

## Unico 精简版软件指南

### 引言

Unico Lite 图形用户界面 (GUI) 是完整的评估软件，它提供源代码，用于管理常见 MEMS 传感器（例如加速度计、陀螺仪、磁力计和气压传感器）的传感器数据流。

本用户手册将介绍所有 Unico Lite GUI 组件。GUI 与 STEVAL-MKI109V3（专业 MEMS 工具板）交互，此板是与所有 ST MEMS 适配器板兼容的母板；关于指令或数据格式的详细信息，请参考 ST 网站 [www.st.com](http://www.st.com) 上的 UM2116。

## 1 Unico Lite 图形用户界面

Unico Lite 软件经过设计，可工作于 Microsoft® Windows 平台，使用 Microsoft® Visual Studio 2017 编写。

Unico Lite 图形用户界面是一款简单的 Windows 窗口应用，用 C# (.NET Framework 3.5) 编写，设计用于显示基本操作，例如板连接、读/写寄存器传感器和如何从 STEVAL-MKI109V3 专业 MEMS 工具板获取连续数据。

下述基本概念适用于不同传感器；源代码可以直接应用于四种不同的评估板（但是，您可以轻松地为其他陀螺仪、加速度计和模块添加新的评估板）：

- 用于 LSM6DSL 6 轴 iNEMO 惯性模块的 STEVAL -MKI178V1
- 用于 LIS2DW12 3 轴数字加速度计的 STEVAL- MKI179V1
- 用于 LIS2MDL 3 轴数字磁力计的 STEVAL- MKI181V1
- 用于 LPS22HB 数字气压传感器的 STEVAL- MET001V1

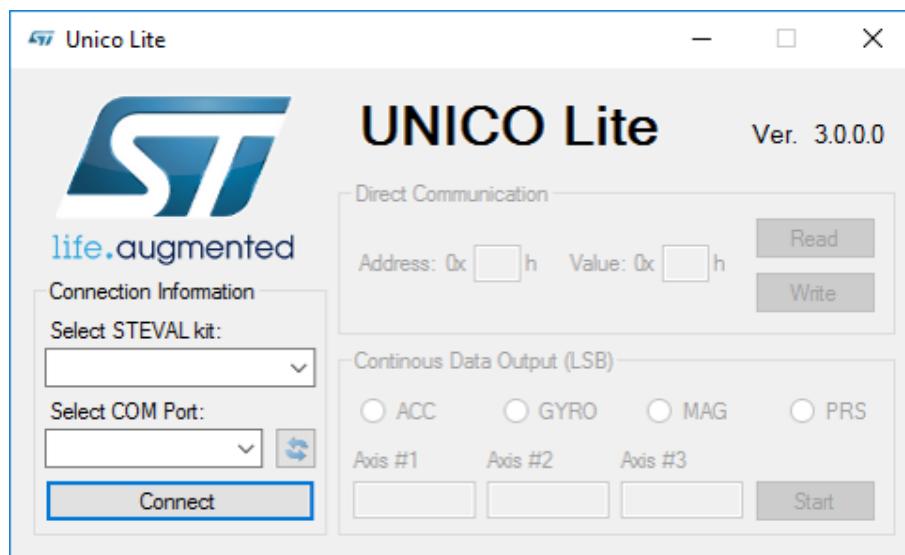
执行 Unico Lite 软件 GUI：

**Step 1.** 通过 USB 端口将 STEVAL-MKI109V3 评估板插到 PC 上。

**Step 2.** 点击[Unico\_Lite]。

GUI 窗口显示：

图 1. Unico Lite GUI 窗口

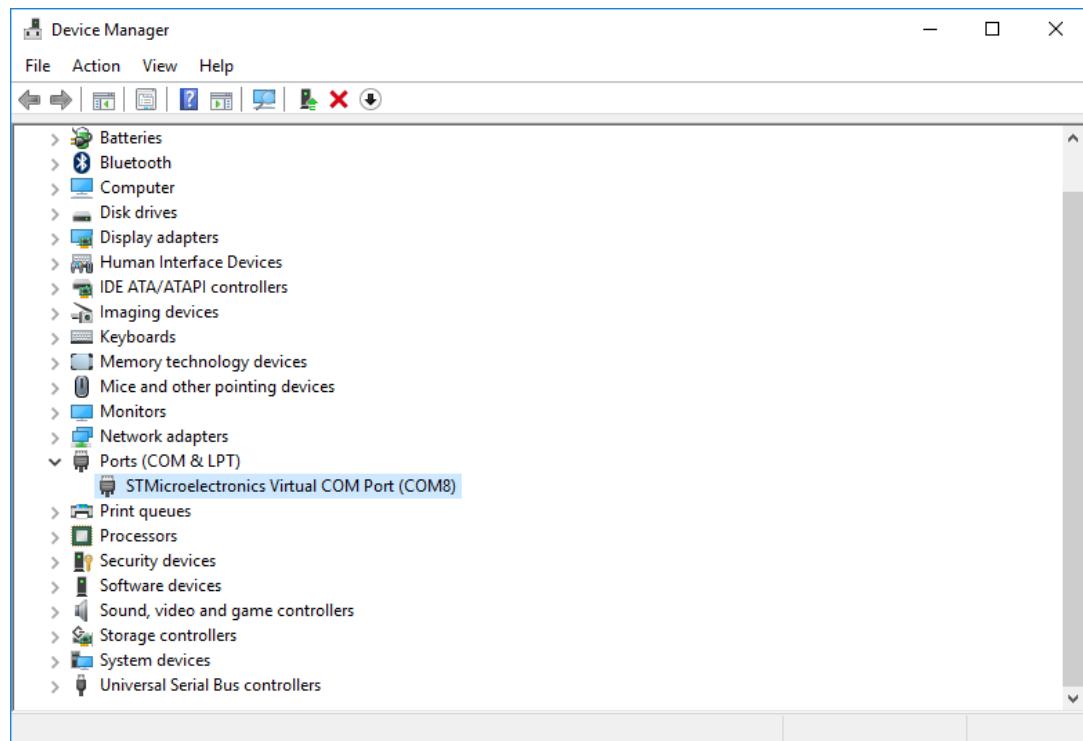


**Step 3.** Unico Lite 自动搜索可用的 STMicroelectronics Virtual COM Port 设备，并相应地设置[选择 COM 端口]组合框。

如果找到的 COM 端口不正确，打开[设备管理器]并滚动列表，直至找到[Ports (COM & LPT)]，并找到 STMicroelectronics Virtual COM Port。

在下图所示的示例中，COM8 已分配给评估板。

图 2. Unico Lite GUI Virtual COM Port 的分配



**Step 4.** 从[选择 STEVAL 套件]组合框中选择当前使用的适配器板。

**Step 5.** 点击[连接]。

现在，您可以使用 GUI：

- 使用四个单选按钮选择一个传感器（加速度计、陀螺仪、磁力计或气压传感器）
- 读取寄存器：在地址框中插入寄存器地址（十六进制值）并点击[读取]按钮。寄存器内容显示在值框中。
- 写入寄存器：在地址框中插入寄存器地址（十六进制值），在值框中插入寄存器值（十六进制值），并点击[写入]按钮。
- 连续获取数据：点击[开始]按钮并检查传感器数据。

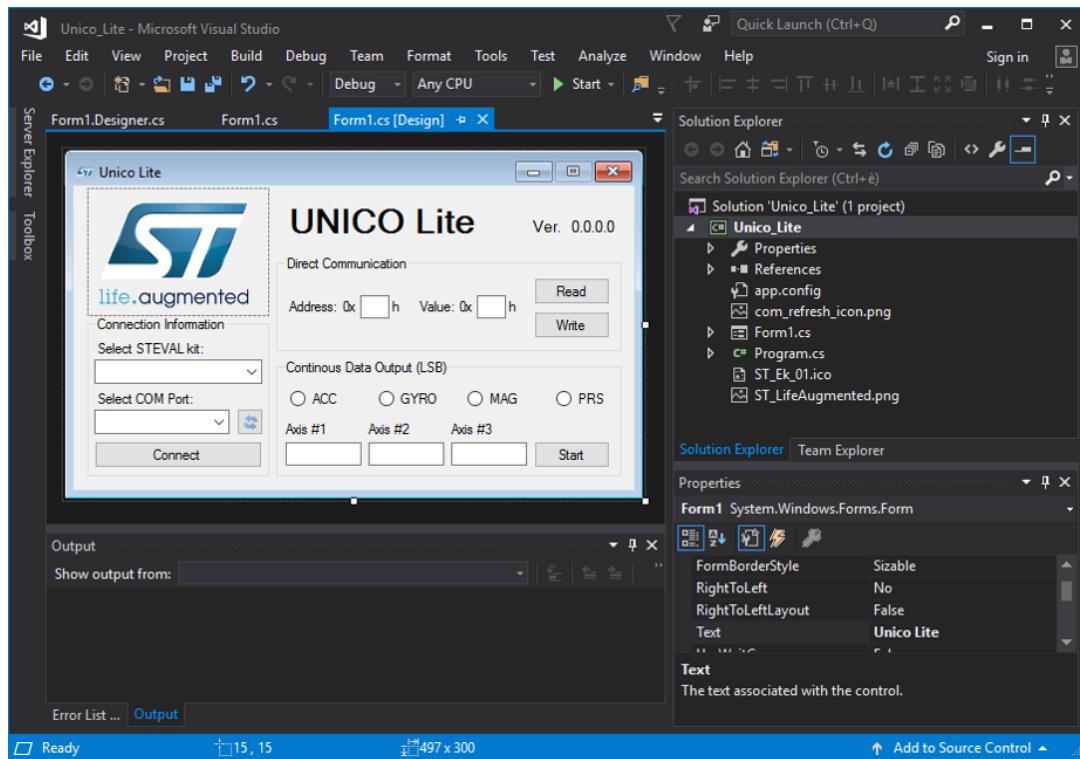
#### 提示

对于连续数据，检查寄存器设置，使传感器处于正常或低功耗模式。目的是为了防止传感器被配置成下电模式而没有数据输出，从而导致框中无数据显示。

## 2 Windows 窗口设计

使用了 Microsoft® Visual Studio 2017 设计窗口应用。

图 3. Microsoft® Visual Studio 2017 主界面



Unico Lite GUI (参见下图) 包含表 1. Unico Lite 的主要控件中描述的控件。

图 4. Unico Lite 的控件

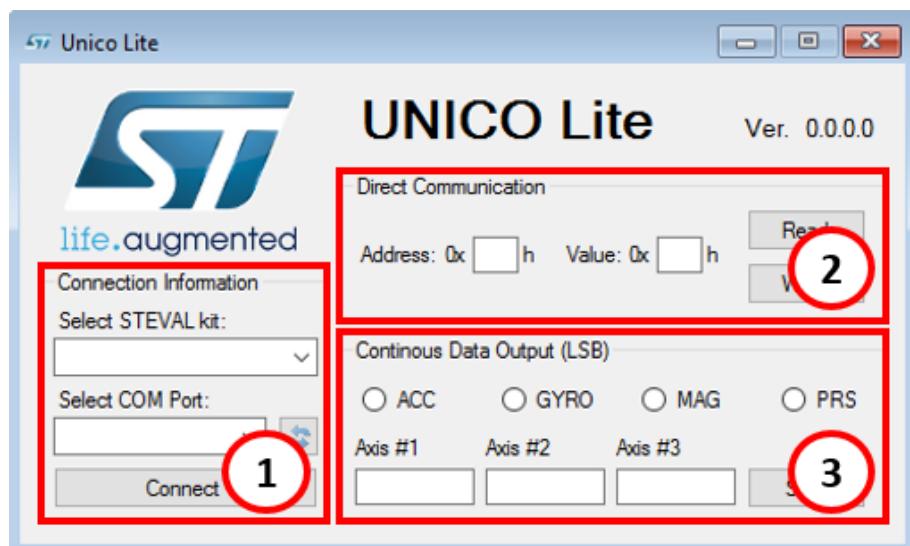


表 1. Unico Lite 的主要控件

参考	名称	控制类型
1 (连接管理)	CBX_Kit	ComboBox
	CBX_ComPorts	ComboBox
	BTN_ComRefresh	按钮
	BTN_Connect	按钮
2 (直接读/写寄存器)	TB_Address	文本框
	TB_Value	文本框
	BTN_Read	按钮
	BTN_Write	按钮
3 (连续数据读取)	RB_Acc	单选按钮
	RB_Gyro	单选按钮
	RB_Mag	单选按钮
	RB_Prs	单选按钮
	TB_Val1	文本框
	TB_Val2	文本框
	TB_Val3	文本框
	BTN_Start	按钮

## 3 代码

### 3.1 常量

- `const int MAX_KITS:` 定义代码支持的最大套件数量。在该版本中，软件支持 4 个套件。

```
• public struct Kits
  {
    public string Name;
    public int words;
    public string setdb;
  }
```

定义可通过以下方式描述的单一套件：

1. 名称: STEVAL 套件代码
2. 字数: 在\*start 指令后返回的数据字节数 (请参考 ST 网站 [www.st.com](http://www.st.com) 上 UM2116 中的表 3)
3. `setdb:` 选择能够处理适配器板的固件组件 (例如, 对于 STEVAL-MKI178V1, 指令为\*setdb178v1)

### 3.2 变量

- `Kits[] MKI = new Kits[MAX_KITS]:` 为套件说明创建结构体数组
- `int Kit_Index:` 组合框套件索引的全局值 (默认值 = -1)
- `string DataReadFromSerialPort:` 用于数据读取的临时字符串
- `bool Start:` 用于监测[开始]按钮是否已按下的标记
- `volatile bool Connected:` 用于监测连接状态定义的易失性 (因为用在不同的线程中) 的标记
- `public string readAcc:` 加速度计传感器的读指令前缀
- `public string writeAcc:` 加速度计传感器的写指令前缀
- `public string readGyr:` 陀螺仪传感器的读指令前缀
- `public string writeGyr:` 陀螺仪传感器的写指令前缀
- `public string readMag:` 磁力计传感器的读指令前缀
- `public string writeMag:` 磁力计传感器的写指令前缀
- `public string readPrs:` 气压传感器的读指令前缀
- `public string writePrs:` 气压传感器的写指令前缀

### 3.3 类

- `public SerialPort c_Serial:` 用于控制串行端口文件资源。此类提供同步的事件驱动 I/O, 访问引脚、断线状态和串行驱动器属性。

## 3.4 初始化

用于初始化控件和类的代码如下：

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

    // 获取文件版本
    LB_VersionValue.Text = GetFileVersion();

    // 填充评估板数组
    // 用于 LSM6DSL Combo (加速度计和陀螺仪) 传感器的 STEVAL-MKI178V1
    MKI[0].Name = "MKI178V1 (LSM6DSL)";
    MKI[0].words = 22;
    MKI[0].setdb = "*setdb178v1";

    // 用于 LIS2DW12 加速度计传感器的 STEVAL-MKI179V1
    MKI[1].Name = "MKI179V1 (LIS2DW12)";
    MKI[1].words = 13;
    MKI[1].setdb = "*setdb179v1";

    // 用于 LIS2MDL 磁力计传感器的 STEVAL-MKI181V1
    MKI[2].Name = "MKI181V1 (LIS2MDL)";
    MKI[2].words = 12;
    MKI[2].setdb = "*setdb181v1";

    // 用于 LPS22HB 气压传感器的 STEVAL-MET001V1
    MKI[3].Name = "MET001V1 (LPS22HB)";
    MKI[3].words = 14;
    MKI[3].setdb = "*setdb001v1";

    // 用套件填充组合框
    for (int i = 0; i < MAX_KITS; i++)
    {
        CBX_Kit.Items.Add(MKI[i].Name);
    }

    // 初始化串行端口对象
    c_Serial = new SerialPort();
    DetectCOMPorts();
}
```

## 3.5 连接

这部分代码负责管理串行端口操作（例如连接、断开、打开和关闭）和按钮迭代。

### 3.5.1 DetectCOMPorts

用于根据系统中发现的“STMicroelectronics Virtual COM Port”自动填充 CBX\_ComPorts 组合框。在 GUI 初始化时调用。

### 3.5.2 BTN\_ComRefresh\_Click

用于调用“DetectCOMPorts”函数，以便根据系统中发现的“STMicroelectronics Virtual COM Port”更新 CBX\_ComPorts 组合框。在打开 GUI 后连接板时可以使用它。

### 3.5.3 BTN\_Connect\_Click

它基于 **Connected** 标记管理连接/断开状态。

如果 **Connected** 标记为 **false**

GUI 必须打开通信。它配置串行端口对象并打开与评估板的连接。具体的端口设置如下：

1. PortName: 通过 CBX\_ComPort 选中的项

2. 波特率: 115200
3. DataBits: 8
4. ReadTimeout: 500 [ms]
5. WriteTimeout: 500 [ms]
6. DataReceived: 事件与方法之间的连接。每次从评估板向串行端口发送新数据时, 主线程调用方法 (在本例中, 为 `USB_DataReceived()`)。

配置后, 软件尝试打开选择的虚拟 COM 端口。如果没有发生异常, 软件将在串行端口上写入:

1. `*setdbxxxv1` (xxx 为 STEVAL 代码: 例如, `LSM6DSL` 为 178, `LIS2MDL` 为 181)
2. `*zoff`

#### 提示

这两个指令是连接的强制性指令。

此外, 单选按钮 (和可能有的取决于套件的控件) 会在连接后立即初始化。

在函数末尾, 软件执行使能/禁用控件。

如果 `Connected` 标记为 `true`

GUI 必须关闭通信。它执行下列操作:

1. 发送`*stop` 指令
2. 发送`*zon` 指令
3. 发送`*dbreset` 指令
4. 关闭串行端口
5. 使能/禁用控件
6. 重置文本框内容以清空 GUI

#### 3.5.4 `BTN_Start_Click`

它基于 `Start` 标记管理开始/停止状态。

如果 `Start` 标记为 `false`

它使能/禁用控件并发送`*start` 指令。`Start` 标记置为 `true` 且 `BTN_Start` 文本置为“Stop”。

如果 `Start` 标记为 `true`

它使能/禁用控件并发送`*stop` 指令。`Start` 标记置为 `false` 且 `BTN_Start` 文本置为“Start”。

#### 3.5.5 `CloseSerialPort`

用于关闭串行端口连接。

#### 3.6 解码

该函数的主要用途是从数据流获取传感器数据并显示数据。

数据流是一个字符序列, 例如:

```
... 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
```

数据必须采用字符串格式, 无 START CHARS (“s”和“t”) 和 END CHARS (“\r”和“\n”, 即回车和换行):

```
...
(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
...
```

在开始解码数据流之前, 首先检查字符“s”, 然后是字符“t”: 如果按顺序接收到这两个字符, 表示输入字符串正确, 可以开始解码。

正如前文所述, 在读取输入缓冲区后, 由 `Manage_Input_Buffer()` 函数执行解码; 函数返回按顺序排列的二维字节数组和完整字符串的数量, 以便轻松地重构传感器数据并将它们从 2 的补码转换为有符号数据; 之后, 在正确的文本框中显示数据。

## 3.7 事件

### 3.7.1 USB\_DataReceived

每次串行端口上写入新数据时，自动从主线程调用函数 `USB_DataReceived()`。

此函数的目标是读取串行端口上的字符，并在读取的字符等于“s”（START CHAR；请参考 ST 网站 [www.st.com](http://www.st.com) 上的 UM2116 获取详细信息）时调用 `String_Vector_Decode()`。

### 3.7.2 CBX\_Kit\_SelectedIndexChanged

该指令读取在 `CBX_Kit` 组合框中选择的套件。

## 3.8 按钮

### 3.8.1 BTN\_Read\_Click

此函数执行寄存器读操作，即：发送停止指令最终停止数据流；软件从 TB\_Address 文本框获取地址寄存器值，若有效，则压缩指令。

MKI read 用作不同类型的传感器，使用不同类型的读/写指令。

完整指令如下：

- 加速度计：\*rAA
- 陀螺仪：\*grAA
- 磁力计：\*mrAA
- 气压：\*prAA

在发送指令后，板回复以寄存器值：字符串格式的示例是“RAAhVVh”，其中的 AA 是地址，VV 是值。

最后，用寄存器值更新 TB\_Value 文本。

### 3.8.2 BTN\_Write\_Click

此函数类似于读函数，但在本例中，软件只需发送由前缀、地址和新寄存器值组成的写指令。

## 版本历史

表 2. 文档版本历史

日期	版本	变更
2011 年 4 月 21 日	1	初始版本。
2012 年 6 月 7 日	2	<ul style="list-style-type: none"><li>- 增加了对 STEVAL-MKI108V1 和 STEVAL- MKI120V1 演示套件的支持。</li><li>- 增加了对单选按钮（第 4.9 节）和自定义区域（第 4.10 节）的描述。</li><li>- 更新了图 1、图 2、图 3 和图 4。</li><li>- 在第 4 页 GUI 的编号列表中增加了列表项 1</li></ul>
2018 年 5 月 28 日	3	为反映 Unico Lite GUI 版本更新了所有内容 3.0.0.0

## 目录

1	Unico Lite 图形用户界面 . . . . .	2
2	Windows 窗口设计 . . . . .	4
3	代码 . . . . .	6
3.1	常量 . . . . .	6
3.2	变量 . . . . .	6
3.3	类 . . . . .	6
3.4	初始化 . . . . .	7
3.5	连接 . . . . .	7
3.5.1	DetectCOMPorts . . . . .	7
3.5.2	BTN_ComRefresh_Click . . . . .	7
3.5.3	BTN_Connect_Click . . . . .	7
3.5.4	BTN_Start_Click . . . . .	8
3.5.5	CloseSerialPort . . . . .	8
3.6	解码 . . . . .	8
3.7	事件 . . . . .	9
3.7.1	USB_DataReceived . . . . .	9
3.7.2	CBX_Kit_SelectedIndexChanged . . . . .	9
3.8	按钮 . . . . .	10
3.8.1	BTN_Read_Click . . . . .	10
3.8.2	BTN_Write_Click . . . . .	10
	版本历史 . . . . .	11

## 图一览

图 1.	Unico Lite GUI 窗口 . . . . .	2
图 2.	Unico Lite GUI Virtual COM Port 的分配 . . . . .	3
图 3.	Microsoft® Visual Studio 2017 主界面 . . . . .	4
图 4.	Unico Lite 的控件 . . . . .	4

## 表一览

表 1.	Unico Lite 的主要控件 . . . . .	5
表 2.	文档版本历史 . . . . .	11

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2019 STMicroelectronics - 保留所有权利