Hollong

BLE Sniffer

VTH201A

API Reference

V1.0

History：

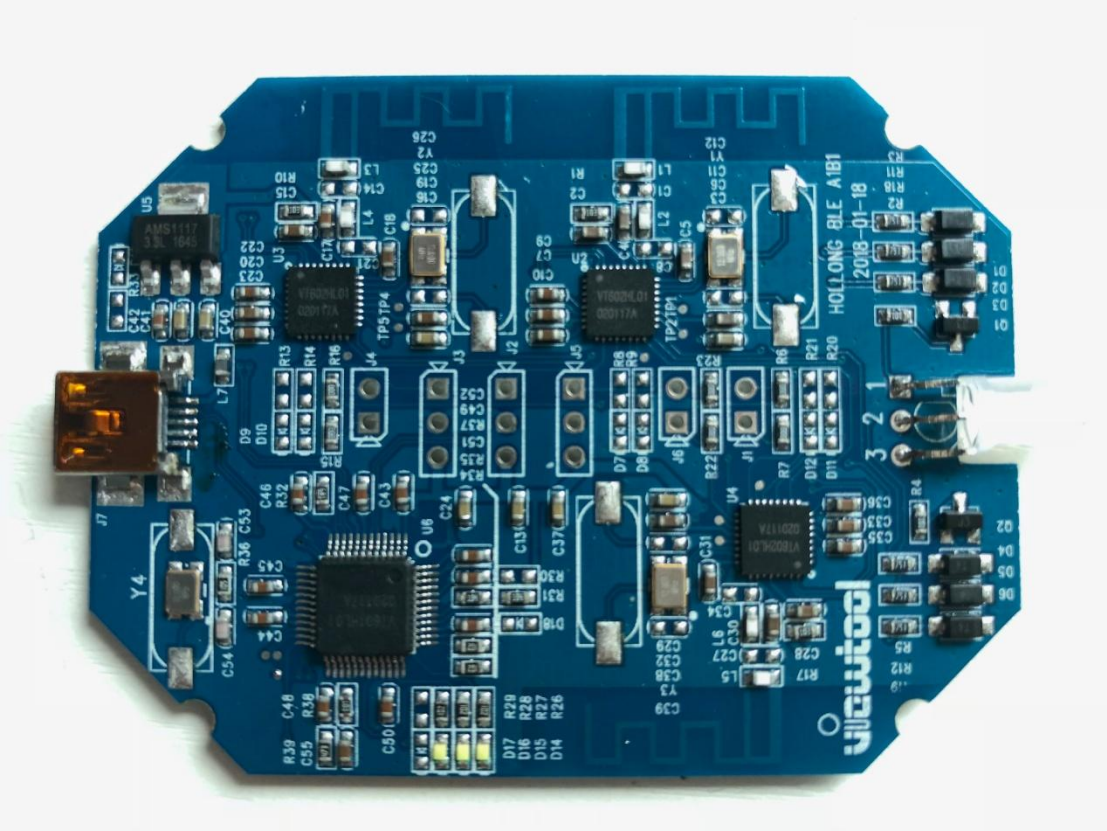
Version	Date	Note
v1.0	2018-12-21	Initial

1. Sniffer Introductoin

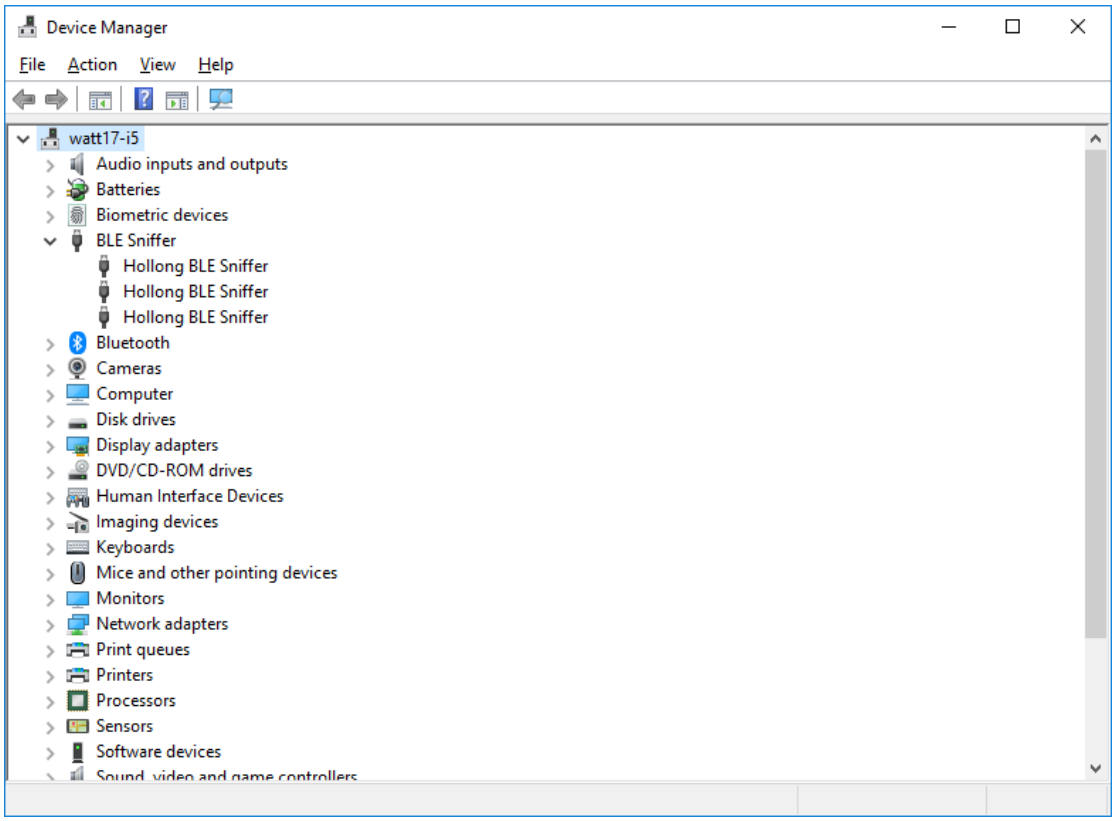
1.1 Hardware

Hollong BLE Sniffer Hardware built with:

1. 3 BLE Chips: capable to capture 3 channels (37,38,39) BLE RF signal simultaneously to guarantee capture all advertising packets (connection packet could be at any one of the three channels) as well as the data after connection setup between master and slave BLE devices.
2. 1 USB chip: to communicate PC with faster speed to transfer all captured data to PC;



3. 3 LEDs indicate the 3 BLE chip working status: any one of them not lit indicates the associated chip is failed (not working);
4. Device driver: after downloaded and installed Hollong BLE Sniffer software, chcek device manager for 3 chips drivers are ready or not (see picture below). If there are 3 yellow note of admiration then means driver has not been installed correctly, need to reinstall driver or “disable digital driver signature enforement” when reboot PC.



Above “Hollong BLE Sniffer” under “BLE Sniffer” indicates the drivers for 3 chips.

5. Connection to PC: by one mini port USB cable;

1.2 Software architecture

1. Hollong open source code takes QT programming language (C++) and QT Creator as IDE environment.

2. It comes with three files for developer:

ble_sniffer_driver.dll: windows DLL file; (linux and Mac please download Hollong BLE Sniffer Linux and Mac version to get corresponding dll files for Linux and Mac);

ble_sniffer_driver.h: DLL header file

ble_sniffer_driver.lib: lib file for static library link usage purpose.

3. Hollong BLE Sniffer software support multiple sniffers simultaneously working in one PC, but need to notice sales about this requirement so could be shipped accordingly.

1.3 Notes for sniffer usage

1. Put sniffer in the middle of BLE master and slave devices;

2. If data missing under capturing, put sniffer closer to master and slave devices (for example: master device <---25cm---> sniffer <---25cm--->slave device.

3.If still missing data, put master, slave devices and sniffer into clean RF environment to eliminate the interference from other devices (like WIFI, Bluetooth, BLE, etc, all 2.4Ghz

radio sources).

2. Hollong BLE Sniffer API function introduction

2.1 API functions

2.1.1. scan_dev()

int scan_dev(uint64_t *pserial)

Parameter: *pserial -> device serial number. Hollong BLE Sniffer support multiple sniffers working simultaneously in one PC so that require different serial number to distinguish sniffer hardware;

Return: >0: sniffer device descriptor,

<=0: failed.

Example:

```
/*************************************************************
   int ret;
        ret = scan_dev(NULL);
        if(ret <= 0){
            qDebug()<<"scan_dev() : "<<ret;
            QMessageBox::warning(this,"Warning","No device connected!");
            return;
```

```
        }
    ************************************************************/
```

2.1.2. open_dev()

int open_dev(int dev_index)

Parameter: dev_index, for multiple sniffer, specify which one to be used in this application

Return: 3: open device success, other values: failed to open devices

Examples:

```
/************************************************************
   Int DevIndex = 0;
            ret = open_dev(DevIndex);
            if(ret != 3){
   QMessageBox::warning(this,"Warning","Open device faild!");
                return;
        }
************************************************************/
```

2.1.3. close_dev()

int close_dev(int dev_index)

Parameter: dev_index, for multiple sniffer, specify which one to be used;

Ret: the device handle of closed device

Examples:

```
/*************************************************************
   Int DevIndex = 0;
            ret = close_dev (DevIndex);
            if(ret != 3){
   QMessageBox::warning(this,"Warning","Close  device faild!");
                return;
          }
*************************************************************/
```

2.1.4. get_raw_data()

int get_raw_data(int dev_index,bsniffer_get_raw_data_cb_fn

get_raw_data_cb_fn,int timeout)

Parameter:

5.int start_get_data(int dev_index,bsniffer_get_raw_data_cb_fn callback)

6.int stop_get_data(int dev_index)

2.1.5. start_get_data()

int start_get_data(int dev_index,bsniffer_get_raw_data_cb_fn callback)

Parameters:

Dev_index: specify which sniffer to be used;

Callback: the callback function to be filled in so dll will call this function to transfer captured data;

Ret:

-1: failed

0: success

Examples:

/***

void WINAPI get_raw_data_cb(int dev_index,int channel,unsigned char *data,int data_len)

```
    {
        printf_raw_data(channel,data,data_len);
        ...
    }

bool Hollong_sniffer_start_capture(bool state)
    {
            start_get_data(DevIndex,get_raw_data_cb);
    }
**************************************************************/
```

2.1.6.stop_get_data()

int stop_get_data(int dev_index)

Parameter: dev_index, specify sniffer device

Ret： 0： success

Example :

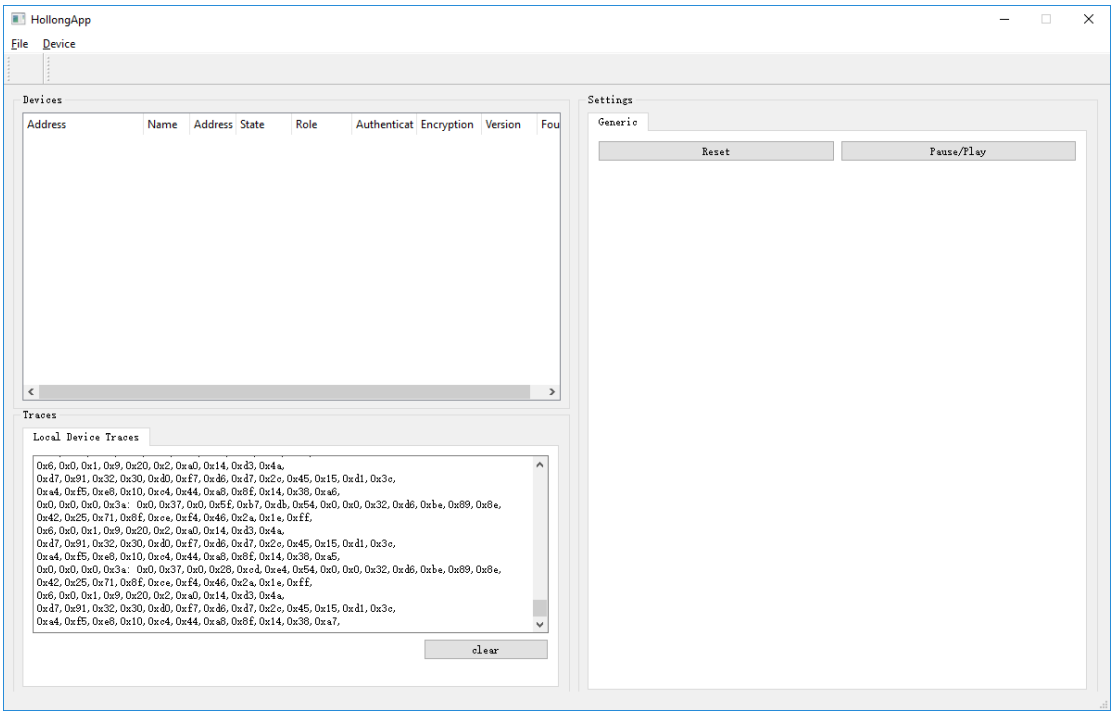
```
/**************************************************************
bool start_stop_get_data(bool start_stop_flag)
    {
        int ret;
        ifstart_stop_flag)
        {
```

```
            ret = open_dev(DevIndex);
            if(ret != 3){

QMessageBox::warning(this,"Warning","Open device faild!");
                return false;
            }
            start_get_data(DevIndex,get_raw_data_cb);
            qDebug()<<"start get data";
            return true;
        }else{
            stop_get_data(DevIndex);
            qDebug()<<"stop get data";
            return true;
        }
    }
*************************************************************/
```

3. Hollong Open Source

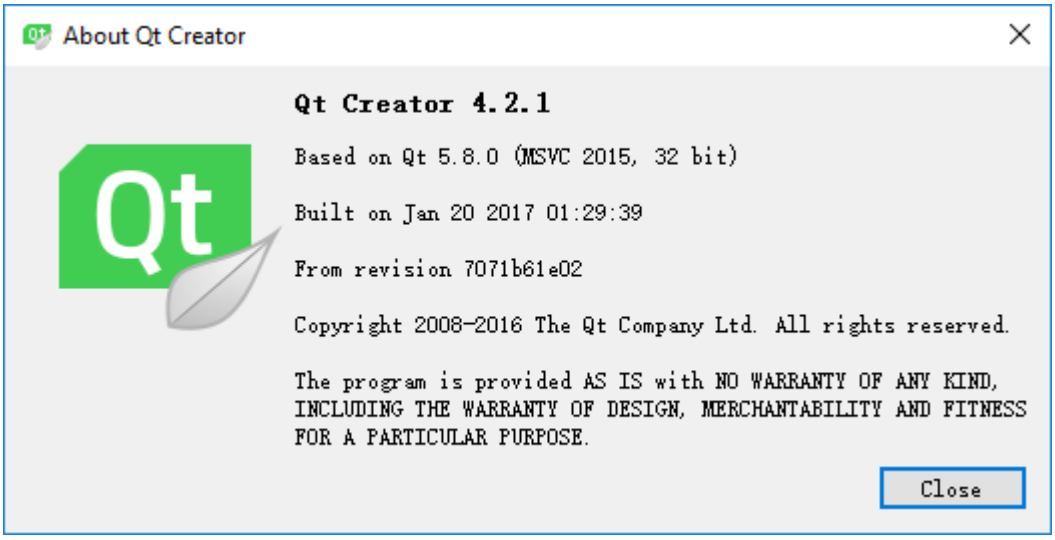
For the convenience of Hollong API usage, the Hollong open source project “Hollong” has been introduced.



It comes with following features:

1. Capture BLE data with Hollong BLE Sniffer;
2. Save captured data by menu “File->Save”;
3. Button “Reset”: start BLE data capturing;
4. Button “Pause/Play”: Pause and continue to capture data;
5. Button “Clear”: clear data

Compiler environment:



4. Technical Support And Services

- Lifetime technical support, one year free repairing or replacement (in case)；
- Driver, Software, User Manual, Application source code could be downloaded at:www.viewtool.com ;
- BBS： www.viewtool.com/bbs (chinese)
- BBS： www.viewtool.com/forum (English)
- Technical support email： fae@viewtool.com
- Sales email： sales@viewtool.com
- Skype: wattsong
- Wechat: wattsong
- QQ: 350207496

Company official wechat account: