User manual

Software guide for Unico Lite

Introduction

The Unico Lite graphical user interface (GUI) is a complete evaluation software which provides the source code to manage sensor data-flow from generic MEMS sensors (such as accelerometers, gyroscopes, magnetometers and pressure sensors).

This user manual describes all the Unico Lite GUI components. The GUI interacts with the STEVAL-MKI109V3 (Professional MEMS tool board), which is the motherboard compatible with all ST MEMS adapter boards; for details on the commands or data format, refer to UM2116 on www.st.com.

1 Unico Lite graphical user interface

The Unico Lite software has been designed to operate with Microsoft® Windows platforms and is written with Microsoft® Visual Studio 2017.

The Unico Lite graphical user interface is a simple Windows form application written in C# (.NET Framework 3.5) designed to show basic operation such as board connection, read/write register sensors and how to acquire continuous data from the STEVAL-MKI109V3 Professional MEMS tool board (data-flow).

The basic concepts described below are suitable for different sensors; the source code can be used directly with four different evaluation boards (but you can easily add new evaluation boards for other gyroscopes, accelerometers and modules):

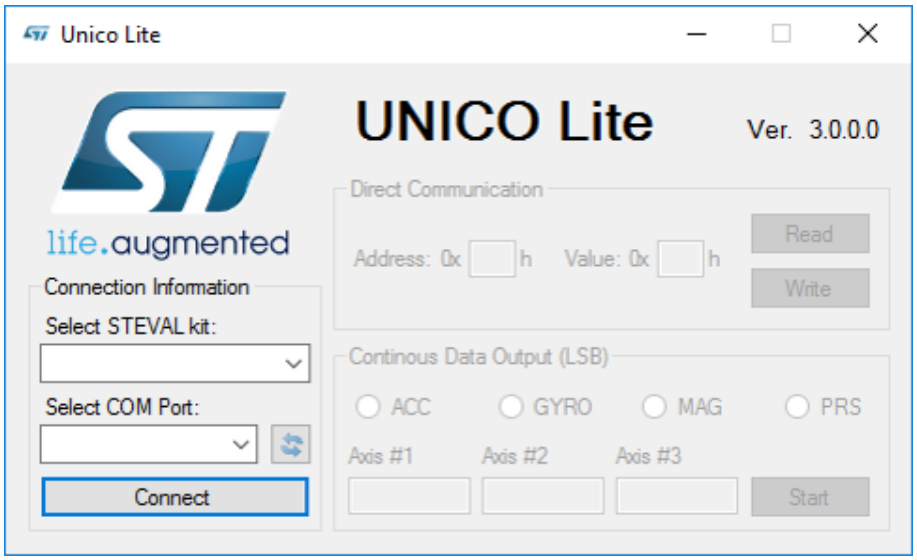
- STEVAL -MKI178V1 for LSM6DSL 6-axis iNEMO inertial module
- STEVAL- MKI179V1 for LIS2DW12 3-axis digital accelerometer
- STEVAL- MKI181V1 for LIS2MDL 3-axis digital magnetometer
- STEVAL- MET001V1 for LPS22HB digital pressure sensor

To execute the Unico Lite software GUI:

Step 1. Plug the STEVAL-MKI109V3 evaluation board to the PC through the USB port.

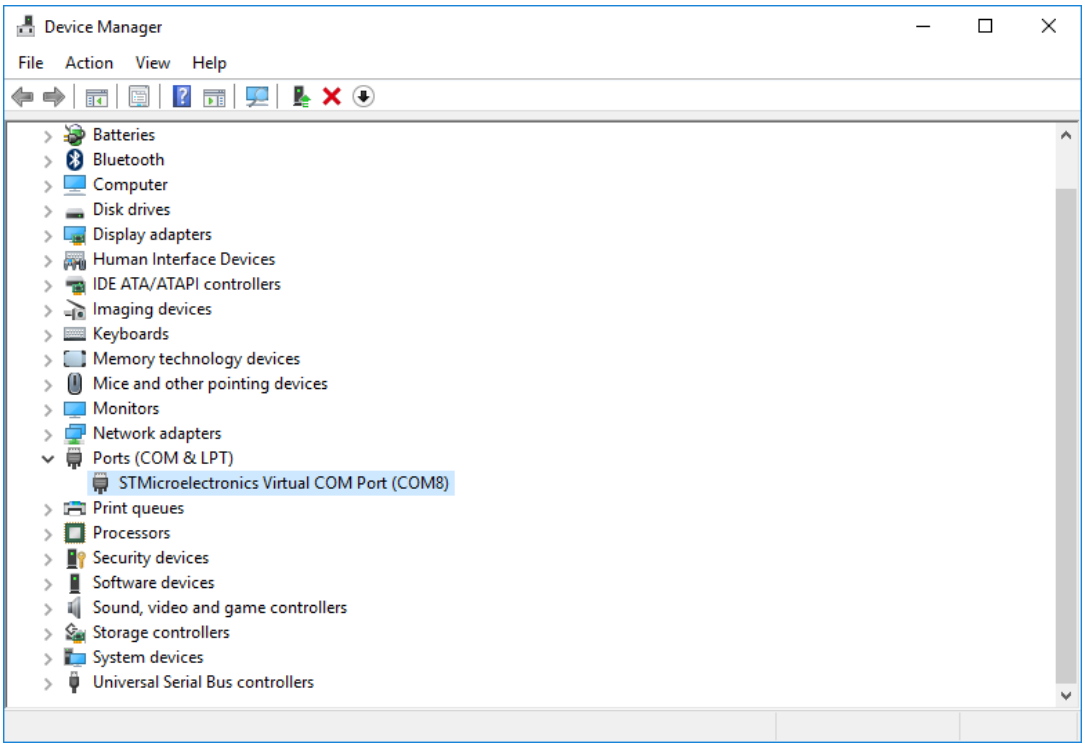
Step 2. Click on [**Unico_Lite**].
The GUI window appears:

Figure 1. Unico Lite GUI window



Step 3. Unico Lite automatically searchs for available STMicroelectronics Virtual COM Port devices and sets the [**Select COM Port**] ComboBox accordingly.
If the COM Port found is not correct, open [**Device Manager**], scroll through the list until you find [**Ports (COM & LPT)**] and look for STMicroelectronics Virtual COM Port.
In the example shown in the figure below, COM8 has been assigned to the evaluation board.

Figure 2. **Unico Lite GUI Virtual COM Port assignment**



Step 4. Select, from [**Select STEVAL kit**] ComboBox, the adapter board currently in use.

Step 5. Click [**Connect**].

Now you can use the GUI to:

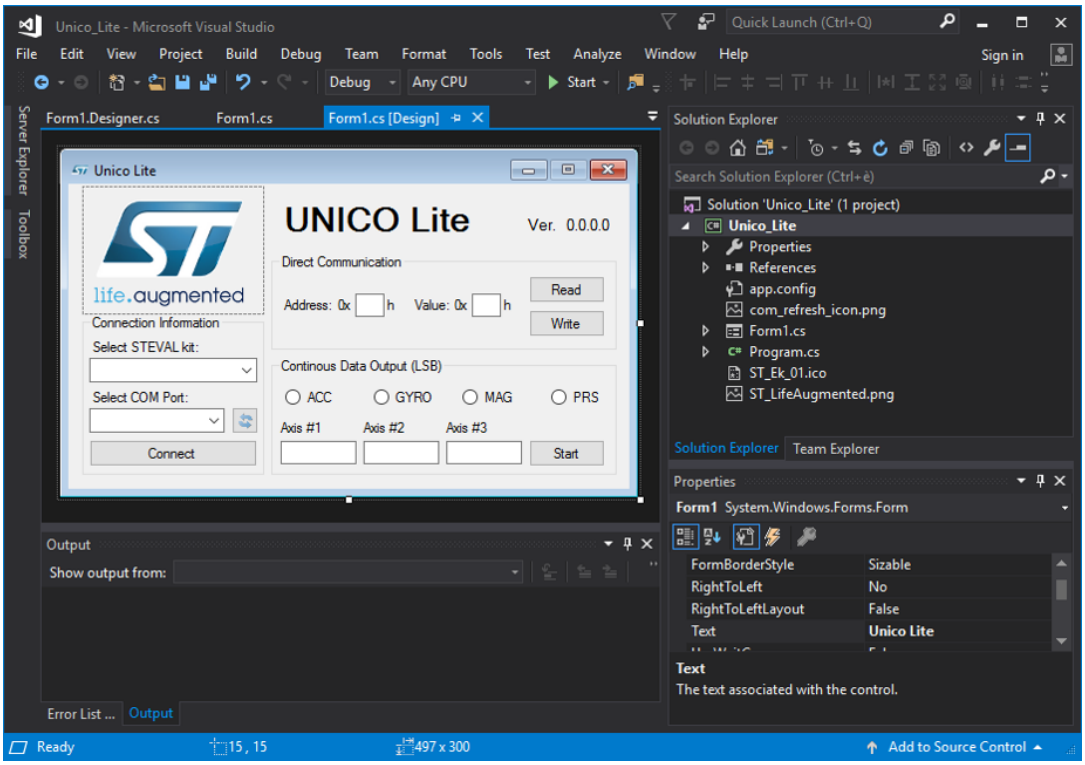
- choose one of the sensors (accelerometer, gyroscope, magnetometer or pressure sensor) by using the four radio buttons
- read a register: insert the register address in the address box (hexadecimal value) and click the [**Read**] button. The register content is shown in the value box.
- write a register: insert the register address in the address box (hexadecimal value), insert the register value in the value box (hexadecimal value), and click the [**Write**] button.
- get data continuously: click the [**Start**] button and check sensor data.

Note: For the continuous data, check the register settings to have the sensor in normal or low power mode. In case of power-down configuration, the boxes show no data as there is no data coming from the sensor.

2 Windows form design

Microsoft® Visual Studio 2017 has been used to design the form application.

Figure 3. Microsoft® Visual Studio 2017 main screen



Unico Lite GUI (see the figure below) is composed of the controls described in Table 1. Unico Lite main controls.

Figure 4. Unico Lite controls

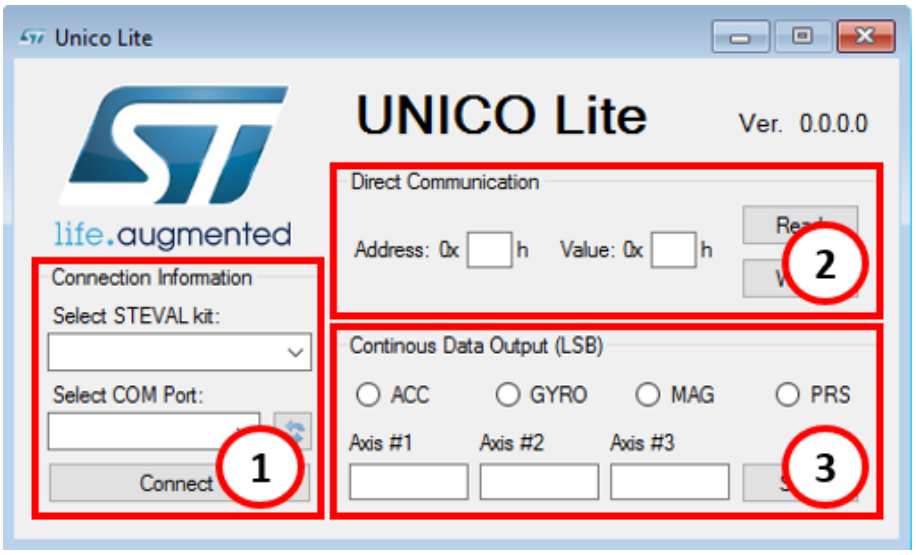


Table 1. Unico Lite main controls

Ref.	Name	Control type
1 (Connection management)	CBX_Kit	ComboBox
	CBX_ComPorts	ComboBox
	BTN_ComRefresh	Button
	BTN_Connect	Button
2 (Direct read/write register)	TB_Address	TextBox
	TB_Value	TextBox
	BTN_Read	Button
	BTN_Write	Button
3 (Continuous data reading)	RB_Acc	RadioButton
	RB_Gyro	RadioButton
	RB_Mag	RadioButton
	RB_Prs	RadioButton
	TB_Val1	TextBox
	TB_Val2	TextBox
	TB_Val3	TextBox
	BTN_Start	Button

3 Code

3.1 Constants

- `const int MAX_KITS`: defines the maximum number of kits supported by the code. In this version the software supports 4 kits.
-
```
public struct Kits
{
public string Name;
public int words;
public string setdb;
}
```

  defines a single kit that can be described by:

  1. Name: STEVAL kit code
  2. words: number of data bytes coming back after `*start` command (refer to UM2116, table 3, on www.st.com)
  3. setdb: selects the part of the firmware able to handle the adapter board (e.g. for STEVAL-MKI178V1 the command is `*setdb178v1`)

3.2 Variables

- `Kits[] MKI = new Kits[MAX_KITS]`: creates a struct array for kit description
- `int Kit_Index`: global value for the ComboBox kit index (default value = -1)
- `string DataReadFromSerialPort`: temporary string for data reading
- `bool Start`: flag used to monitor whether the [**Start**] button has been pressed or not
- `volatile bool Connected`: flag used to monitor the connection status defined volatile because it is used in different threads
- `public string readAcc`: prefix of the read command for the accelerometer sensor
- `public string writeAcc`: prefix of the write command for the accelerometer sensor
- `public string readGyr`: prefix of the read command for the gyroscope sensor
- `public string writeGyr`: prefix of the write command for the gyroscope sensor
- `public string readMag`: prefix of the read command for the magnetometer sensor
- `public string writeMag`: prefix of the write command for the magnetometer sensor
- `public string readPrs`: prefix of the read command for the pressure sensor
- `public string writePrs`: prefix of the write command for the pressure sensor

3.3 Classes

- `public SerialPort c_Serial`: used to control a serial port file resource. This class provides synchronous and event-driven I/O, access to pin and break states, and to serial driver properties.

3.4 Initialization

The code that initializes controls and classes is the following:

```
public Form1()
{
   InitializeComponent();
```

```
    //  Get file version
    LB_VersionValue.Text = GetFileVersion();

    //  Fill the evaluation board array
    //  STEVAL-MKI178V1 for LSM6DSL Combo (accelerometer and gyroscope) sensor
    MKI[0].Name = "MKI178V1 (LSM6DSL)";
    MKI[0].words = 22;
    MKI[0].setdb = "*setdb178v1";

    //  STEVAL-MKI179V1 for LIS2DW12 accelerometer sensor
    MKI[1].Name = "MKI179V1 (LIS2DW12)";
    MKI[1].words = 13;
    MKI[1].setdb = "*setdb179v1";


    //  STEVAL-MKI181V1 for LIS2MDL magnetometer sensor
    MKI[2].Name = "MKI181V1 (LIS2MDL)";
    MKI[2].words = 12;
    MKI[2].setdb = "*setdb181v1";

    //  STEVAL-MET001V1 for LPS22HB pressure sensor
    MKI[3].Name = "MET001V1 (LPS22HB)";
    MKI[3].words = 14;
    MKI[3].setdb = "*setdb001V1";



    //  Fill the Combo Box with Kits
    for (int i = 0; i < MAX_KITS; i++)
    {
       CBX_Kit.Items.Add(MKI[i].Name);
    }

    //  Initialize Serial Port Objects
    c_Serial = new SerialPort();
    DetectCOMPorts();
}
```

3.5 Connection

This part of the code is in charge of managing serial port operations (like connect, disconnect, open and close) and button iterations.

3.5.1 DetectCOMPorts

It is used to fill automatically the CBX_ComPorts ComboBox depending on the “STMicroeletronics Virtual COM Port” found in the system. It is called at the GUI initialization.

3.5.2 BTN_ComRefresh_Click

It is used to call the “DetectCOMPorts” function to update the CBX_ComPorts ComboBox depending on the “STMicroeletronics Virtual COM Port” found in the system. It can be used when the board is connected after the GUI has been opened.

3.5.3 BTN_Connect_Click

It manages the connect/disconnect status based on *Connected* flag.

If Connected flag is false

The GUI has to open the communication. It configures the serial port object and open the connection to the evaluation board. In detail, the port settings are:

1. PortName: the one selected with CBX_ComPort
2. BaudRate: 115200
3. DataBits: 8
4. ReadTimeout: 500 [ms]
5. WriteTimeout: 500 [ms]
6. DataReceived: connection between Event and Method. Each time new data are sent to the serial port from the evaluation board, the main thread calls the Method (in this case *USB_DataReceived()*).

After configuration, the software tries to open the selected Virtual COM port. If no exception occurs, the software writes on the serial port:

1. "`*setdb`*xxx*`v1`" (*xxx* is the STEVAL code: for instance, 178 for LSM6DSL or 181 for LIS2MDL)
2. `*zoff`

Note: These two commands are mandatory for the connection.

Moreover, radio buttons (and possible kit dependent controls) are initialized just after the connection.

At the end of the function, the software performs Enable/Disable controls.

If Connected flag is true

The GUI has to close the communication. It performs the following operations:

1. sends `*stop` command
2. sends `*zon` command
3. sends `*dbreset` command
4. closes the serial port
5. enables/disables controls
6. resets the TextBox contents to clear the GUI

3.5.4 BTN_Start_Click

It manages the start/stop status based on *Start* flag.

If Start flag is false

It enables/disables controls and sends the `*start` command. The *Start* flag is set to true and the BTN_Start text is set to "Stop".

If Start flag is true

It enables/disables controls and sends the `*stop` command. The *Start* flag is set to false and the BTN_Start text is set to "Start".

3.5.5 CloseSerialPort

It closes the serial port connection.

3.6 Decode

The main purpose of this function is to get sensor data from the data stream and visualize them.

The data stream is a sequence of characters such as:

```
... s t xh xl yh yi zh zl i1 i2 s \r\n s t xh xl yh yi zh zl i1 i2 s \r\n s t xh xl yh yi zh zl i1 i2 s \r\n
```

Data has to be organized in strings without START CHARS ("s" and "t") and END CHARS ("\r" and "\n", that are carriage return and new line):

```
...
(string #1) xh xl yh yi zh zl i1 i2 s
(string #2) xh xl yh yi zh zl i1 i2 s
(string #3) xh xl yh yi zh zl i1 i2 s
...
```

Before starting to decode the stream, character “s” is firstly checked followed by the “t” character: if both characters have been received in sequence, it means that the incoming string is correct and it is then possible to start decoding.

Decoding, as already mentioned, is executed by the `Manage_Input_Buffer()` function after reading the input buffer; the function returns with an ordered two-dimensional byte array and the number of complete strings to easily reconstruct sensor data and convert them from two's complement to magnitude and sign; after this, data are shown in the right TextBox.

3.7 Events

3.7.1 USB_DataReceived

Each time new data are written on the serial port, the function `USB_DataReceived()` is called automatically from the main thread.

The target of this function is to read the character on the serial port and call `String_Vector_Decode()` only if the character read is equal to “s” (START CHAR; for details, refer to UM2116 on www.st.com).

3.7.2 CBX_Kit_SelectedIndexChanged

This command reads the kit selected on CBX_Kit ComboBox.

3.8 Buttons

3.8.1 BTN_Read_Click

This function performs the register reading, that is: a stop command is sent to stop, eventually, the data stream; the software gets the address register from TB_Address TextBox and, if it is valid, composes the command.

MKI read is used as different kinds of sensors use different kinds of read/write commands.

The complete commands are:

- Accelerometer: `*rAA`
- Gyroscope: `*grAA`
- Magnetometer: `*mrAA`
- Pressure: `*prAA`

After the command is sent, the board replies with the register value: an example of the string format is “RAAhVVh”, where AA is the address and VV is the value.

TB_Value text is finally updated with the register value.

3.8.2 BTN_Write_Click

This is similar to the read function, but in this case the software has only to send a write command, composed of prefix, address and new register value.

Revision history

Table 2. **Document revision history**

Date	Version	Changes
21-Apr-2011	1	Initial release.
07-Jun-2012	2	- Added support to STEVAL-MKI108V1 and STEVAL- MKI120V1 demonstration kits. - Added description of RadioButton (*Section 4.9*) and Custom regions (*Section 4.10*). - Updated *Figure 1*, *Figure 2*, *Figure 3* and *Figure 4*. - Added list item 1 in the numbered list of GUI uses on *page 4*
28-May-2018	3	Updated all content to reflect Unico Lite GUI ver. 3.0.0.0.

Contents

List of figures

List of tables