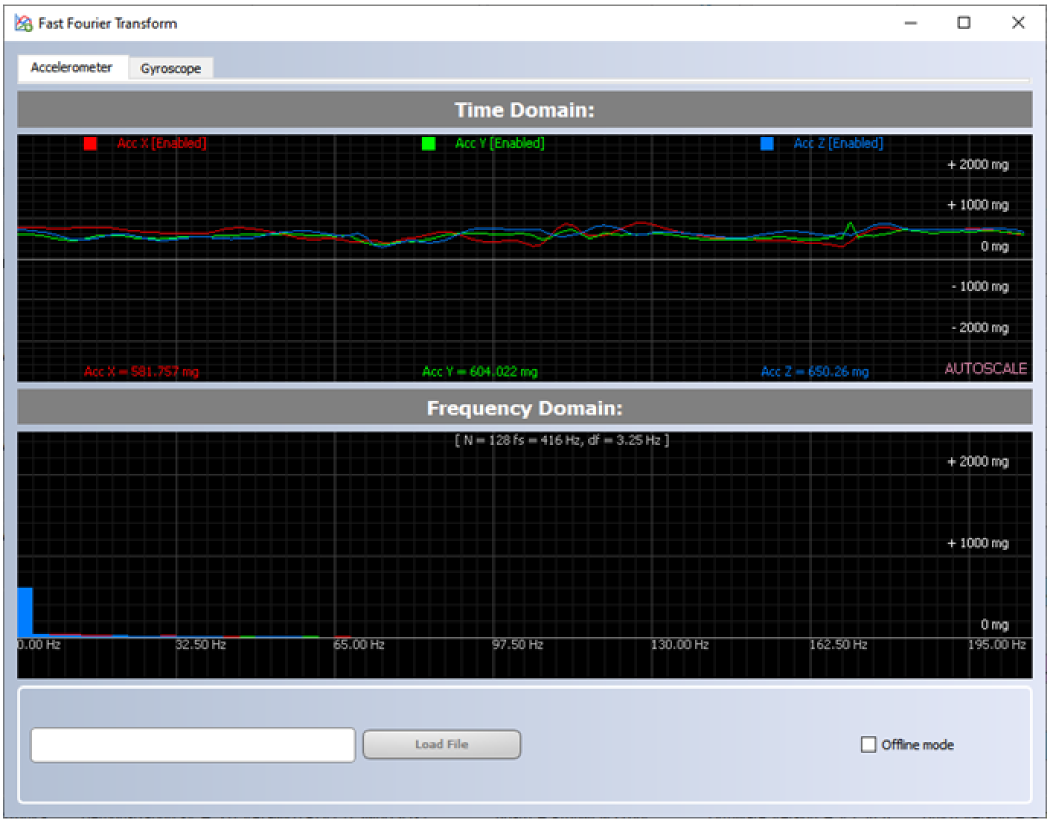
The FFT is performed on the latest 128 samples of the time-domain waveform, so the spectrum in the frequency domain is divided in 64 frequencies (from 0 to ODR/2 Hz).

The frequency domain plots show the module of the Fourier transform of each axis, expressed in *g* for the accelerometer and dps for the gyroscope.

Offline mode allows the user to load a recorded data log both with Unico data log format and with .csv data.

Note: *A header line is required for data labeling. Once the data log is selected and the FFT computation starts, the Fast Fourier Transform is evaluated using a single window and applying a zero padding for evaluation on a number of samples to the power of two.*

Figure 21. [FFT] tool



2.18 Pedometer tool

The [**Pedometer**] tool can be used to configure and test the pedometer embedded in the device when this feature is supported by the sensor (see the device datasheet for more details).

Three different tabs are available for this tool:

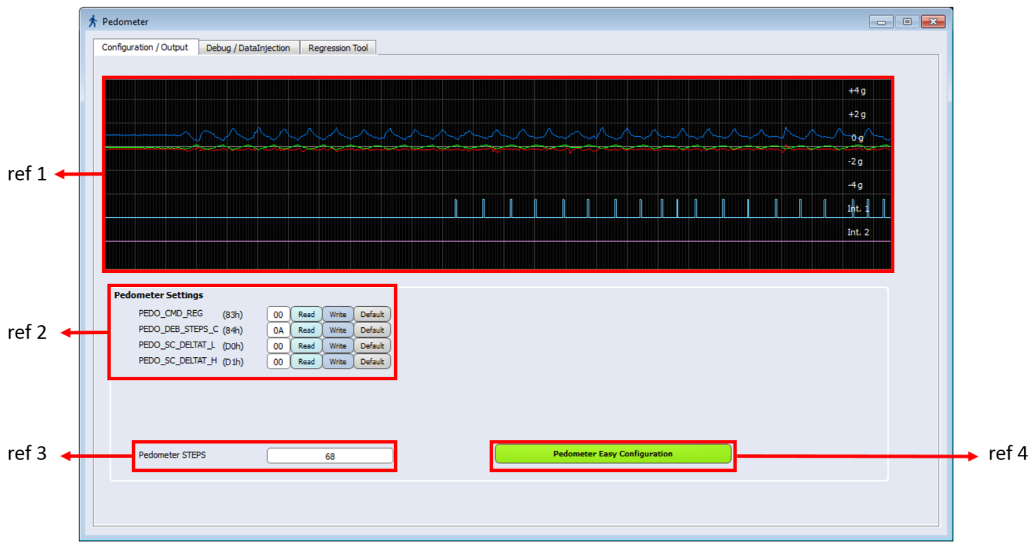
- **[Configuration / Output] tab (see Section 2.18.1 Configuration / Output tab)**: it allows fast-configuring the pedometer with its default configuration.
- **[Debug / Data Injection] tab (see Section 2.18.2 Debug / Data Injection tab)**: it is used to load a data pattern into the device in order to test (offline) the pedometer on the pattern itself.
- **[Regression Tool] tab (see Section 2.18.3 Regression Tool tab)**: it allows finding an optimal configuration on the base of a predefined dataset.

2.18.1 Configuration / Output tab

The [**Configuration / Output**] tab (Figure 22. [**Configuration / Output**] tab) allows fast-configuring the pedometer. In this window it is possible to visualize in real-time the evolution of both the accelerometer signal and the two interrupt lines (Figure 22. [**Configuration / Output**] tab, ref 1) and to read the output pedometer step count (Figure 22. [**Configuration / Output**] tab, ref 3).

One button is available in the tool in order to easily enable and configure the pedometer with its default configuration (Figure 22. [**Configuration / Output**] tab, ref 4). The GUI also allows directly accessing the pedometer registers (such as PEDO_CMD_REG, PEDO_DEB_STEPS_CONF, PEDO_SC_DELTAT_L, PEDO_SC_DELTAT_H) in order to set a user-defined pedometer configuration (Figure 22. [**Configuration / Output**] tab, ref 2).

Figure 22. [Configuration / Output] tab



2.18.2 Debug / Data Injection tab

The [**Debug / Data Injection**] tab (Figure 23. [**Debug / Data Injection**] tab) is used to load a data pattern into the device and run the pedometer logic on the pattern itself (offline post-processing).

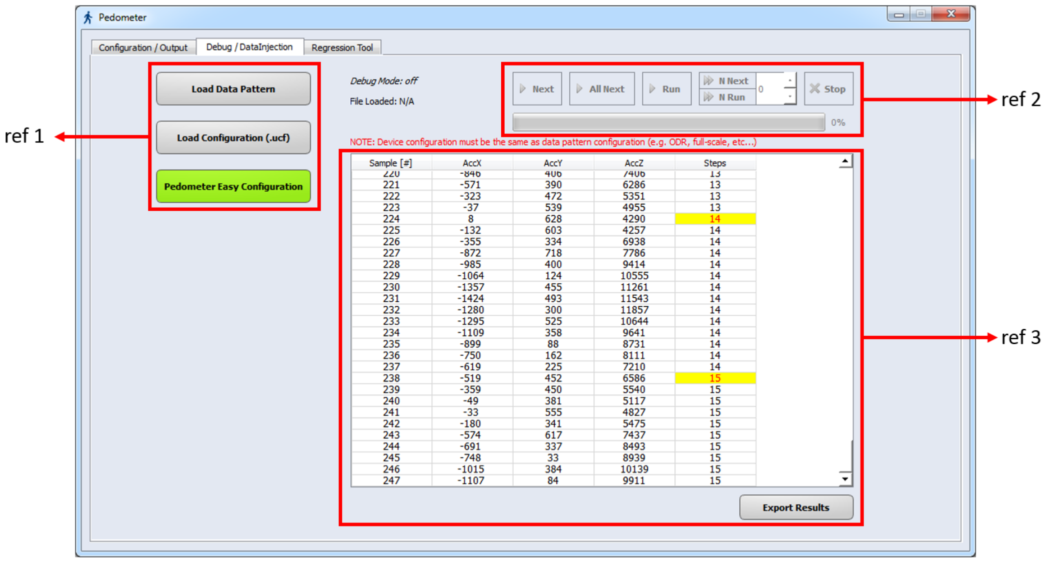
On the left side of the GUI a three-button toolbar is available for loading a data pattern in the GUI. Pedometer configuration can be loaded from the .ucf file; it is also possible to fast-configure the pedometer in its default configuration by clicking on the [**Pedometer Easy Configuration**] button (Figure 23. [**Debug / Data Injection**] tab, ref 1). The device enters the debug mode right after the data pattern has been loaded.

On the top of the GUI, a toolbar is available to control the data injection (Figure 23. [**Debug / Data Injection**] tab, ref 2), which can be applied with the maximum level of flexibility:

- [**Next**] button is used to load one sample in the pedometer.
- [**All Next**] button is used to load sample-by-sample the full data pattern in the pedometer
- [**Run**] button is used to load the entire pattern in the pedometer, displaying the final result only.
- [**N Next**] button allows loading the indicated number of samples in the pedometer, displaying all the samples.
- [**N Run**] button allows loading the indicated number of samples, displaying the final result only
- [**Stop**] button allows stopping the current debug session. If the button [**All Next**] or [**Run**] was pressed, the [**Stop**] button changes into a [**Pause**] button, which allows pausing the current debug session with the possibility to resume it. When the debug session is paused, the [**Stop**] button changes again into [**Stop**] button, allowing the possibility to stop the current debug session. After that, another debug session can be launched (the data pattern must be reloaded).

During a debug session, the current status (sample loaded and output number of steps) is displayed in the central part of the GUI (Figure 23. [**Debug / Data Injection**] tab, ref 3). A dedicated button is also available in order to export the results of the debug session in TSV format.

Figure 23. [Debug / Data Injection] tab



2.18.3 Regression Tool tab

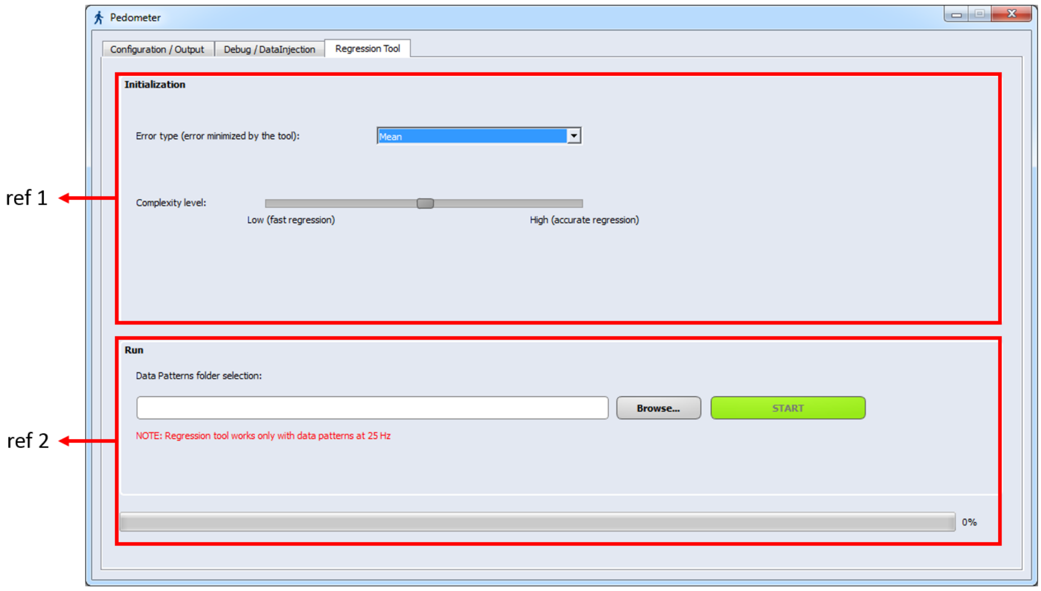
The [**Regression Tool**] tab (Figure 24. [**Regression Tool**] tab) allows finding an optimal configuration on the basis of a predefined dataset (in this context, a dataset is a collection of data patterns with a reference number of steps for each pattern).

At the top of the GUI, the initialization view (Figure 24. [**Regression Tool**] tab, ref 1) contains the two initialization parameters:

- **[Error type]**, the error to be minimized by the tool (that is, mean, mean plus standard deviation, mean plus two times the standard deviation);
- **[Complexity level]**, the level of depth (in terms of iterations) of the regression. Execution time of a low-complexity level regression is lower than a high-complexity level, but the parameter space is less deeply explored.

At the bottom of the GUI, the run view (Figure 24. [**Regression Tool**] tab, ref 2) contains the textbox for the input data path and the [**Start**] button, which runs the regression. A progress bar is available in order to notify the user about the progress of the regression.

Figure 24. [Regression Tool] tab



Note: The input data path to be specified in the textbox must be a folder containing only the pedometer data patterns to be applied to the regression session. Each file must contain accelerometer data in tab or space separated format in [mg] and collected at 26 Hz output data rate. Each filename must contain the 'stepsXXX' string, where XXX is the effective reference number of steps (that is, test001_steps100.txt, data_steps54.txt, pattern_steps1043.txt).

Once the regression has been completed, the tool generates a folder under the Unico root, named [data]_[hour]_Pedometer, containing two files:

- pedometer.ucf, containing the optimal pedometer configuration generated by the tool;
- regression_tool.config, containing some meta-information used by the [**Regression Tool**] itself and statistical data about the pedometer performance on the input dataset.

Under the Unico root, another regression_tool.config file is generated. This file is automatically used by the [**Regression Tool**] in order to start a new regression analysis from the optimal configuration found in the latest regression executed. If the user's intention is to run a new regression from scratch (starting from the default pedometer configuration), this file has to be manually deleted.

3 Data acquisition quick start

This section describes the basic steps that must be performed to acquire the data from the demonstration board:

1. Plug the demonstration board into the USB port.
2. Start the Unico GUI.
3. Select the STEVAL-MKI according to the device/demonstration board in use (Figure 1. Unico launcher).
4. Go to the [**Options**] or [**Registers**] tab and click on [**Easy Configuration**] (Figure 3. [**Options**] tab, ref 1; Figure 4. [**Registers**] setup tab, ref 3)
5. Click on the [**Start**] (or [**Stop**]) button to activate (or stop) the sensor data collection.
6. Use the buttons on the left (Section 2.1 , ref 3) to display the desired tool.
7. To close the application, click on the button [**Exit**] or simply close the main window.

4 Port detection

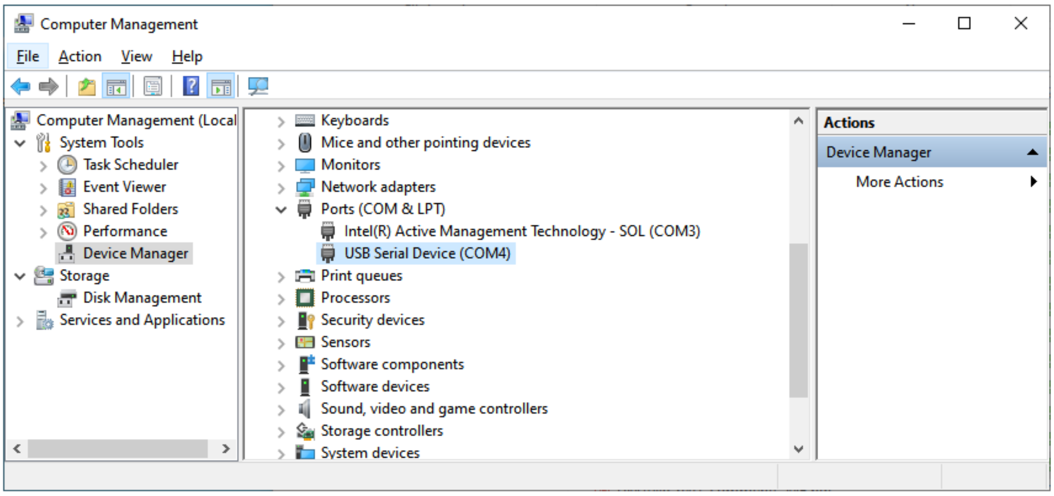
In some cases, the Unico software cannot automatically detect the port where the board is connected. In these cases, the user needs to check the correct port and select it manually in the Unico GUI.

This section describes how to detect the port for each operating system.

4.1 Port detection on Windows

Right-click on [**This PC**] (or [**My Computer**]), choose [**Manage**], and then [**Device Manager**]. You will find the COM port in the [**Ports**] section of the [**Device Manager**], with the name [**USB Serial Device (COMxx)**], as shown in Figure 25. Port on Windows.

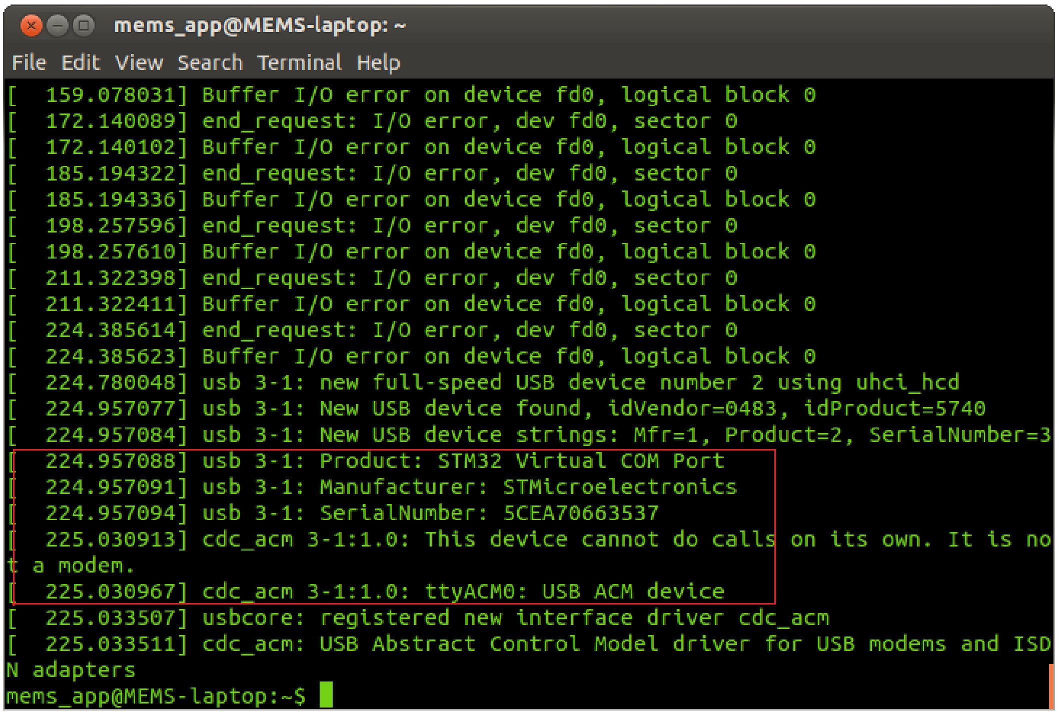
Figure 25. Port on Windows



4.2 Port detection on Linux (Ubuntu)

Just after connecting the board, open a terminal and type “dmesg”. The port has a name similar to “ttyACM0” (or “ttyS0”), see details in Figure 26. Port on Linux.

Figure 26. Port on Linux



4.3 Port detection on Mac OS

Before connecting the board to your Mac, open a terminal and type:

*ls /dev/tty.**

A list of device files appears.

Now, if you connect the board to your Mac and type the same command again, a new file appears named */dev/tty.usbmodemXXX* (the last three characters are numbers automatically assigned by the system).

This means that the board is connected to the port *usb.modemXXX*.

Revision history

Table 1. Document revision history

Date	Revision	Changes
02-Mar-2011	1	Initial release
06-Jun-2012	2	Added 'Automatic COM Port Detection' flag in Section 2: "Unico graphical user interface" Updated Table 1: Device vs supported tabs including new supported devices All figures have been updated
10-Sep-2013	3	Updated title of document Updated Figure 1: "Unico Launcher" Updated Table 1 with new supported devices and added "State machine" tab; removed obsolete demonstration board (STEVAL-MKI063V1 based on the LSM303DLH) Added Section 2.15: State machine tab Minor textual updates
03-Nov-2014	4	Entire document revised according to release 4.0.0.0 of Unico • All figures updated • Linux and Mac OS subsections added to Section 1: "PC system requirements" • Removed irrelevant subsections and table from Section 2: "Unico graphical user interface" and added Section 2.7: "Scatter plot tool"; Section 2.12: "State machine tool"; Section 2.16: "FFT tool" and Section 4: Port detection"
20-Oct-2016	5	Added STEVAL-MKI109V3 to "Introduction" Textual updates in Section 1.1: "Windows platforms" and Section 4.1: Port detection on Windows"
23-Jan-2019	6	Updated Introduction Updated Section 2 Unico graphical user interface and Figure 1. Unico launcher Updated Section 2.12 State machine tool Updated Section 2.12.1 Configuration tab Updated Section 2.12.2 Interrupt tab Updated Section 2.12.3 Debug tab Added Section 2.13 Machine Learning Core tool Added Section 2.18 Pedometer tool
24-Mar-2022	7	Updated Section 1.1 Windows platforms, Section 1.2 Linux platforms, Section 1.3 Mac OS X platforms Updated Figure 1. Unico launcher, Section 2.1 Tools available Updated Section 2.2 Options tab Updated Section 2.3 Registers setup tab Updated Section 2.4 Load/Save tab Updated Figure 16. MLC [**Data Patterns**] tab and Figure 17. MLC [**Configuration**] tab Updated Section 2.17 FFT tool Updated Section 4.1 Port detection on Windows

Contents

List of figures