VisualHMI - 数据记录控件

本章节主要介绍LUA脚本API 对数值、字符串类型数据记录控件的使用说明。

> [!note|tip:注意]
> 数据记录控件的功能，本章节只介绍需要LUA脚本设置部分，其他界面上直接配置的功能（如记录的上下翻页、时间戳定位等）不做说明，详细可参考[VisualHMI - 数据记录控件 (点击跳转)]

> 适用范围：VisualHMI - HMI&M系列&Dx系列
>
> 例程下载链接：[ViusalHMI - 数据记录控件(点击下载)]

1.Lua API说明

1.1.record_get_count(sn)

获取资料采样记录总条数函数，`record_get_count(sn)` 是 HMI 系统提供的**数据记录查询接口**，用于获取指定资料采样通道（索引为 `sn`）中**已存储的历史记录总条数**

📊 **参数说明**

参数	类型	说明
`sn`	number	**资料采样通道索引** • 取值范围：0 ~ 19（共 20 个通道）

1.2.record_modify_data(sn, index, data)

修改资料记录内容函数，`record_modify_data(sn, index, data)` 是 HMI 系统提供的**数据记录修改接口**，用于**修改指定资料采样通道中某一条历史记录的数据内容**

📊 **参数说明**

参数	类型	说明
`sn`	number	**资料记录通道索引** • 取值范围：0 ~ 19 • 对应工程中配置的第 `sn` 个采样通道
`index`	number	**记录行号（从 0 开始）** • 表示要修改的第 `index` 条记录 • 必须小于 `record_get_count(sn)` 返回值
`data`	table (array)	**待写入的数据数组** • 字一维数组，Lua 表索引从 1 开始

⚠️ **重要限制**：若该资料采样已启用**块地址存储模式**，数据不可篡改性。

1.3.record_write_data(sn, data)

添加资料记录函数，`record_write_data(sn, data)` 是 HMI 系统提供的**数据记录写入接口**，用于**向指定资料采样通道追加一条新的记录。**

📊 **参数说明**

参数	类型	说明
`sn`	number	**资料记录通道索引** • 取值范围：0 ~ 19 • 对应工程中已配置的第 `sn` 个采样通道
`data`	table (array)	**待写入的数据数组** • 字一维数组，Lua 表索引从 1 开始

1.4.record_read_data(sn, index)

读取资料记录内容函数，`record_read_data(sn, index)` 是 HMI 系统提供的**历史数据查询接口**，用于从指定资料采样通道中**读取某一条记录的原始数据及其时间戳。**

📊 **参数说明**

参数	类型	说明
`sn`	number	**资料记录通道索引** • 取值范围：0 ~ 19 • 对应工程中配置的第 `sn` 个采样通道
`index`	number	**记录行号（从 0 开始）** • 表示要读取的第 `index` 条记录 • 必须满足 `0 ≤ index < record_get_count(sn)`

返回值

返回值	类型	说明
`data`	table (array)	**记录数据数组** • 元素为 **16 位无符号整数（uint16）** • **Lua 表索引从 1 开始**，即 `data[1]` 为记录的第一个 word 字段
`timestamp`	number	**32 位 Unix 时间戳（秒）**

1.5.record_modify_string(sn, index, strings)

修改字符串类型资料记录函数，`record_modify_string(sn, index, strings)` 是 HMI 系统为**字符串型资料记录通道**提供的专用接口，用于**修改指定记录行中的内容。每一列数据**以分号 `;` 分隔的字符串形式传入

📊 **参数说明**

参数	类型	说明
`sn`	number	**资料记录通道索引** • 取值范围：0 ~ 19 • 必须对应一个配置为字符串类型的采样通道
`index`	number	**记录行号（从 0 开始）** • 表示要修改的第 `index` 条记录 • 必须满足 `0 ≤ index < record_get_count(sn)`
`strings`	string	**分号分隔的字符串字段序列** • 格式：`"字段1;字段2;字段3;..."`

⚠️ **重要限制**：若该资料采样已启用**块地址存储模式**，数据不可篡改性。

1.6.record_read_string(sn, index)

读取字符串类型资料记录函数，`record_read_string(sn, index)` 是 HMI 系统为**字符串型资料记录通道**提供的专用查询接口，用于**读取指定记录行的内容及其时间戳。**返回的数据以分号 `;` 分隔的字符串

📊 **参数说明**

参数	类型	说明
`sn`	number	**资料记录通道索引** • 取值范围：0 ~ 19
`index`	number	**记录行号（从 0 开始）** • 表示要读取的第 `index` 条记录 • 必须满足 `0 ≤ index < record_get_count(sn)`

返回值

返回值	类型	说明
`strings`	string	**分号分隔的字符串字段序列** • 格式：`"字段1;字段2;字段3;..."`
`timestamp`	number	**32 位时间戳（秒）**

1.7.record_write_string(sn, strings)

添加字符串类型资料记录函数，`record_write_string(sn, strings)` 是 HMI 系统为**字符串型资料记录通道**提供的专用写入接口，用于**向指定通道追加一条新的文本记录。**数据以分号 `;` 分隔的字符串形式传入。

📊 **参数说明**

参数	类型	说明
`sn`	number	**资料记录通道索引** • 取值范围：0 ~ 19
`strings`	string	**分号分隔的字符串字段序列** • 格式：`"字段1;字段2;字段3;..."`

1.8.record_clear(sn)

清除资料记录函数，`record_clear(sn)` 是 HMI 系统提供的**资料记录重置接口**，用于**清空指定通道（`sn`）中的所有历史记录数据。**

📊 **参数说明**

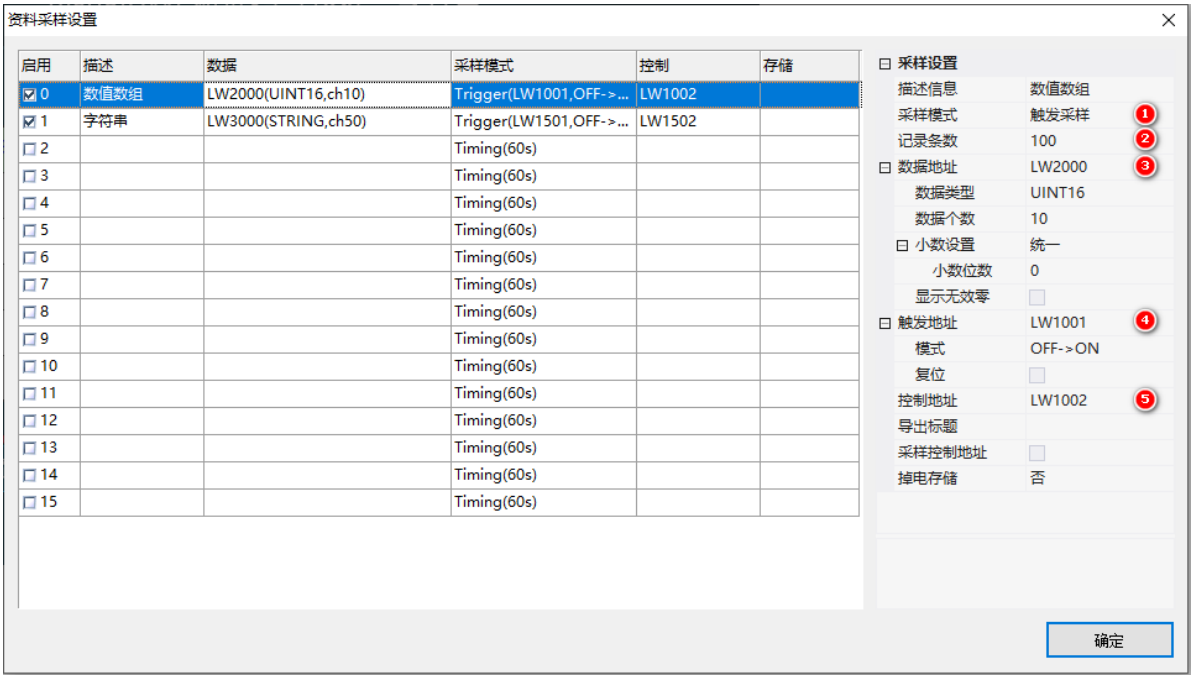
参数	类型	说明
`sn`	number	**资料记录通道索引** • 取值范围：0 ~ 19 • 对应工程中配置的第 `sn` 个采样通道（数值型或字符串型均适用）

2.数值表格

2.1.资料采集配置

在菜单栏工程→资料采样中，设置数值表格属性如下所示：

1. 采集模式：触发采样；
2. 记录条数：100；
3. 数据地址：LW2000
    - 数据类型：UINT16
    - 数据个数：10
4. 控制地址:LW1001
5. 采样控制地址：LW1002
    - 启动模式OFF→ON

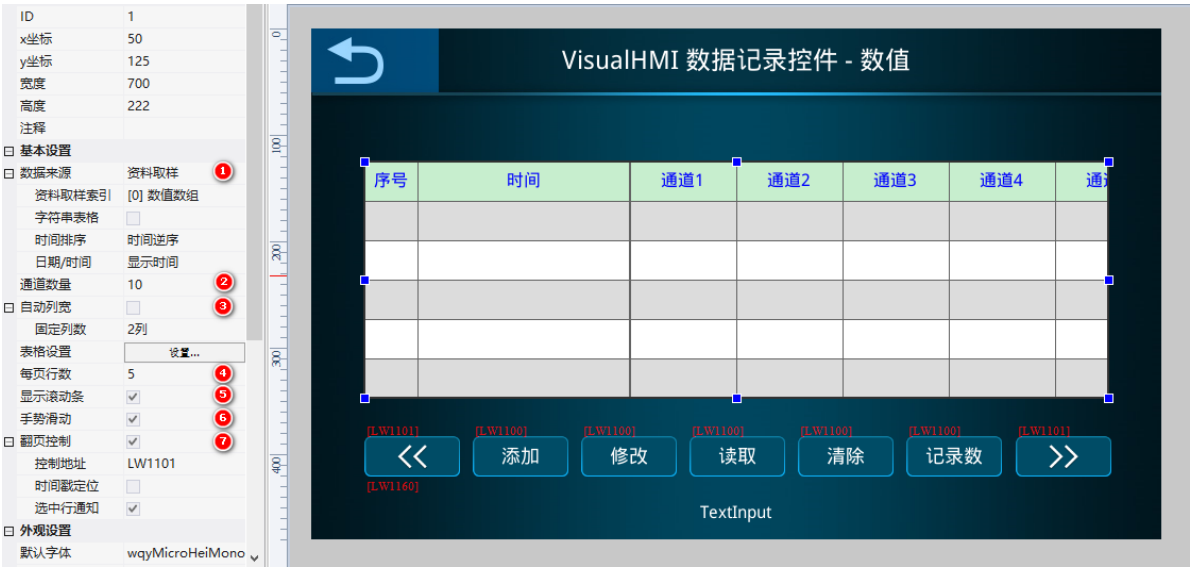


2.2.画面配置

2.2.1.数据记录控件

画面中添加一个数据记录表格，属性配置如下所示：

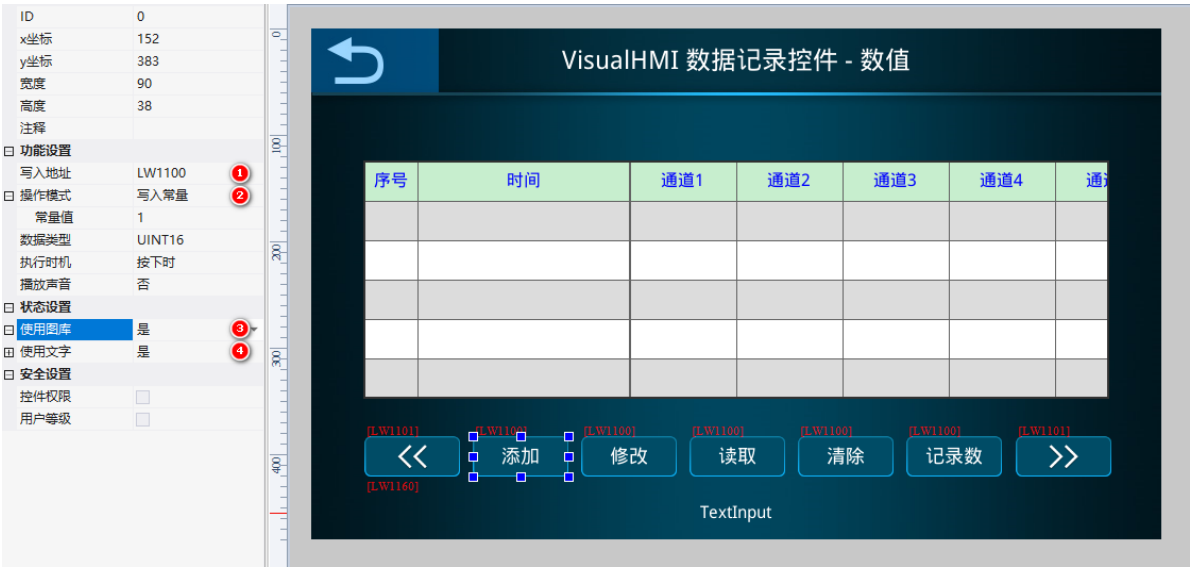
1. 数据来源：资料采集，资料取样索引为“[0]数值数组”
    - 时间顺序：时间逆序
    - 时间日期：显示时间
2. 通道数：为10。和资料采集→数据个数→10一致
3. 自动列宽：不勾选
    - 固定列数为2：即是序号、时间两列固定不动
4. 每页行数：5
5. 显示滚动条：√，当记录数大于5时，显示滚动条，可拖动滚动条查看
6. 手势滑动：√，当记录数大于5时，可以上下滑动表格查看数据
7. 控制地址：
    - 翻页控制：√，地址为LW1004，UINT16类型
    - 允许翻页：地址为LW1005，UINT16，只读
    - 时间戳定位：√，地址为LW1006，UINT32类型
    - 选中行通知：√，地址为LW1008，UINT32类型



2.2.2.Lua键设置

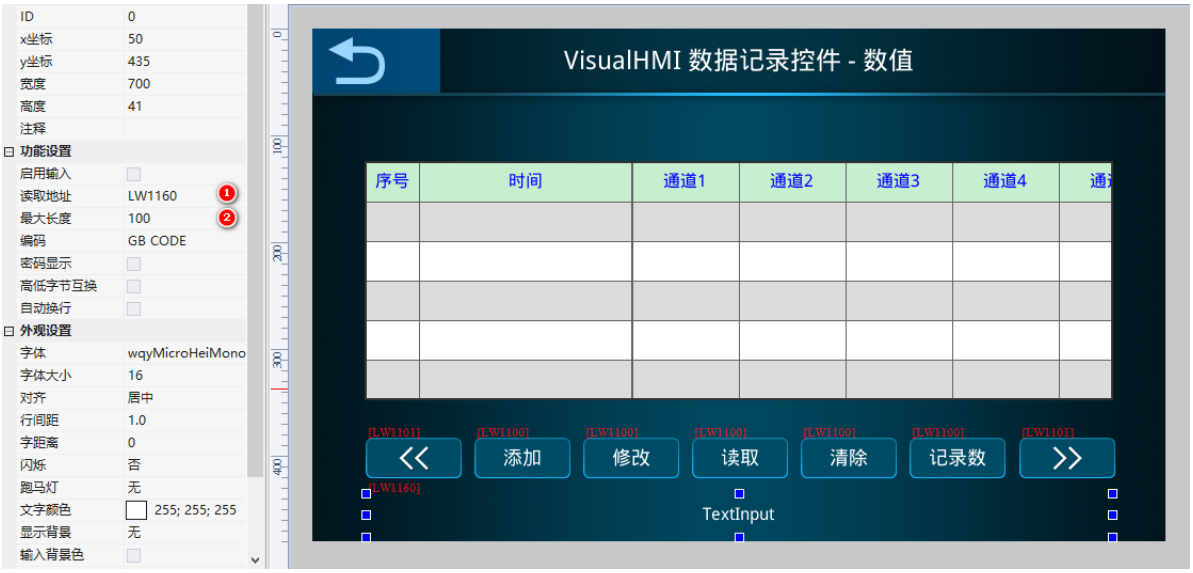
添加5个字设置按钮，用于触发Lua控制数值表格，以“添加”按钮为例，控件配置如下所示：

1. 写入地址：LW1000
2. 操作模式：写入常量
   - 常量值：1/2/3/4/5(依次为添加、修改、读取、清除、读取记录数)
3. 使用图库：√
4. 使用文字：√



2.2.3.文本设置

添加一个文本控件，用于显示表格操作的信息，如读取选中行内容，总记录数等，配置如下所示：



2.3.Lua 和 预览

2.3.1.添加记录

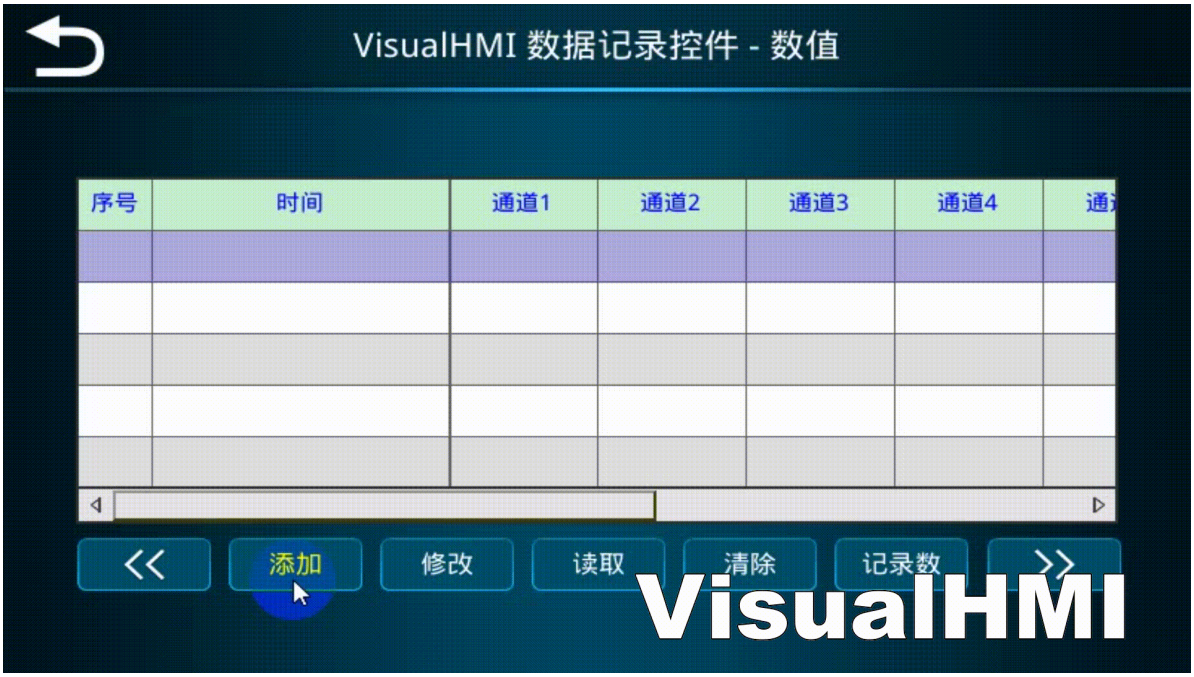
1.Lua

在on_update里面，判断触发地址和键值，若地址为LW1100、键值为1，调用record_write_data添加新的记录，程序如下所示：

```
function on_update(slave,vtype,addr)
    if vtype == VT_LW
    then
        if addr == 0x1100 --操作
        then
            local key = get_uint16(vtype, 0x1100) --获取key值，操作类型

            if key == 1 --添加记录
            then
                local dataTb = {}
                addByteRatio = addByteRatio + 1
                for i = 1, 10
                do
                    dataTb[i] = i*addByteRatio
                end
                record_write_data(0, dataTb)
            end
        end
    end
end
```

2.运行预览

运行虚拟屏，点击添加，第一次点击，添加“1,2,3,4,5,6,7,8,9,10”，在点击将添加“2,4,6,8,10,12,14,16,18,20”，如下所示：



2.3.2.修改记录

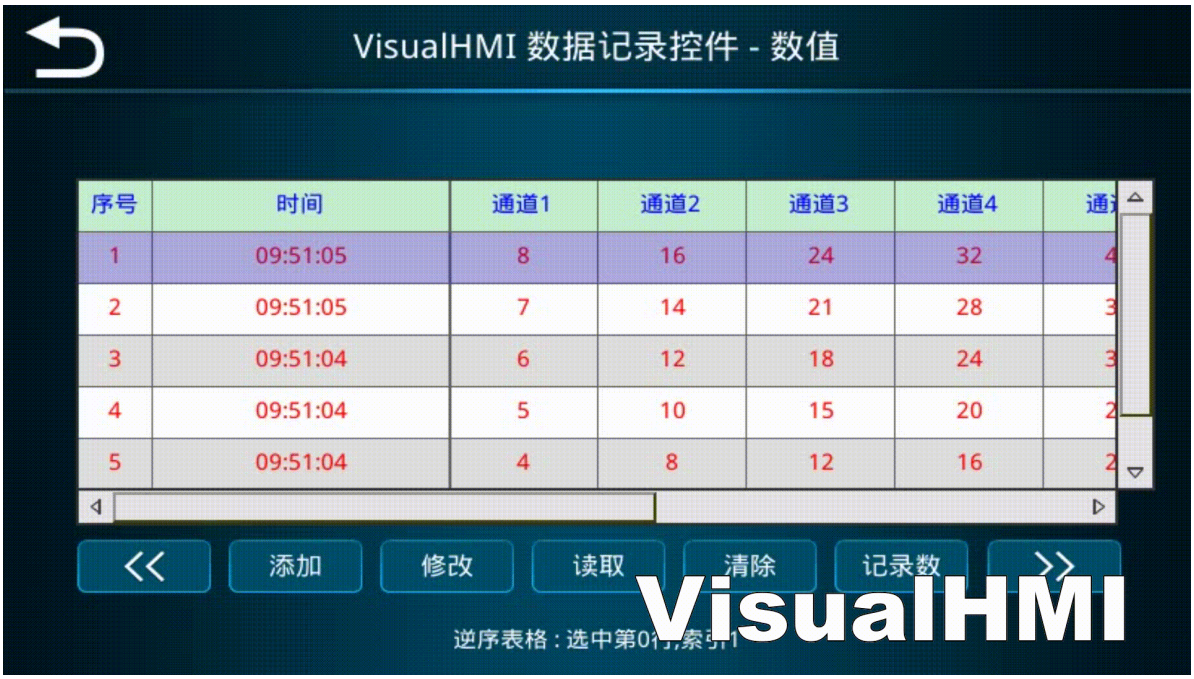
1.Lua

在on_update里面，判断触发地址和键值，若地址为LW1100、键值为2，调用record_write_data修改对应选中行内容，程序如下所示：

```
4function on_update(slave,vtype,addr)
    if vtype == VT_LW
    then
        if addr == 0x1100 --操作
        then
            local key = get_uint16(vtype, 0x1100) --获取key值，操作类型

            if key == 1 --添加记录
            then
                ....
            elseif key == 2 --修改记录
            then
                local dataTb = {}
                for i = 1, 10
                do
                    dataTb[i] = 12345
                end
                record_modify_data(0, number_select_line, dataTb)
            end

        elseif addr == 0x1105 --选择行通知
        then
            local cnt = record_get_count(0) --逆序表格
            number_select_line = cnt - get_uint32(vtype, 0x1105) - 1
        end
    end
end
```

2.运行预览

运行虚拟屏，选中第2行，点击修改，选中内容为“12345;12345;12345;12345;12345;12345;12345;12345;12345;12345;”,如下所示：



2.3.3.读取记录

1.Lua

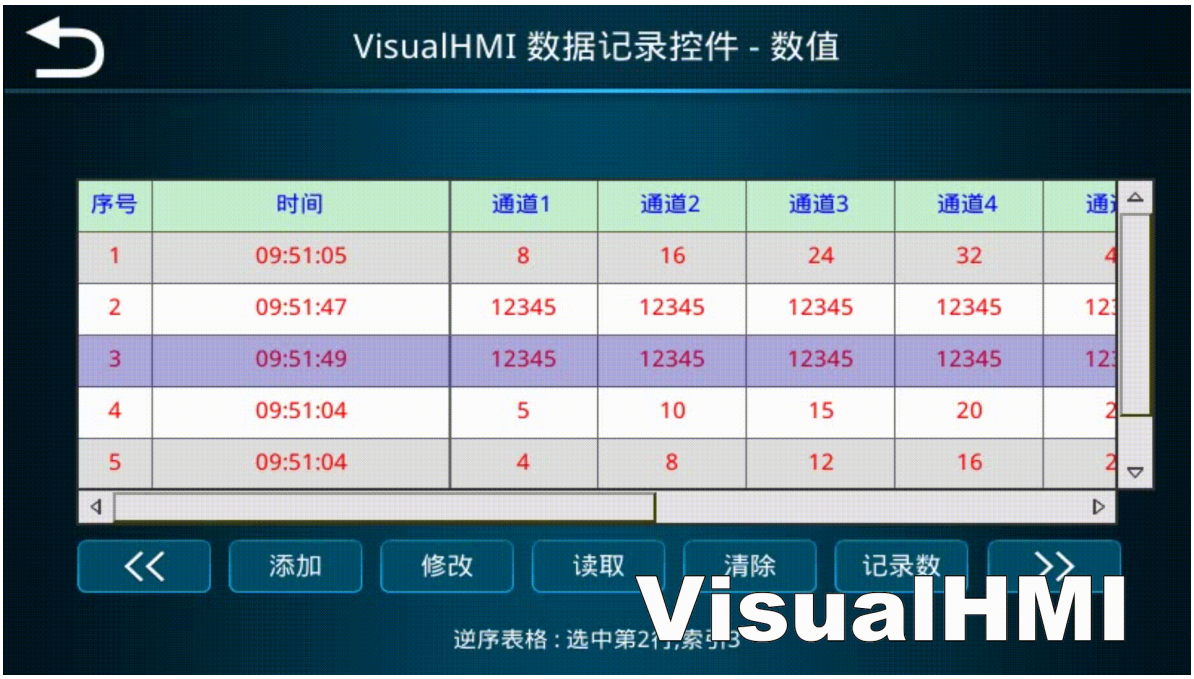
在on_update里面，判断触发地址和键值，若地址为LW1100、键值为3，调用record_read_data读取选中对应行内容，并显示在文本，程序如下所示：

```
function on_update(slave,vtype,addr)
    if vtype == VT_LW
    then
        if addr == 0x1100 --操作
        then
            local key = get_uint16(vtype, 0x1100) --获取key值，操作类型

            if key == 1 --添加记录
            then
                ....
            elseif key == 3 --读取记录
            then
                local  dataTb,timeUnix = record_read_data(0, number_select_line)
--返回参数1，数据数组；参数2，时间戳
                local dataMsg = ''
                for i = 1, #(dataTb)
                do
                    dataMsg = dataMsg..' '..dataTb[i]
                end
                set_string(VT_LW, 0x1160, timeUnix..dataMsg)
            end
        elseif addr == 0x1105 --选择行通知
        then
            local cnt = record_get_count(0) --逆序表格
            number_select_line = cnt - get_uint32(vtype, 0x1105) - 1
```

```
        end
    end
end
```

2.运行预览

运行虚拟屏，选中第2行，点击读取，将内容显示在文本上，如下所示：



2.3.4.清除记录

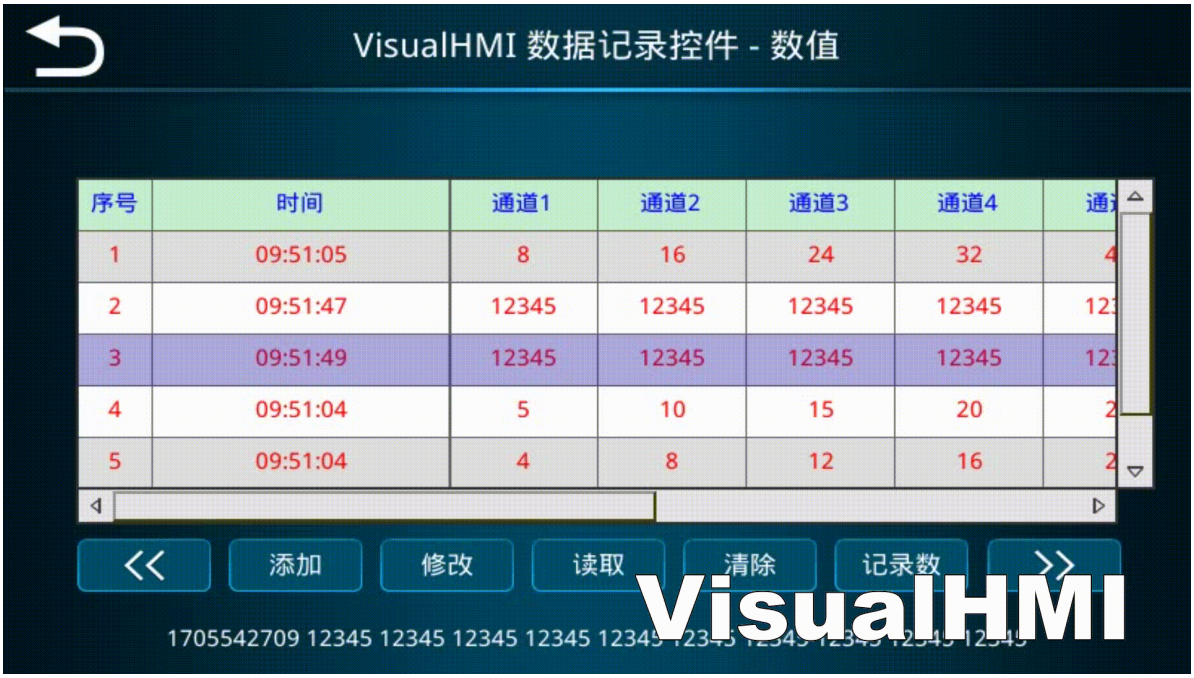
1.Lua

在on_update里面，判断触发地址和键值，若地址为LW1100、键值为4，调用record_clear清除对应的资料，程序如下所示：

```
function on_update(slave,vtype,addr)
    if vtype == VT_LW
    then
        if addr == 0x1100 --操作
        then
            local key = get_uint16(vtype, 0x1100) --获取key值，操作类型

            if key == 1 --添加记录
            then
                ....
            elseif key == 4 --清除数据
            then
                record_clear(0)
            end
        end
    end
end
```

2.运行预览

运行虚拟屏，点击清除，表格的内容清空，如下所示：



2.3.5.读取记录数

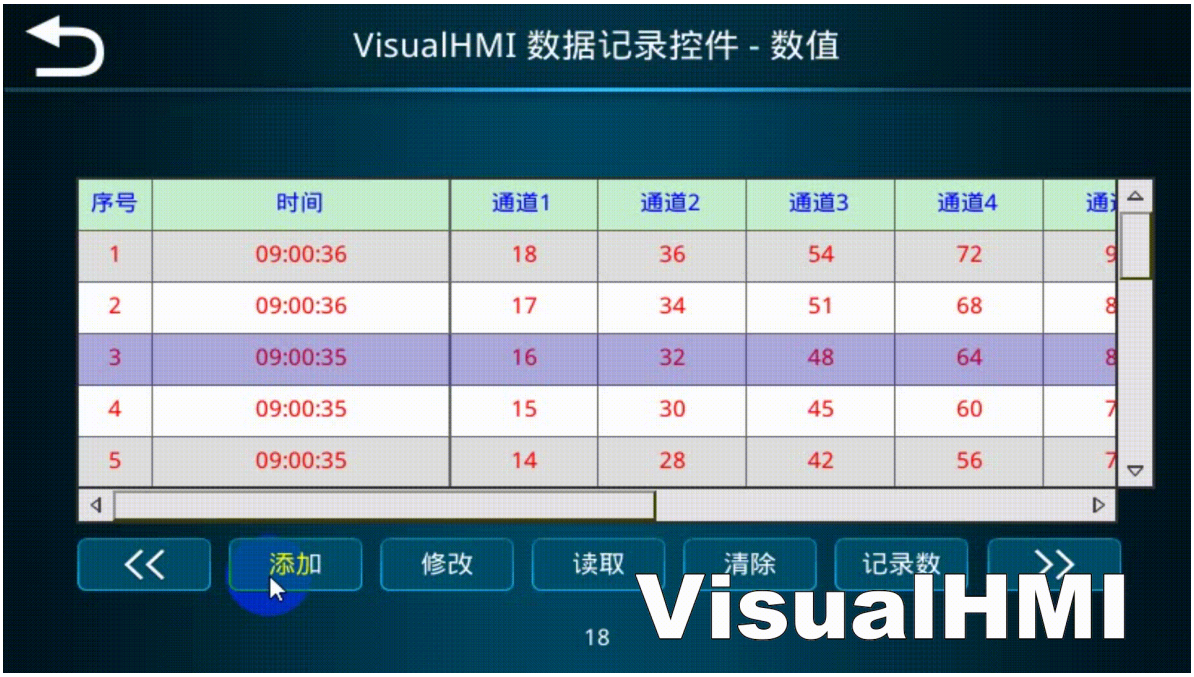
1.Lua

在on_update里面，判断触发地址和键值，若地址为LW1100、键值为5，调用record_get_count，并将记录数显示在文本上，程序如下所示：

```
function on_update(slave,vtype,addr)
    if vtype == VT_LW
    then
        if addr == 0x1100 --操作
        then
            local key = get_uint16(vtype, 0x1100) --获取key值，操作类型

            if key == 1 --添加记录
            then
                ....
            elseif key == 5 --获取记录数
            then
                local cnt = record_get_count(0)
                set_string(VT_LW, 0x1160, cnt)
            end
        end
    end
end
```

2.运行预览

运行虚拟屏，获取记录数，将当前的记录数显示在文本上，如下所示：

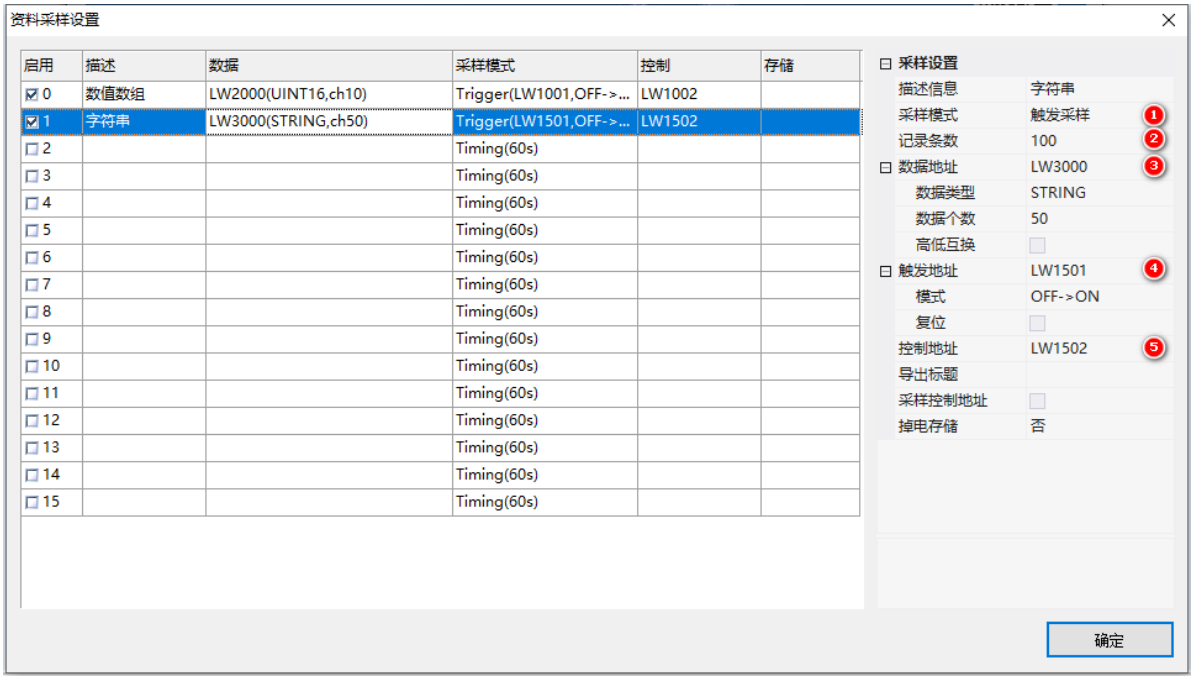


3.字符串表格

3.1.资料采集配置

在菜单栏工程→资料采样中，设置字符串表格属性如下所示：

1. 采集模式：触发采样；
2. 记录条数：100；
3. 数据地址：LW3000
   - 数据类型：STRING
   - 数据个数：50
4. 控制地址:LW1501
5. 采样控制地址：LW1502
   - 启动模式OFF→ON

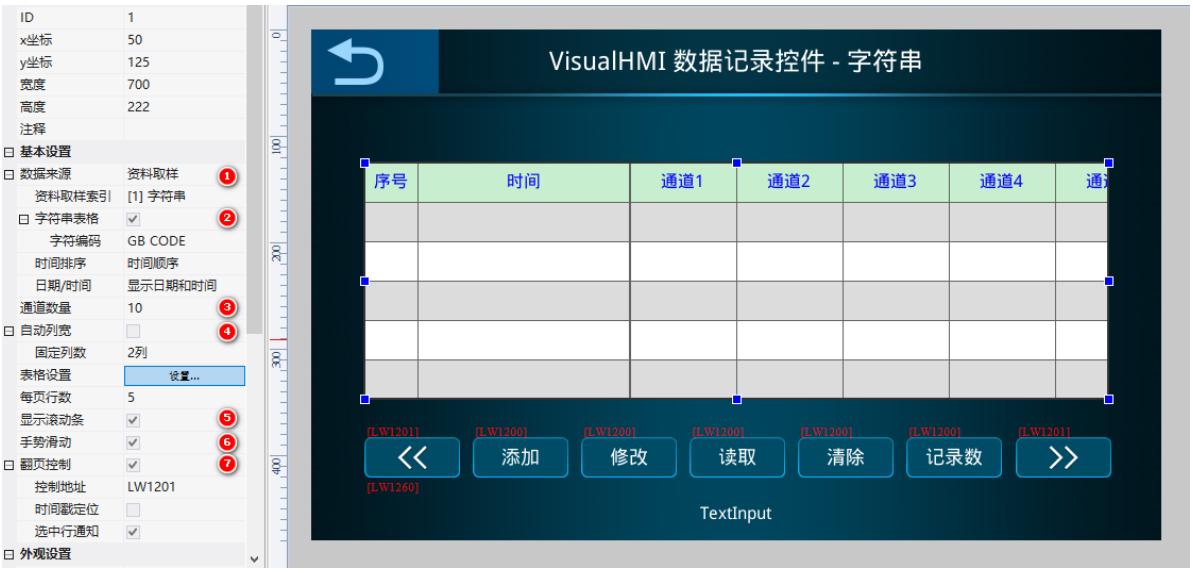


3.2.画面配置

3.2.1.数据记录控件

画面中添加一个数据记录表格，属性配置如下所示：

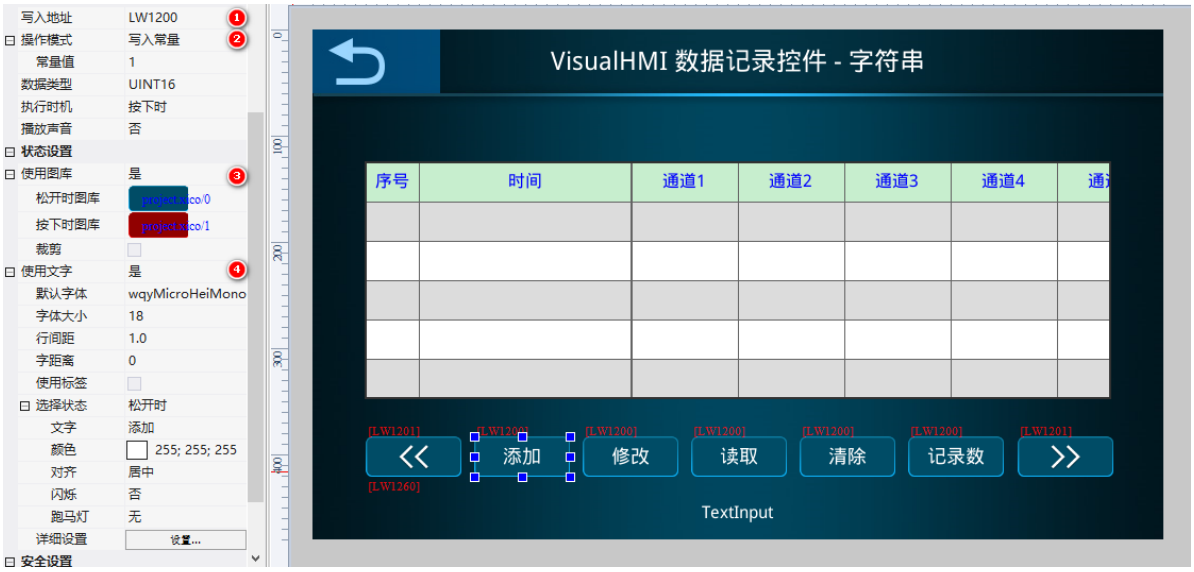
1. 数据来源：资料采集，资料取样索引为“[1]字符串”
    - 字符串表格：√
        - 字符编码：GB CODE
    - 时间顺序：时间顺序
    - 时间日期：显示日期和时间
2. 通道数：为10。和资料采集→数据个数→10一致
3. 自动列宽：不勾选
    - 固定列数为2：即是序号、时间两列固定不动
4. 每页行数：5
5. 显示滚动条：√，当记录数大于5时，显示滚动条，可拖动滚动条查看
6. 手势滑动：√，当记录数大于5时，可以上下滑动表格查看数据
7. 控制地址：LW1201
    - 翻页控制：√，地址为LW1201，UINT16类型
    - 允许翻页：地址为LW1202，UINT16，只读
    - 时间戳定位：√，地址为LW1203，UINT32类型
    - 选中行通知：√，地址为LW1205，UINT32类型



3.2.2.Lua键设置

添加5个字设置按钮，用于触发Lua控制字符串表格，以“添加”按钮为例，控件配置如下所示：

1. 写入地址：LW1200
2. 操作模式：写入常量
    - 常量值：1/2/3/4/5(依次为添加、修改、读取、清除、读取记录数)
3. 使用图库：√
4. 使用文字：√



3.2.3.文本设置

添加一个文本控件，用于显示表格操作的信息，如读取选中行内容，总记录数等，配置如下所示：

3.3.Lua 和 预览

3.3.1.添加记录

1.Lua

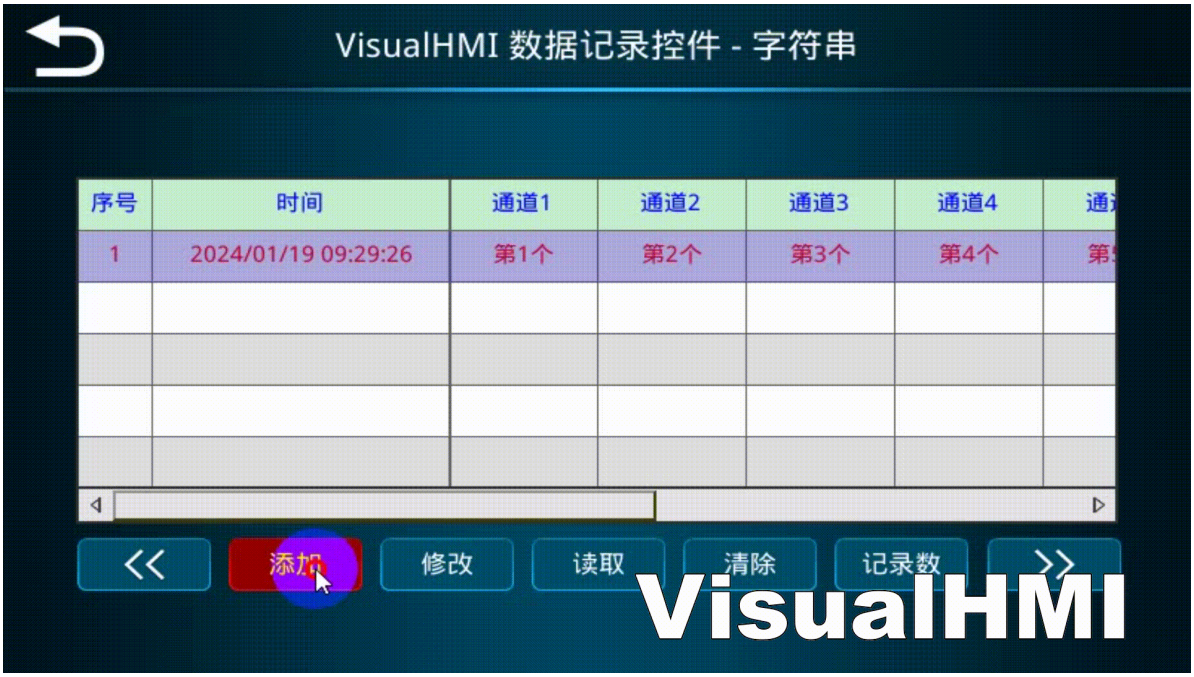
在on_update里面，判断触发地址和键值，若地址为LW1200、键值为1，调用record_write_string添加新的记录，程序如下所示：

```
function on_update(slave,vtype,addr)
    if vtype == VT_LW
    then
        if addr == 0x1200 --操作
        then
            local key = get_uint16(vtype, 0x1200) --获取key值，操作类型

            if key == 1 --添加记录
            then
                local allCnt = record_get_count(1)
                if allCnt < string_select_line - 1
```

```
                then
                    return
                end

                local dataStr = ''
                for i = 1, 10
                do
                    dataStr = dataStr..'第'..(math.ceil(addStr + i)..'个')..';'
                end
                addStr = addStr + 10
                record_write_string(1, dataStr)
            end
        end
    end
end
```

2.运行预览

运行虚拟屏，“添加”：点击表格显示“第1个;第2个;第3个;第4个;第5个;第6个;第7个;第8个;第9个;第10个;”，在点击将显示“第11个;第12个;第13个;第14个;第15个;第16个;第17个;第18个;第19个;第20个;”依次类推，显示效果，如下所示：



3.3.2.修改记录

1.Lua

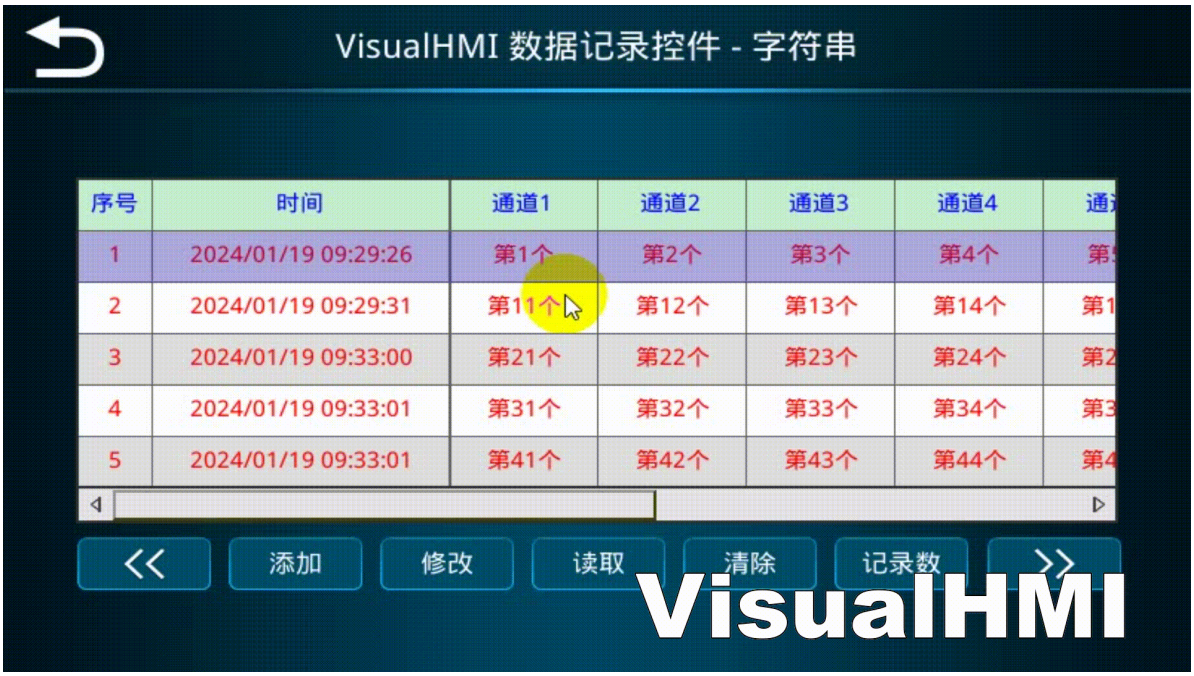
在on_update里面，判断触发地址和键值，若地址为LW1200、键值为2，调用record_modify_string修改对应选中行内容，程序如下所示：

```
function on_update(slave,vtype,addr)
    if vtype == VT_LW
    then
        if addr == 0x1200 --操作
        then
            local key = get_uint16(vtype, 0x1200) --获取key值，操作类型

            if key == 1 --添加记录
```

```
            then
                ....
            elseif key == 2 --修改记录
            then
                local allCnt = record_get_count(1)
                if allCnt < string_select_line - 1
                then
                    return
                end
                record_modify_string(1, string_select_line,
'change;change;change;change;change;change;change;change;change;change;')

            end

        elseif addr == 0x1205 --选择行通知
        then
            local cnt = record_get_count(1)
            string_select_line = get_uint32(vtype, 0x1205)
            print('string_select_line = '..string_select_line)
            set_string(VT_LW, 0x1260, '顺序表格 : 选中第'..((get_uint32(vtype,
0x1205) + 1) // 1)..'行,索引'..(string_select_line // 1))
        end
    end
end
```

2.运行预览

运行虚拟屏，选中第2行，点击修改，选中内容修改为“change;change;change;change;change;change;change;change;change;change;”,如下所示：



3.3.3.读取记录

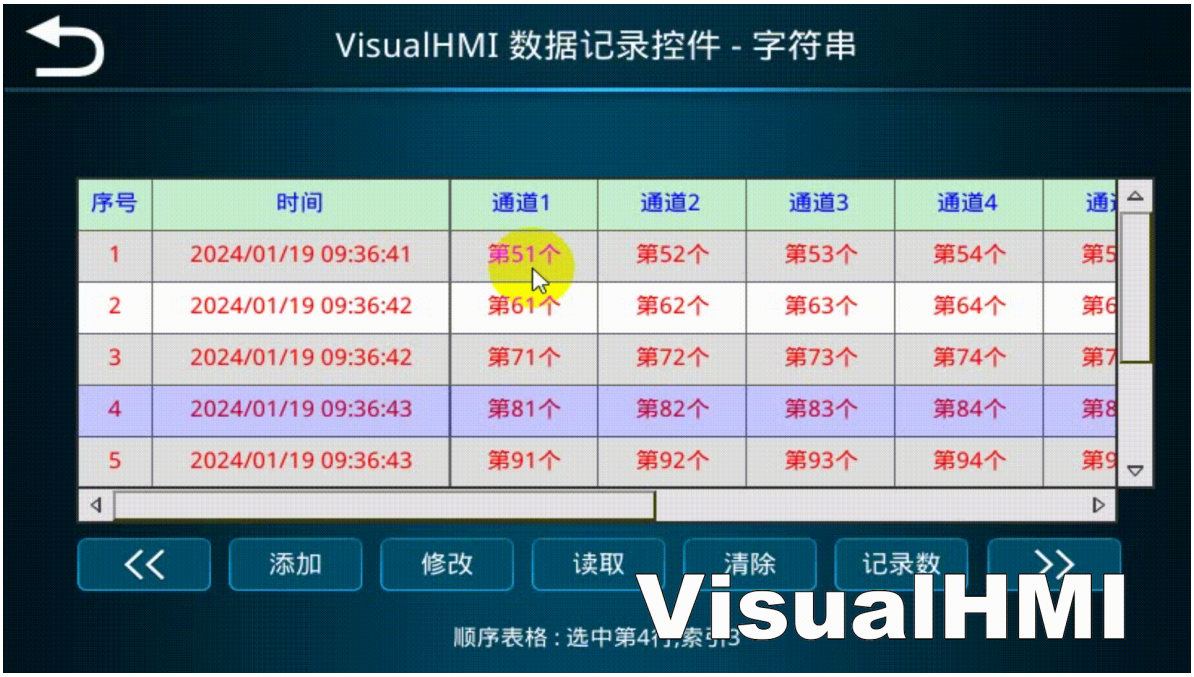
1.Lua

在on_update里面，判断触发地址和键值，若地址为LW1200、键值为3，调用record_read_string读取选中对应行内容，并显示在文本，程序如下所示：

```
function on_update(slave,vtype,addr)
    if vtype == VT_LW
    then
        if addr == 0x1200 --操作
        then
            local key = get_uint16(vtype, 0x1200) --获取key值，操作类型

            if key == 1 --添加记录
            then
                ....
            elseif key == 3 --读取记录
            then
                local allCnt = record_get_count(1)
                if allCnt < string_select_line - 1
                then
                    return
                end
                local dataStr,timeUnix = record_read_string(1,
string_select_line) --返回参数1，字符串；参数2，时间戳
                set_string(VT_LW, 0x1260, timeUnix..dataStr)
            end
        elseif addr == 0x1205 --选择行通知
        then
            local cnt = record_get_count(1)
            string_select_line = get_uint32(vtype, 0x1205)
            print('string_select_line = '..string_select_line)
            set_string(VT_LW, 0x1260, '顺序表格 : 选中第'..((get_uint32(vtype,
0x1205) + 1) // 1)..'行,索引'..(string_select_line // 1))
        end
    end
end
```

2.运行预览

运行虚拟屏，选中第2行，点击读取，将内容显示在文本上，如下所示：



3.3.4.清除记录

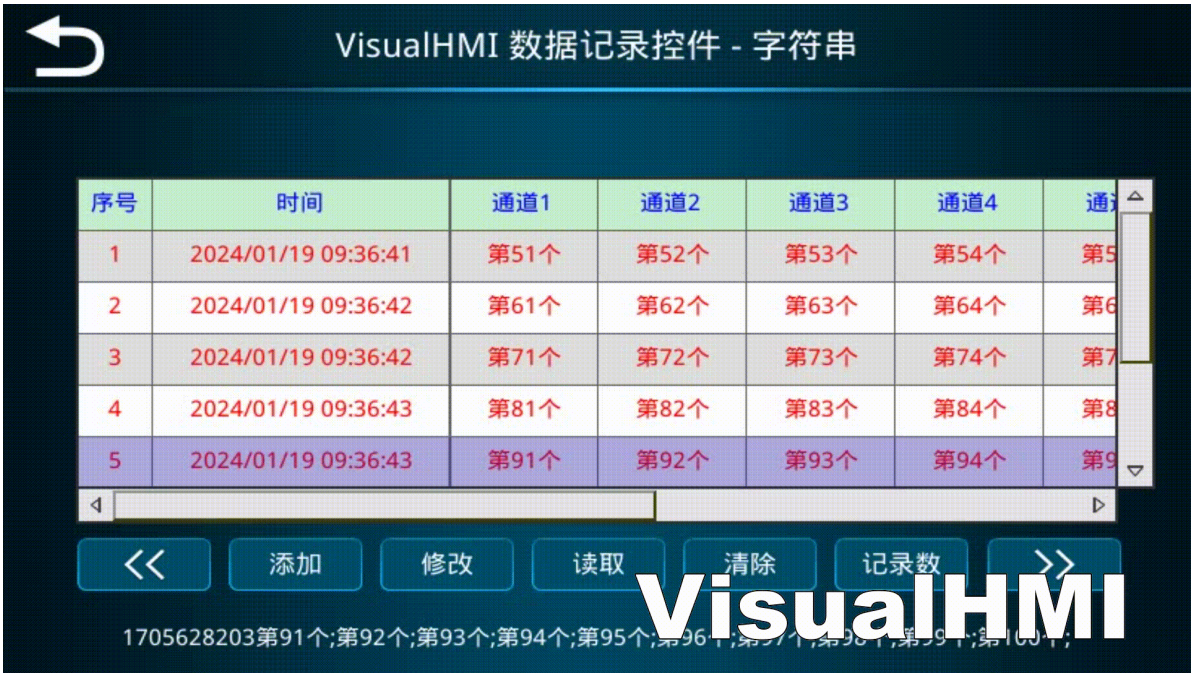
1.Lua

在on_update里面，判断触发地址和键值，若地址为LW1200、键值为4，调用record_clear清除对应的资料，程序如下所示：

```
function on_update(slave,vtype,addr)
    if vtype == VT_LW
    then
        if addr == 0x1200 --操作
        then
            local key = get_uint16(vtype, 0x1200) --获取key值，操作类型

            if key == 1 --添加记录
            then
                ....
            elseif key == 4 --清除数据
            then
                record_clear(1)
            end
        end
    end
end
```

2.运行预览

运行虚拟屏，点击清除，表格的内容清空，如下所示：



3.3.5.读取记录数

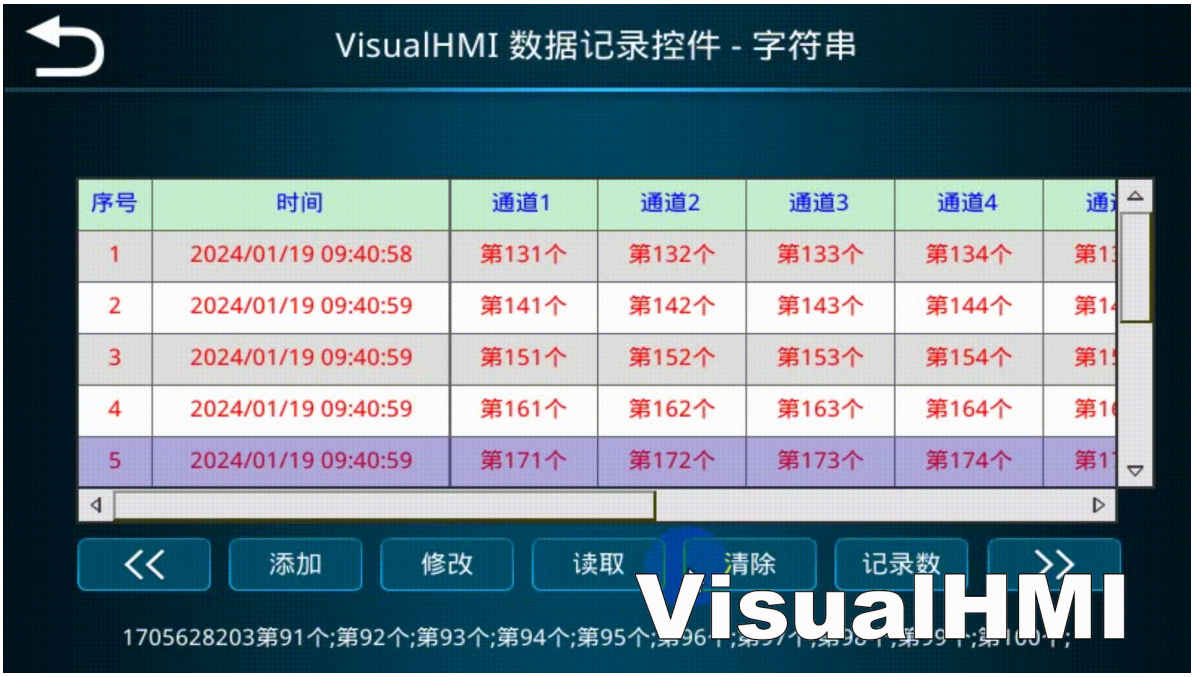
1.Lua

在on_update里面，判断触发地址和键值，若地址为LW1200、键值为5，调用record_get_count，并将记录数显示在文本上，程序如下所示：

```
function on_update(slave,vtype,addr)
    if vtype == VT_LW
    then
        if addr == 0x1200 --操作
        then
            local key = get_uint16(vtype, 0x1200) --获取key值，操作类型

            if key == 1 --添加记录
            then
                ....
            elseif key == 5 --获取记录数
            then
                local cnt = record_get_count(1)
            string_select_line = get_uint32(vtype, 0x1205)
            print('string_select_line = '..string_select_line)
            set_string(VT_LW, 0x1260, '顺序表格 : 选中第'..((get_uint32(vtype,
0x1205) + 1) // 1)..'行,索引'..(string_select_line // 1))
            end
        end
    end
end
```

2.运行预览

运行虚拟屏，获取记录数，将当前的记录数显示在文本上，如下所示：


VisualHMI 数据记录控件 - 字符串
序号
时间
通道1
通道2
通道3
通道4
1
2024/01/19 09:40:58
第131个
第132个
第133个
第134个
2
2024/01/19 09:40:59
第141个
第142个
第143个
第144个
3
2024/01/19 09:40:59
第151个
第152个
第153个
第154个
4
2024/01/19 09:40:59
第161个
第162个
第163个
第164个
5
2024/01/19 09:40:59
第171个
第172个
第173个
第174个
<<
添加
修改
读取
清除
记录数
>>
1705628203第91个;第92个;第93个;第94个;第95个;
VisualHMI