LC-1000U

2.4G High-Performance Transparent Wireless USB to UART Device

1 Introduction

The LC-1000U is an application of LC-1000. It is composed of a LC-1000, a controller MCU and some peripheral circuit. The controller MCU enumerates the LC-1000U as a USB device, and realizes the transformation from USB to UART interface which is transparent to the user. So using the LC-1000U, user can send/receive data directly through USB interface conveniently without considering the wireless transmission process. Meanwhile, two large FIFO are allocated in LC-1000U for data transmission, in combination with LC-1000's data flow control mechanism, the high speed and reliable data transmission can be guaranteed.

There are two applications for the LC-1000U: One is a CDC device application and the other is a HID device application. User can download any one of them to LC-1000U flexibly using the PC program "RF2410U Loader.exe". For the CDC device application, a VCP device implemented on windows system, user can access the LC-1000U in the same way as access the standard serial port. And for the HID device, a HID interface device implemented, there is no driver needs on most of Windows systems. User can access the LC-1000U just by sending the HID command packets or calling the functions of LC1000U_HID.dll (provided by INHAOS for HID access).

2 Features

- USB 2.4G wireless data transmission device
- Full duplex transparent data transmission
- Configurable baud rate, range: 2400bps to 57600bps (Only for CDC Application)
- Frequency range: 2400-2483.5 MHz ISM
- 4 bytes RF TX/RX configurable address
- Maximum duplex RF air data rate reaches 38.4kbps
- Transmission distance more than 60 meters
- Adopt C8051F321 MCU, 25MIPS, 16KB Flash, 1280B RAM

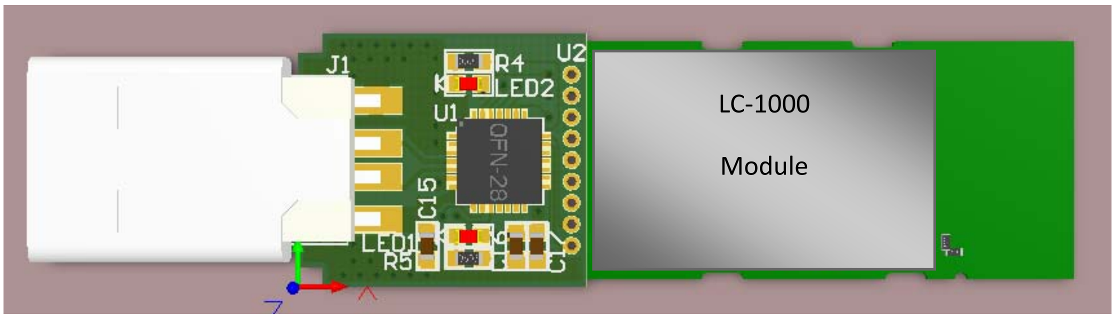
- Built-in Bootloader, you can download the firmware directly via the USB
- Built-in 4 bytes UID (Unique ID) for each units

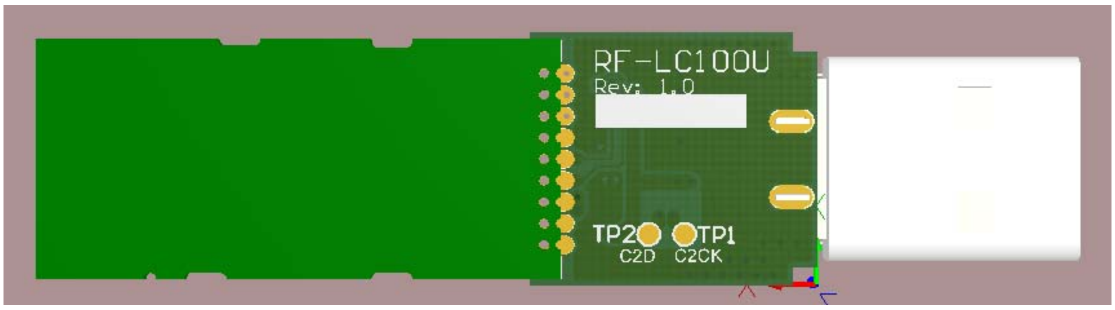
3 Typical Application

- Wireless audio transmission
- Handheld device
- Wireless monitoring and control System
- Remote controlled toys
- Short distance wireless data transmission
- 1 to N wireless data acquisition

4 Hardware description

4.1 Product Demonstration





4.2 Hardware Circuit Diagram

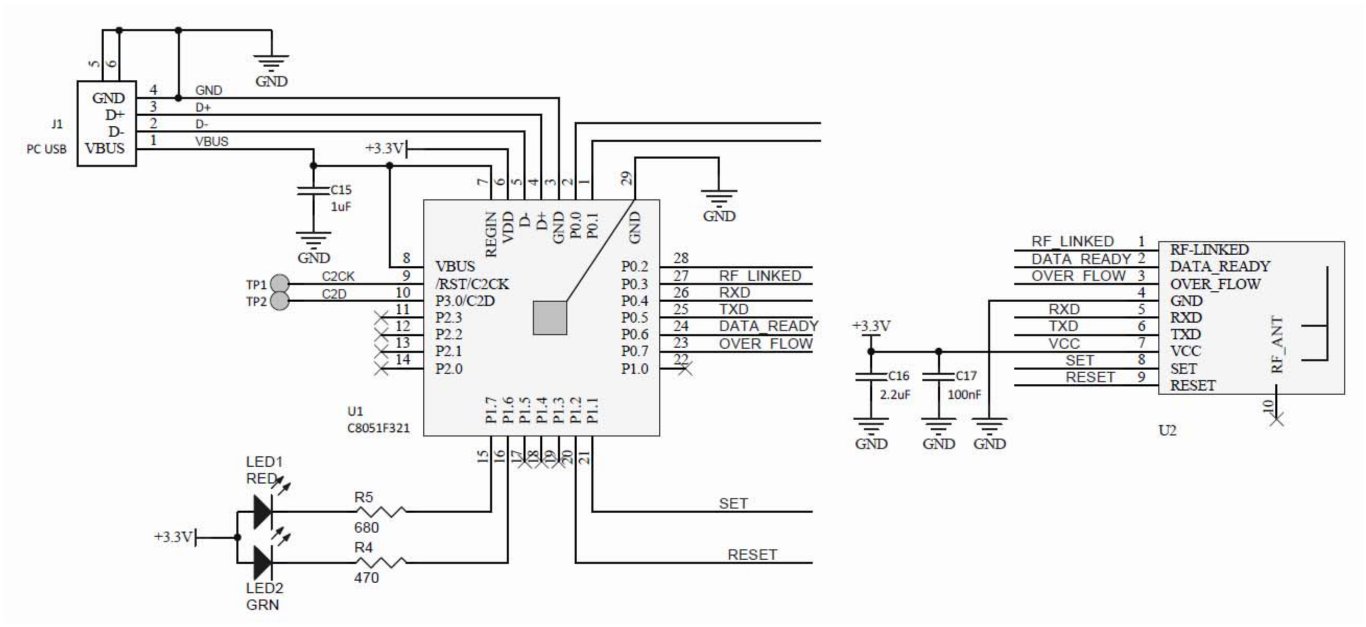


Figure 1 LC-1000U Circuit Diagram

5 LC-1000 Brief Introduction

LC-1000 is a 2.4G transparent low power consumption wireless UART module. It supports full duplex transparent data transmit, and the baud rate range is: 2400bps to 57600bps. Its maximum duplex RF air data rate can reach 38.4kbps, and the transmission distance more than 60 meters. Further mode, A PSM Mode is supplied by the LC-1000, which can significantly reduce the power consumption of application system.

For more details about the LC-1000, please reference the datasheet "**UM-LC-1000-V10-EN Wireless UART Module** ", which can be downloaded from our website: http://www.inhaos.com.

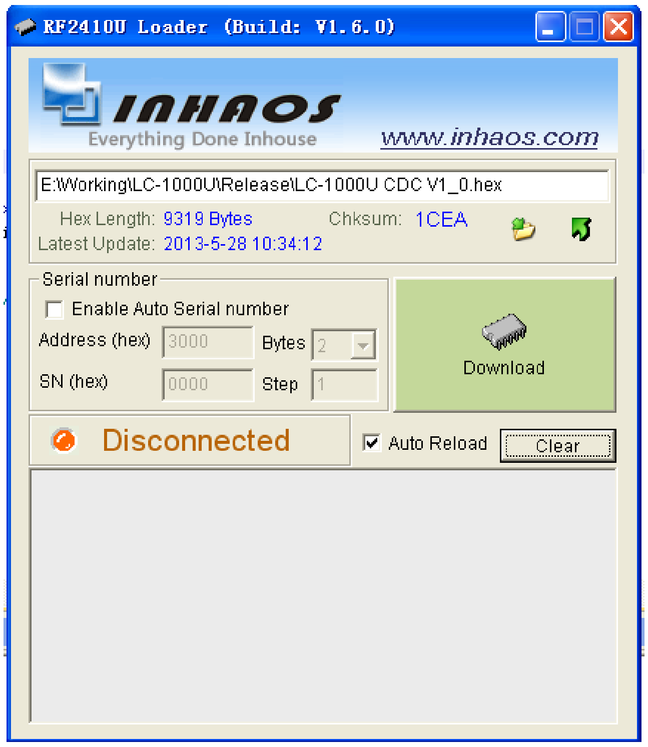
6 Downloading the Application FW to LC-1000U

A bootloader firmware already programmed into the LC-1000U by INHAOS. And we provide two application HEX files for the LC-1000U: LC-1000U HID V1_0.hex and LC-1000U CDC V1_0.hex. The LC-1000U HID V1_0.hex is a HID interface device application firmware, and LC-1000U CDC V1_0.hex is a CDC interface device application firmware. User can reload anyone of them to LC-1000U at will, after get any LC-1000U device.

By default, LC-1000U CDC V1_0.hex is pre-loaded into the devices by factory.

The steps for downloading the application firmware:

1) Open the program "RF2410U Loader.exe"


RF2410U Loader (Build: V1.6.0)

INHAOS Everything Done Inhouse www.inhaos.com

E:\Working\LC-1000U\Release\LC-1000U CDC V1_0.hex

Hex Length: 9319 Bytes Chksum: 1CEA
Latest Update: 2013-5-28 10:34:12

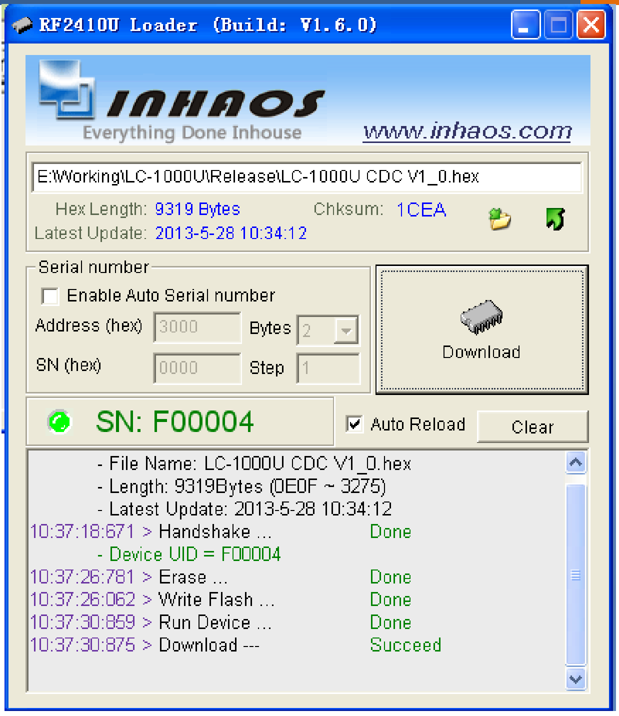
Serial number
☐ Enable Auto Serial number
Address (hex) 3000 Bytes 2
SN (hex) 0000 Step 1

Download

Disconnected ☑ Auto Reload Clear

Figure 2 “RF2410U Loader” program

2) Specify the hex file path by click the button.
3) Click the “Download” button firstly, and then plug the LC-1000U device into any USB port. After the program detected the device, the downloading progress will start.
4) Waiting for downloading complete.



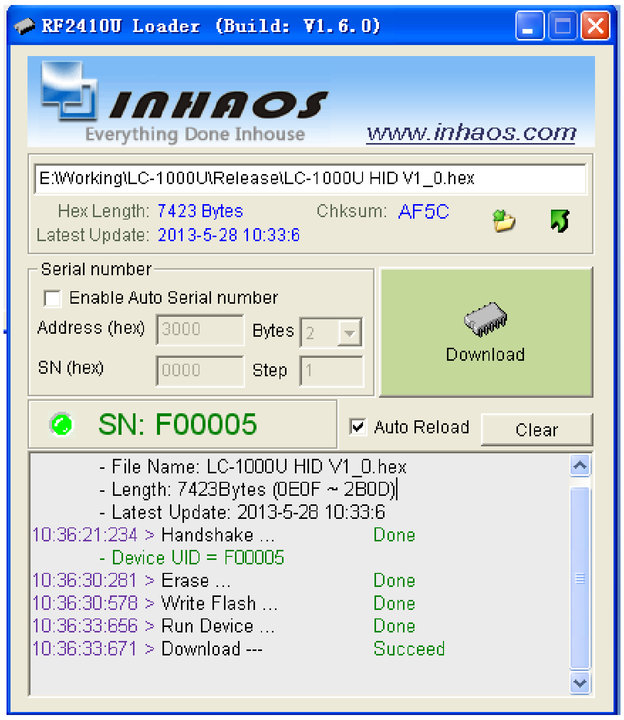
```
E:\Working\LC-1000U\Release\LC-1000U CDC V1_0.hex
Hex Length: 9319 Bytes        Chksum: 1CEA
Latest Update: 2013-5-28 10:34:12

[ ] Enable Auto Serial number
Address (hex) 3000    Bytes 2
SN (hex)      0000    Step  1

SN: F00004    [✓] Auto Reload

        - File Name: LC-1000U CDC V1_0.hex
        - Length: 9319Bytes (0E0F ~ 3275)
        - Latest Update: 2013-5-28 10:34:12
10:37:18:671 > Handshake ...          Done
        - Device UID = F00004
10:37:26:781 > Erase ...              Done
10:37:26:062 > Write Flash ...        Done
10:37:30:859 > Run Device ...         Done
10:37:30:875 > Download ---           Succeed
```

Figure 3 Download example for LC-1000U CDC V1_0.hex



```
E:\Working\LC-1000U\Release\LC-1000U HID V1_0.hex
Hex Length: 7423 Bytes        Chksum: AF5C
Latest Update: 2013-5-28 10:33:6

[ ] Enable Auto Serial number
Address (hex) 3000    Bytes 2
SN (hex)      0000    Step  1

SN: F00005    [✓] Auto Reload

        - File Name: LC-1000U HID V1_0.hex
        - Length: 7423Bytes (0E0F ~ 2B0D)
        - Latest Update: 2013-5-28 10:33:6
10:36:21:234 > Handshake ...          Done
        - Device UID = F00005
10:36:30:281 > Erase ...              Done
10:36:30:578 > Write Flash ...        Done
10:36:33:656 > Run Device ...         Done
10:36:33:671 > Download ---           Succeed
```

Figure 4 Download example for LC-1000U HID V1_0.hex

7 LC-1000U HID Application Description

7.1 System architectures for HID Application

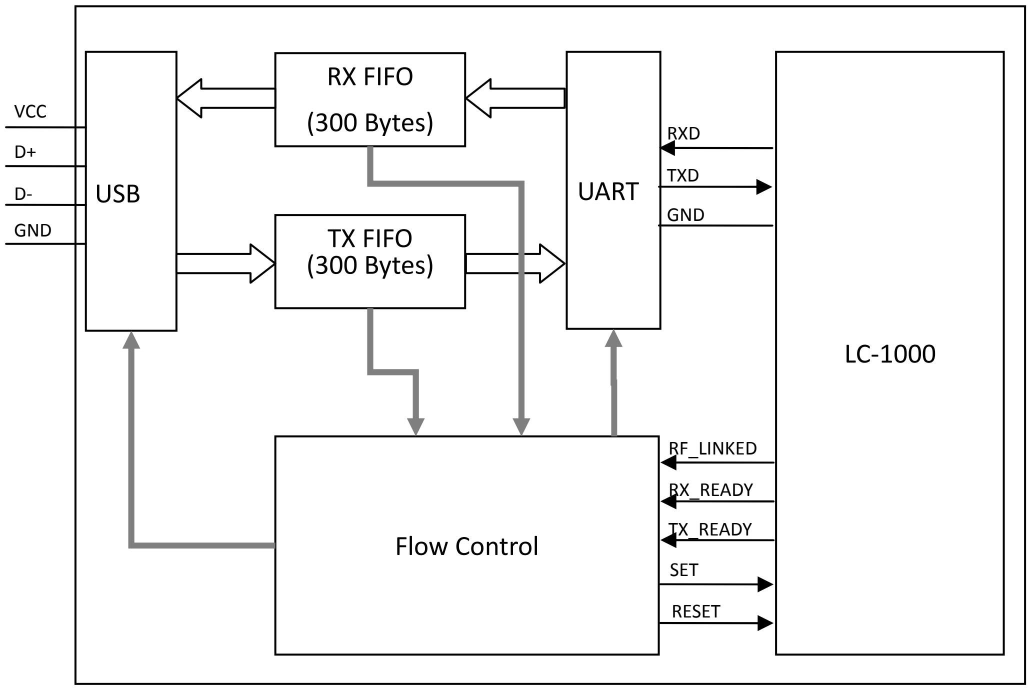


Figure 5 System architectures for HID Application

7.2 HID Application Command Description

7.2.1 HID Packet Format

Name	Checksum	Status	Command	SN_s	SN_r	Length	Parameter
Length	1	1	1	1	1	1	58
Value	0x00~ 0xFF	0x00~ 0xFF	0x00~ 0xFF	0x00~ 0xFF	0x00~ 0xFF	0x00~ 0xFF	0x00~ 0xFF

Table 1 Packet format for HID application

- **Checksum**

The checksum of total packet, it can be calculate by follow formula:

Checksum = NOT (Status + Command + SN_s + SN_r + Length + Parameter)

- **Status**

  The pin states of LC-1000:

  Bit 0, RF_LINKED state

  Bit 1, RX_READY state

  Bit 2, TX_READY state

  Bit 3, SET state

  Bit 4, RESET state

  Bit 5, TX_FIFO_OVERRUN

  Bit 6, RX_FIFO_OVERRUN

- **Command**

  Command field, for details please reference the 7.2.2.

- **SN_s**

  Serial number for the data packet

- **SN_r**

  Serial number for the next packet of transmission another side

- **Length**

  The valid data length of the packet

- **Parameter**

  The data field of the packet

7.2.2 Command List

For simplify, the following command list table only contains the relevant field, and other fields be ignored.

Function	Command	Length	Parameter	Comment
Data Transmission	0XC0	0x00~0x3A	Maximum 58 bytes: Param 0: Data byte 0 Param 1: Data byte 1 …	Specify the packet contains valid user data.

			Param 57: Data byte 57	
Set Configure Mode	0XC1	0x01	Total 1 byte: Param 0: Mode 0x00 busy 0x01 NML Mode 0x02 Configure Mode Param 1 ~ 57: Reserved	Set the LC-1000U into/exit from configure mode. After received this command, LC-1000U start to switching to the specified mode. User must query the device's mode using the 'Get Configure' command to check whether the switch operation completed or not.
Get Configure Mode	0XC2	0x01	Total 1 byte: Param 0: Mode 0x00 busy 0x01 NML Mode 0x02 Configure Mode Param 1 ~ 57: Reserved	Get the LC-1000U's current mode
Purge	0XC3	0x01	Total 1 byte: Param 0: Mode 0x01 Input Buffer 0x02 Output Buffer Param 1 ~ 57: Reserved	Purge the LC-1000U's input/output buffer
Reset	0XCB	0x00		Reset the LC-1000. After reset, the LC-1000's baud rate is set to default 57600bps. And the Local address of LC-1000 reset to the UID bytes.

Figure 6 Command List for HID Application

7.2.3 Serial Number Control for data transmission

In order to improve the transmission reliability, a serial number control mechanism is adopted by LC-1000U. Either reading from the LC-1000U or writing to the LC-1000U, the SN_r and SN_s field in the packet must comply with the serial number control mechanism. Otherwise, the data maybe duplication or discard by LC-1000U. Figure 7 shows the details of serial number control mechanism.

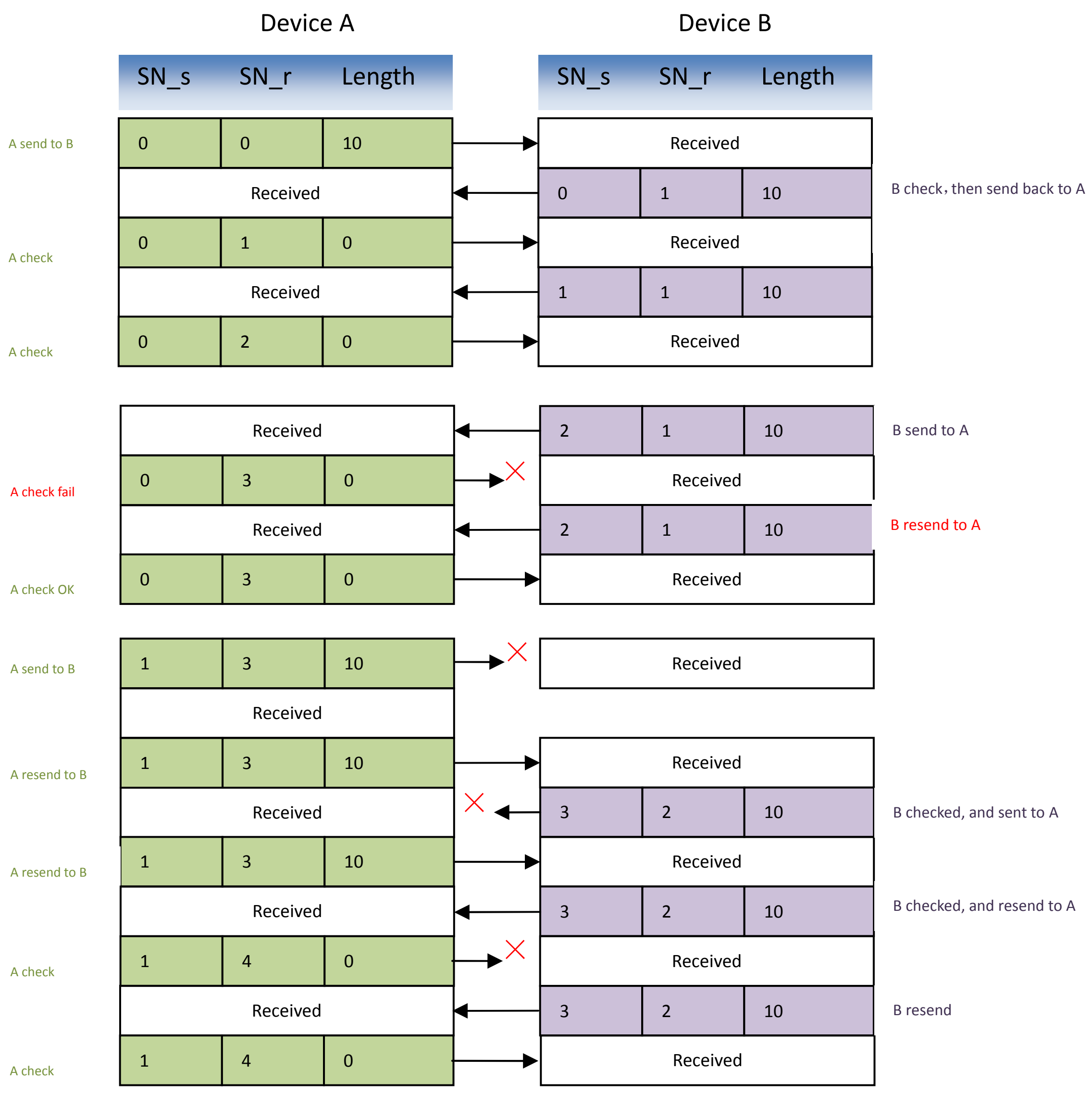


Figure 7 Serial number control mechanism diagram

7.3 HID Application DLL Description

For the convenience of the user, HID application supplies a common HID DLL: LC1000U_HID.dll, which designed using Visual studio C++ environment. It can be used on most windows platforms (such as Windows XP, Windows 2000 Professional, Windows NT, WIN7 and later). And it also supports vast majority of high-level languages environment which supports DLL invoke, such as VC, VB, or VS.NET.

7.3.1 DLL Interface functions

The LC1000U_HID.dll contains following interface functions:

- **LC1000U_Open**
  Open the LC-1000U HID Device
- **LC1000U_Close**
  Close the LC-1000U HID Device
- **LC1000U_IsOpen**
  Check whether the LC-1000U HID Device is opened or not
- **LC1000U_GetAllDevice**
  Search for all the HID devices on the system
- **LC1000U_WriteDataToDevice**
  Write data to device through HID interface
- **LC1000U_ReceiveDataFromDevice**
  Read the received data in the receive buffer
- **LC1000U_GetBytesReceived**
  Get the data length received by DLL
- **LC1000U_SetDeviceIntoConfigMode**
  Set the LC-1000 to configure mode
- **LC1000U_SetDeviceExitFromConfigMode**
  Set the LC-1000 exit from configure mode
- **LC1000U_DiscardOutBuffer**
  Discard the output buffer, after this command operated, all the data in the output buffer of the DLL and the output buffer of device will be cleared.
- **LC1000U_DiscardInBuffer**
  Discard the input buffer, after this command operated, all the data in the input buffer of the DLL and the input buffer of device will be cleared.
- **LC1000U_SetSleepTime**
  Set the sleep time of LC-1000
- **LC1000U_SetTXAddr**
  Set the TX address of LC-1000
- **LC1000U_SetLocalAddr**
  Set the local address of LC-1000
- **LC1000U_SetWorkMode**
  Set the work mode of LC-1000
- **LC1000U_SetRFPower**
  Set the RF power of LC-1000
- **LC1000U_SetCarrierOut**
  Set the LC-1000 output the carrier wave (only for test)
- **LC1000U_GetSleepTime**
  Get the sleep time setting of the LC-1000
- **LC1000U_GetTXAddr**
  Get the TX address setting of the LC-1000

- **LC1000U_GetWorkMode**
  Get the work mode setting of the LC-1000
- **LC1000U_GetRFPower**
  Get the RF Power setting of the LC-1000
- **LC1000U_GetLocalAddress**
  Get the local address setting of the LC-1000
- **LC1000U_GetIDN**
  Get the IDN of the LC-1000
- **LC1000U_GetVersion**
  Get the version setting of the LC-1000
- **LC1000U_Reset**
  Reset the LC-1000. After reset, the LC-1000's baud rate is set to default 57600bps. And the Local address of LC-1000 reset to the UID bytes.
- **LC1000U_GetPinState**
  Get the pin state of LC-1000

For more details please reference Appendix – HID DLL interface functions

7.3.2 Using DLL Interface functions in VC

The following steps may guide you to using the DLL interface functions in VC (only one function "LC1000U_Open" loaded and invoked for an example):

- Declare a function pointer type

```
typedef BOOL (*DLL_Open)(char *Serial);
```

- Declare a function pointer

```
DLL_Open pFunc_Open;
```

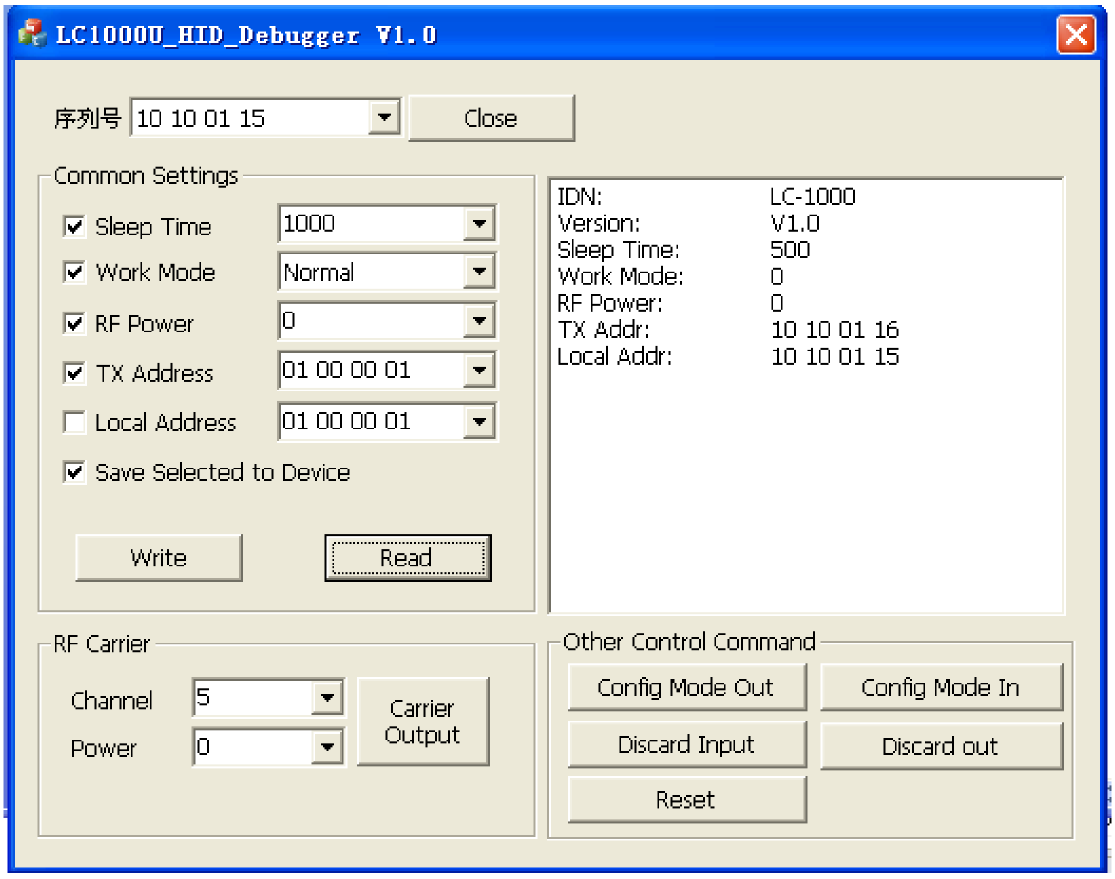
- Dynamic loading the function from the LC1000U_HID.dll

```
HANDLE DllHandle = LoadLibrary("LC1000U_HID.dll");
pFunc_Open = (DLL_Open)GetProcAddress( DllHandle, "LC1000U_Open" );
```

- Invoke the function

```
BOOL Open(char *Serial)
{
	return (*pFunc_Open)(Serial);
}
```

For more detail using process in VC, please reference our VC demo Application: LC1000U_HID_Debugger_VC, which can be downloaded from our website: http://www.inhaos.com. All the common operations for the LC-1000U are gathered in this demo by directly using the LC1000U_HID.dll.



序列号 10 10 01 15

Common Settings
- [✓] Sleep Time: 1000
- [✓] Work Mode: Normal
- [✓] RF Power: 0
- [✓] TX Address: 01 00 00 01
- [ ] Local Address: 01 00 00 01
- [✓] Save Selected to Device

```
IDN:            LC-1000
Version:        V1.0
Sleep Time:     500
Work Mode:      0
RF Power:       0
TX Addr:        10 10 01 16
Local Addr:     10 10 01 15
```

RF Carrier
- Channel: 5
- Power: 0

Figure 8 LC1000U_HID_Debugger V1.0 Demo Program

7.3.3 Using DLL Interface functions in VB.NET

The following steps may guide you to using the DLL interface functions in VB.NET (only one function “LC1000U_Open” loaded and invoked for an example):

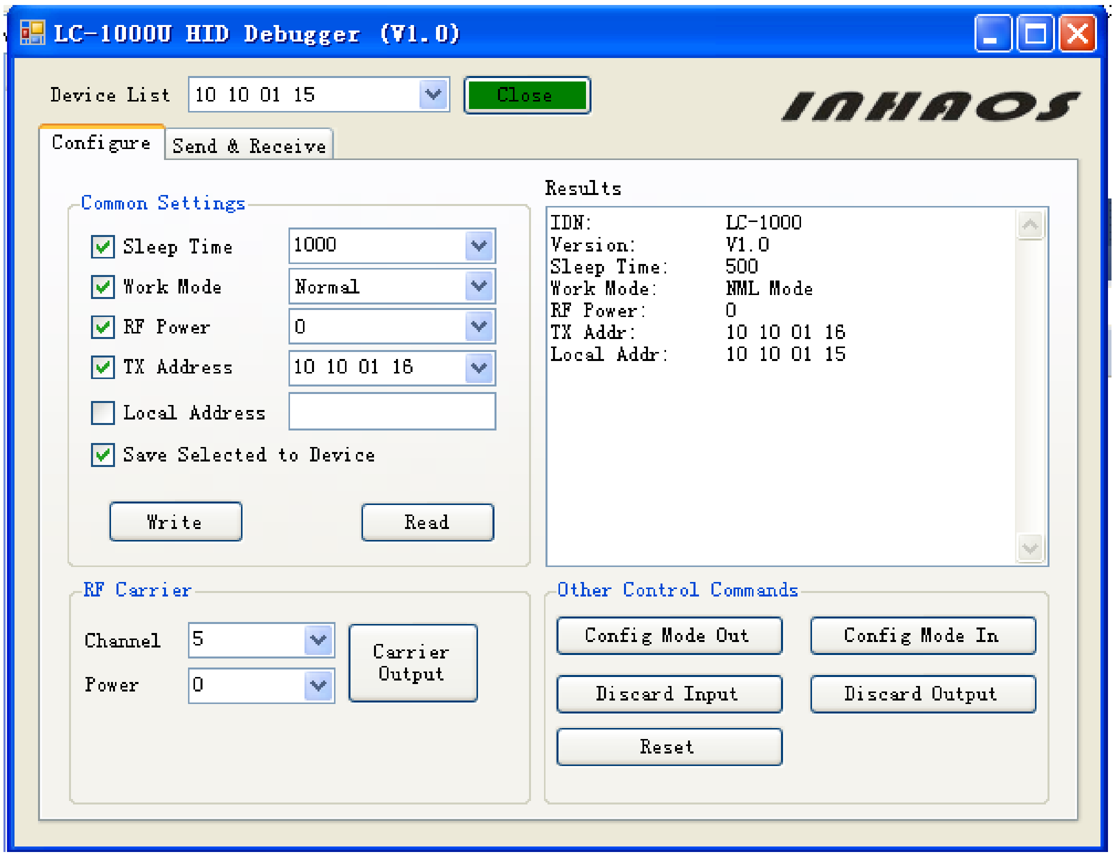
- Declare a function pointer type

```
<DllImport("LC1000U_HID.dll")> Function LC1000U_Open _
        (ByRef Serial As Byte) _
        As UInt16
End Function
```

- Invoke the function

```
Dim NameBytes() As Byte = System.Text.Encoding.ASCII.GetBytes("10 00 00 01")
If LC1000U_Open(NameBytes(0)) > 0 Then
        MsgBox("Open Successful!")
Else
        MsgBox("Open Fail!")
End If
```

For more detail using process in VB.NET, please reference our VB.NET demo Application: LC1000U_HID_Debugger, which can be downloaded from our website: http://www.inhaos.com. All the common operations for the LC-1000U are gathered in this demo by directly using the LC1000U_HID.dll.



Device List: 10 10 01 15

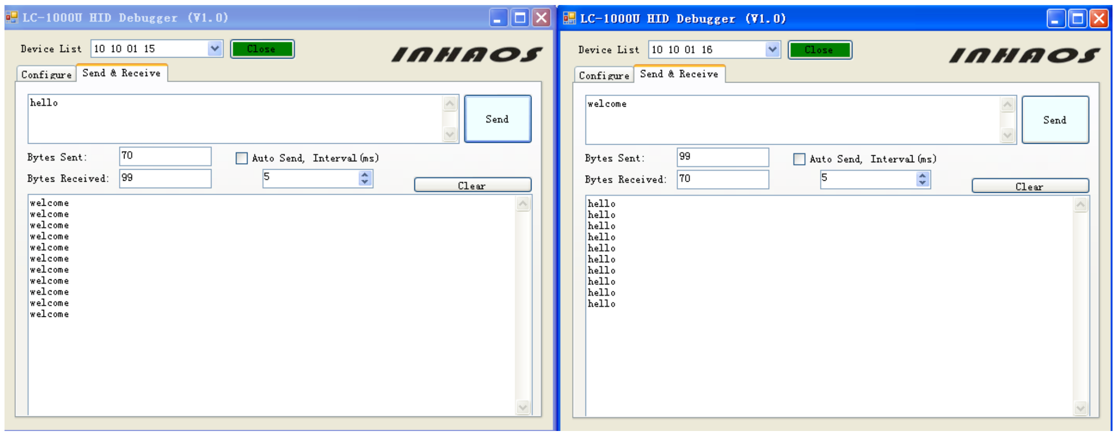
Common Settings:
- [✓] Sleep Time: 1000
- [✓] Work Mode: Normal
- [✓] RF Power: 0
- [✓] TX Address: 10 10 01 16
- [ ] Local Address:
- [✓] Save Selected to Device

RF Carrier: Channel 5, Power 0

Results

```
IDN:            LC-1000
Version:        V1.0
Sleep Time:     500
Work Mode:      NML Mode
RF Power:       0
TX Addr:        10 10 01 16
Local Addr:     10 10 01 15
```

Figure 9 LC-1000U HID Debugger Demo Program



Left window: Device List: 10 10 01 15

hello

Bytes Sent: 70
Bytes Received: 99
[ ] Auto Send, Interval (ms): 5

```
welcome
welcome
welcome
welcome
welcome
welcome
welcome
welcome
welcome
welcome
welcome
```

Right window: Device List: 10 10 01 16

welcome

Bytes Sent: 99
Bytes Received: 70
[ ] Auto Send, Interval (ms): 5

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

Figure 10 Data Transmission between Two HID Devices Example

7.4 Using the LC-1000U with HID Application

7.4.1 Get the device list

Before open the device, you must get the USB serial number firstly. Certainly this step can be skipped if you already know the USB serial number of USB HID Device. According call the LC1000U_GetAllDevice() function, a device list can be returned (Please reference the 30, for details).

```
Private Sub GetDeviceList()
    Dim m_DevList(100) As Byte
    Dim m_DevString As String
    Dim m_DevSplit() As String

    'Clear the Device List
    comboDeviceList.Items.Clear()

    'Get the device list
    If LC1000U_GetAllDevice(m_DevList(0), 100) > 0 Then
        m_DevString = System.Text.Encoding.ASCII.GetChars(m_DevList)
        m_DevSplit = Split(m_DevString, ",")

        'Get each Item in the return list and insert into combo List
        For i As Integer = 0 To m_DevSplit.Length - 1
            If m_DevSplit(i).Length > 0 Then
                comboDeviceList.Items.Add(m_DevSplit(i))
            End If
        Next
    End If
End Sub
```

7.4.2 Open the Device

After got the specify USB serial number, the device can be open by call the function LC1000U_Open().

```
Private Function OpenDevice() As Boolean
    Dim NameBytes() As Byte = System.Text.Encoding.ASCII.GetBytes("10 00 00 01")
    If LC1000U_Open(NameBytes(0)) > 0 Then              'Open the device
        Return True
    Else
        Return False
    End If
End Function
```

7.4.3 Check the state of LC-1000

Before write data to LC-1000U, you'd better check the state of LC-1000. For the data will only send out by LC-1000 after it connected with another side of the transmission. Otherwise, the data wrote in stored in the output buffer and cannot be transmit. To check the state of LC-1000, you can call the function LC1000U_GetPinState().

```
Private Function CheckConnectState() As Boolean
    Dim state As Byte = LC1000U_GetPinState()

    If state And &H1 Then
        MsgBox("Device Connected.")
        Return True
    Else
        MsgBox("Device Disconnected.")
        Return False
    End If
End Function
```

7.4.4 Configure the Device

Before writing configure to device or reading configure data from device. The LC1000U_SetDeviceIntoConfigMode() function must be called to set the LC-1000 into configure Mode, Otherwise the data wrote in will be treated as a user transparent data. And after the configure operation completed, the LC1000U_SetDeviceExitFromConfigMode() function must be called to let the LC-1000 exit the configure Mode and back to the normal Mode. Otherwise the user transparent data will be treated as the configure data.

Write configure data to LC-1000 example code:

```
Private Function WriteConfigToLC1000() As Boolean
    Dim Result As Boolean = False
    Dim ResultText As String = ""
    Dim issave As Boolean = False

    'Set Device Into Config Mode
    If LC1000U_SetDeviceIntoConfigMode() <= 0 Then
        MsgBox("Set Device Into Config Mode Error!")
        Return False
    End If

    'Set Sleep Time = 1000
    If LC1000U_SetSleepTime(1000, issave) <= 0 Then
        MsgBox("Set Sleep Time Error!")
        Return False
    End If
```

```
'Set Work Mode
If LC1000U_SetWorkMode(0, issave) <= 0 Then
    MsgBox("Set Work Mode Error!")
    Return False
End If

'Set RF Power
If LC1000U_SetRFPower(0, issave) <= 0 Then
    MsgBox("Set RF Power Error!")
    Return False
End If

'Set TX Address
Dim TXAddr(3) As Byte          'TX Address Length = 4
Dim strTXAddr() As String = Split("10 00 00 01", " ")
TXAddr(0) = "&h" & strTXAddr(3)
TXAddr(1) = "&h" & strTXAddr(2)
TXAddr(2) = "&h" & strTXAddr(1)
TXAddr(3) = "&h" & strTXAddr(0)
If LC1000U_SetTXAddr(TXAddr(0), issave) <= 0 Then
    MsgBox("Set TX Address Error!")
    Return False
End If

'Set Local Address
Dim LocAddr(3) As Byte          'Local Address Length = 4
Dim strLocAddr() As String = Split("10 00 00 02", " ")
LocAddr(0) = "&h" & strLocAddr(3)
LocAddr(1) = "&h" & strLocAddr(2)
LocAddr(2) = "&h" & strLocAddr(1)
LocAddr(3) = "&h" & strLocAddr(0)

If LC1000U_SetLocalAddr(LocAddr(0), issave) <= 0 Then
    MsgBox("Set Local Address Error!")
    Return False
End If

'Set Device Exit Config Mode
If LC1000U_SetDeviceExitFromConfigMode() <= 0 Then
    MsgBox("Set Device Into Config Mode Error!")
    Return False
```

```
        End If

        Return True
    End Function
```

Read configure data from LC-1000 example:

```
    Private Function ReadConfigFromLC1000() As Boolean
        'Set Device Into Config Mode
        If LC1000U_SetDeviceIntoConfigMode() <= 0 Then
            MsgBox("Set Device Into Config Mode Error!")
            Return False
        End If

        'Read IDN
        Dim IDN_Bytes(6) As Byte        'IDN length = 7
        Dim strIDN As String
        If LC1000U_GetIDN(IDN_Bytes(0)) > 0 Then
            strIDN = System.Text.Encoding.ASCII.GetChars(IDN_Bytes)
        Else
            Return False
        End If

        'Read Version
        Dim Version_Bytes(1) As Byte        'Version Length = 2
        Dim strVersion As String
        If LC1000U_GetVersion(Version_Bytes(0)) > 0 Then
            strVersion = "V" & Version_Bytes(0) & "." & Version_Bytes(1)
        Else
            Return False
        End If

        'Read Sleep Time
        Dim SleepTime As UInt16 = 0
        If LC1000U_GetSleepTime(SleepTime) > 0 Then
        Else
            Return False
        End If

        'Read Work Mode
        Dim WorkMode As Byte = 0
        If LC1000U_GetWorkMode(WorkMode) > 0 Then
        Else
            Return False
        End If

        'Read RF Power
        Dim RFPower As Byte = 0
        If LC1000U_GetRFPower(RFPower) > 0 Then
        Else
            Return False
        End If

        'Read TX Address
        Dim TXAddr(3) As Byte           'TX Address Length = 4
        dim strTXAddr as String =""
        If LC1000U_GetTXAddr(TXAddr(0)) > 0 Then
            strTXAddr = vbTab & TXAddr(3).ToString("X2") & " " & _
                TXAddr(2).ToString("X2") & " " & _
                TXAddr(1).ToString("X2") & " " & _
                TXAddr(0).ToString("X2")
        Else
            Return False
        End If

        'Read Local Address
        Dim LocAddr(3) As Byte          'Local Address Length = 4
        Dim strLocalAddr As String
        If LC1000U_GetLocalAddress(LocAddr(0)) > 0 Then
            strLocalAddr = vbTab & LocAddr(3).ToString("X2") & " " & _
                LocAddr(2).ToString("X2") & " " & _
                LocAddr(1).ToString("X2") & " " & _
                LocAddr(0).ToString("X2")
        Else
            Return False
        End If

        'Set Device Exit Config Mode
        If LC1000U_SetDeviceExitFromConfigMode() <= 0 Then
            MsgBox("Set Device Into Config Mode Error!")
            Return False
        End If

        Return True
    End Function
```

7.4.5 Write data to the device

For writing data to the device, you just need to call the function LC1000U_WriteDataToDevice(), then the data be write into the LC-1000U's output buffer immediately, after that the LC-1000 will transmit them to another side. If the output buffer is full or the empty buffer length is less than the request length of the function, then the request data will be discarded and the function LC1000U_WriteDataToDevice() returns a zero to indicates write operation fail.

```
Private Function SendData() As Boolean
    If tb_SendTxt.Text.Length = 0 Then
        Return False
    End If

    Dim dataSend() As Byte = System.Text.Encoding.ASCII.GetBytes(tb_SendTxt.Text)
    If LC1000U_WriteDataToDevice(dataSend(0), 0, dataSend.Length) > 0 Then
        Count_TX += dataSend.Length
        Return True
    End If

    Return False
End Function
```

7.4.6 Read data from the device

After any data received by the LC-1000, the LC-1000U's controller MCU will fill the received data into the input buffer and waiting for PC to reading. And the buffer maybe overrun (oldest data will be overwrite by the new data), if PC's reading operation delay too long lead to no more buffer can contain the incoming new data. User can call the function LC1000U_GetPinState() to check the input buffer's overrun state.

```
Private Sub DataReceiveProc(ByVal txt As Object)
    Dim strNowRecv As String = ""
    Dim RecvLength As Integer = 0

    g_SendRecvThreadStopEvent.Reset()

    While 1
        If g_ExitEvent.WaitOne(0) = True Then
            Exit While
        End If

        'Read the data from input buffer of the LC-1000U
        RecvLength = LC1000U_ReceiveDataFromDevice(m_RecvBuffer(0), 0,
m_RecvBuffer.Length)
```

```
                If RecvLength > 0 Then
                    strNowRecv = System.Text.Encoding.ASCII.GetString(m_RecvBuffer, 0, RecvLength)
                    UpdateText_RX(strNowRecv)
                End If

                Threading.Thread.Sleep(50)
            End While

            g_SendRecvThreadStopEvent.Set()
        End Sub
```

7.4.7 Reset the LC-1000

In some case, user may feel confused about LC-1000's configure setting. Then function LC1000U_Reset() can be called to reset the LC-1000's setting. After this function called, the LC-1000's baud rate is set to default 57600bps. And the Local address of LC-1000 reset to the UID bytes.

```
    Private Sub ResetLC1000()
        If LC1000U_Reset() > 0 Then
            MsgBox("Reset LC-1000 OK")
        Else
            MsgBox("Reset LC-1000 FAIL")
        End If
    End Sub
```

8 LC-1000U CDC Application Description

8.1 System architecture for CDC Application

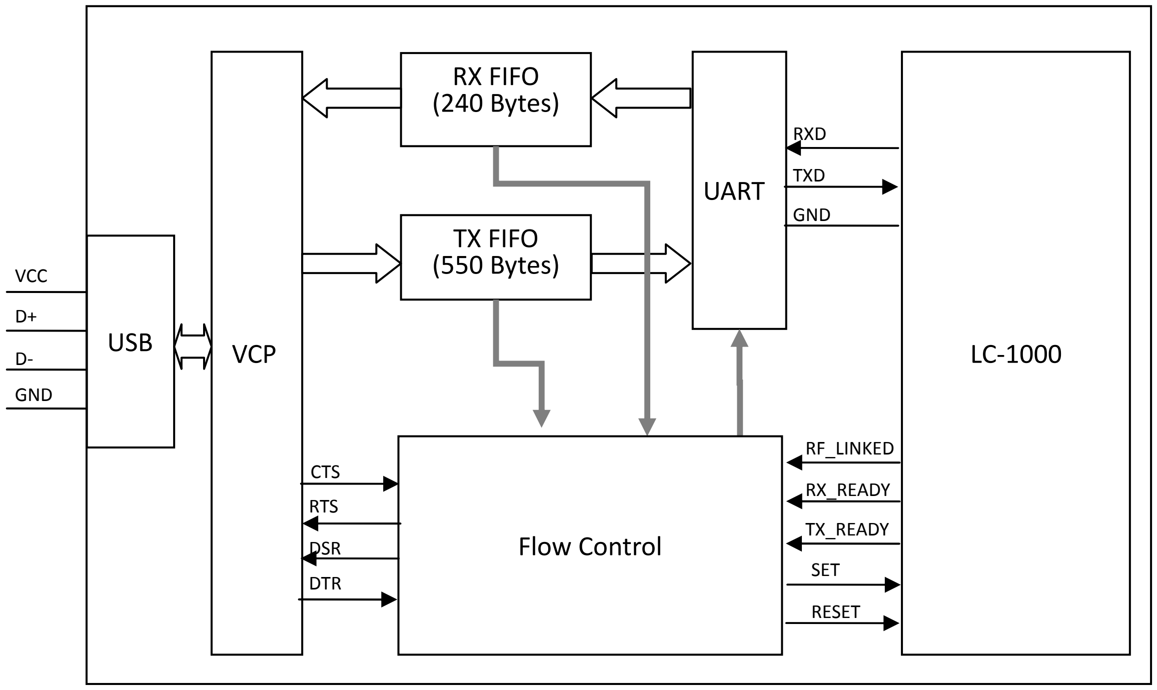


Figure 11 System Architecture for CDC application

8.2 CDC Application Function Description

Unlike the HID Application, user can see a serial port device in the computer's device manage, just the same as a standard serial port, when the CDC application running.

In order to improve transmission reliability, a flow control mechanism is defined by the LC-1000U CDC Device, and some standard pins of VCP are treated as special function pins.

VCP Pin Name	Direction	Functions In LC-1000U	Description
DCD	Input	No use	
RXD	Input	Receive Data	
TXD	Output	Transmit Data	
DTR	Output	Linked to LC-1000's SET pin	PC software can control this pin state to set LC-1000 into/exit configure mode. If DTR = 0, LC-1000 will start switch to configure mode If DTR = 1, LC-1000 will switch back to work mode

GND	GND	GND	
DSR	Input	Linked to LC-1000's TX_READY pin	PC can get the LC-1000's TX_READY pin state by check the DSR state. If DSR = 0, LC-1000 is ready for user data receive; If DTR = 1, LC-1000 is busy, user data cannot be received
RTS	Input	Request To Send	When the LC-1000's output buffer almost full, the RTS is set to 1; When the LC-1000's output buffer has enough data space the RTS will set back to 0
CTS	Output	Clear To Send	If the CTS = 1, the LC-1000Uwill stop data report progress; If the CTS = 0, the LC-1000Uwill report data to PC once received any from LC-1000
RI	Input	No use	

Table 2 Special function of USB VCP PINs

8.3 Using the LC-1000U with CDC Application

The method using the LC-1000U with CDC application is much the same with accessing a serial port of PC.

8.3.1 First of all a serial port should be declared

```
Public m_DevicePort As New IO.Ports.SerialPort
```

8.3.2 Open the Device

It is very important to set the DTR to HIGH, after the device's port was opened. When the DTR = 1, the LC-1000 will go into normal work mode, otherwise, it will keep in configure mode. Please reference 8.2 for detail.

```
Private Function OpenDevice(ByVal strPort As String, ByVal baudrate As Integer) As Boolean
    If m_DevicePort.IsOpen = True Then   'If the port is already opened, close it
        m_DevicePort.Close()
    End If

    m_DevicePort.PortName = strPort          'Set the port name
    m_DevicePort.BaudRate = baudrate         'Set the baudrate

    m_DevicePort.Open()                      'Open the port

    'Set the DTR to high (this very import step,
    'for the LC-1000 go into normal work mode)
```

```
        m_DevicePort.DtrEnable = True

        Return m_DevicePort.IsOpen
    End Function
```

8.3.3 Configure the Device

In order to configure the LC-1000, the following steps are suggested:

1) Set LC-1000 into configure mode
    a) Set the DTR to LOW
    b) Waiting for DSR becomes LOW

```
    Public Function SetDeviceIntoConfigMode(Optional ByVal timeout As Integer = 1000) As Boolean
        Dim result As Boolean = False

        'Set DTR to LOW
        m_DevicePort.DtrEnable = False

        'Waiting for DSR becomes LOW
        Dim timeend As Integer = My.Computer.Clock.TickCount + timeout
        Do
            Threading.Thread.Sleep(10)
            Application.DoEvents()
            If m_DevicePort.DsrHolding = False Then
                result = True
                Exit Do
            End If
        Loop Until My.Computer.Clock.TickCount > timeend

        Return result
    End Function
```

2) Write configure commands to LC-1000, or read configuration from LC-1000

```
    'Set the LC-1000's TX Address
    Public Function SetTXAddr(ByVal addr As String, ByVal issave As Boolean) As Boolean
        Dim str() As String = Split(addr, " ")
        If str.Length < 4 Then
            Return False
        End If

        Array.Clear(m_ParamBuffer, 0, m_ParamBuffer.Length)
        m_ParamBuffer(0) = CByte("&H" & str(3))
        m_ParamBuffer(1) = CByte("&H" & str(2))
        m_ParamBuffer(2) = CByte("&H" & str(1))
        m_ParamBuffer(3) = CByte("&H" & str(0))
        Return WriteConfigToDevice(CMD.CMD_SET_TX_ADDR, 4, issave)
    End Function
    'Get the LC-1000's Work mode
    Public Function GetWorkMode(ByRef mode As Integer) As Boolean
        Array.Clear(m_ParamBuffer, 0, m_ParamBuffer.Length)
        If WriteConfigToDevice(CMD.CMD_GET_WORK_MODE, 1, False) = True Then
            mode = m_ParamBuffer(0)
            Return True
        End If
        Return False
    End Function
```

3) Exit the configure mode: by set the DTR to HIGH

```
    Public Sub SetDeviceExitFromConfigMode()
        m_DevicePort.DtrEnable = True
    End Sub
```

8.3.4 Write data to the device

For the data wrote to the device will be received by the VCP at once, so the data writing operation must follow principles:

1) The data length of write must less than the output buffer length of LC-1000U
2) A delay time should be added after any write operation, and the delay time can be calculated using the formula:

Delay time(ms) = 1000 * 10 * length / baudrate

```
    Public Function GetTimeout(ByVal BytesCnt As Integer) As Integer
        Dim TimeoutVal As Double
        TimeoutVal = BytesCnt * 10000 / m_DevicePort.BaudRate
        Return TimeoutVal
    End Function
    Public Sub WriteDataToDevice(ByRef wData() As Byte, ByVal length As Integer)

        m_DevicePort.Write(wData, 0, length)

        'Delay for sending
        DelayMillSecond(GetTimeout(length))
    End Sub
```

8.3.5 Read data from the device

```
    Public Function ReceiveDataFromDevice(ByRef rData() As Byte, ByVal maxLength As Integer) As
```

```
Integer
            Dim rLen As Integer = 0

            rLen = m_DevicePort.BytesToRead
            If rLen > maxLength Then
                  rLen = maxLength
            End If

            rLen = m_DevicePort.Read(rData, 0, rLen)

            Return rLen
      End Function
```

8.3.6 Flow control for data transmission

The flow control mechanism can make the data transmission more reliable.

Under this flow control mechanism:

- For the downstream, the CTS pin will be set to HIGH by LC-1000U, once the output buffer almost full and be set back to LOW when it has enough data space.
- For the upstream, the LC-1000U will report data immediately after received from LC-1000 when the CTS pin keeps LOW, but it will stop data reporting after the CTS HIGH level detected. In this case the new coming data from the LC-1000 will only store into the input buffer. And once the input buffer full, the oldest data in the input buffer will be overwrote.

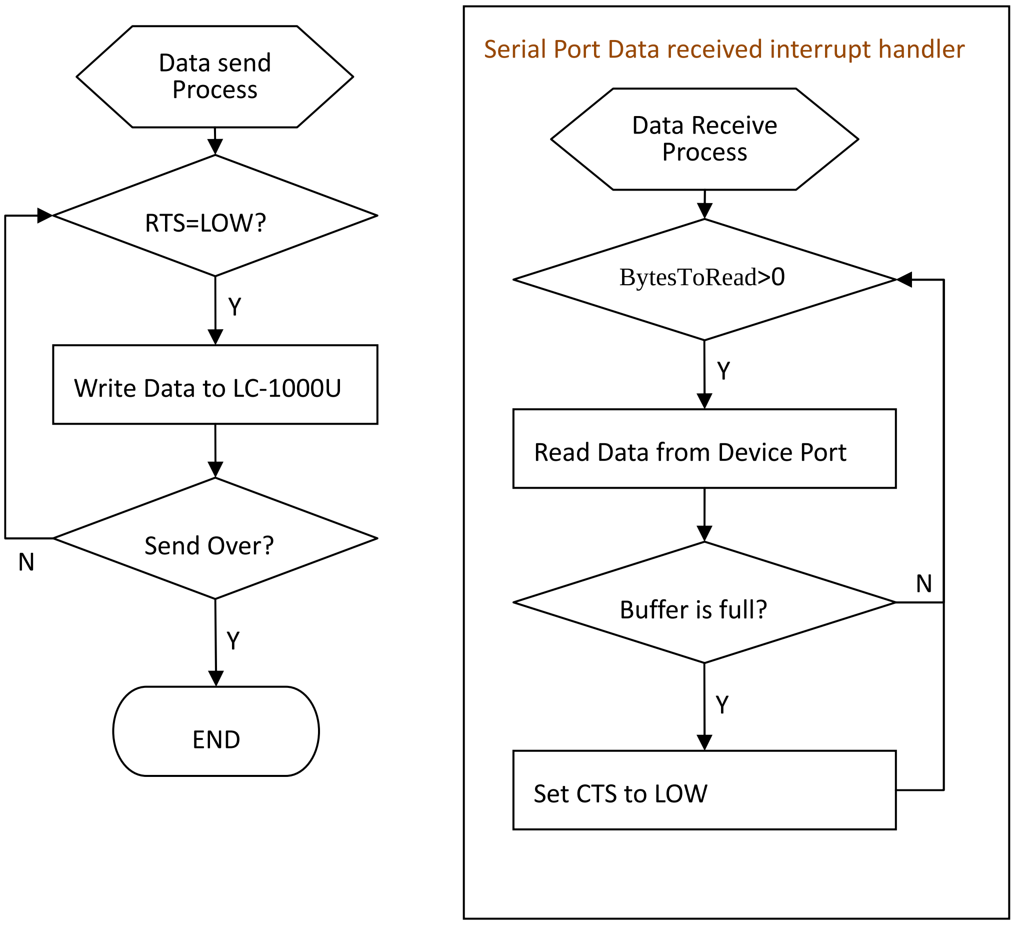


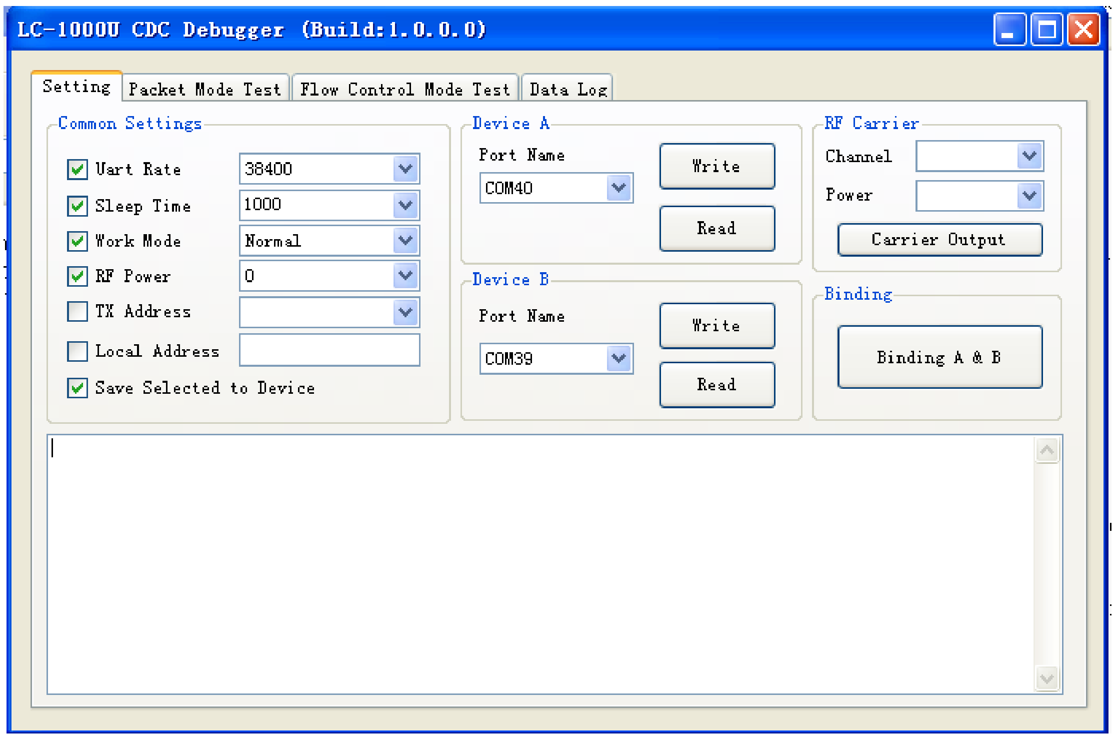
Figure 12 Data transmission with flow control flow chart

8.3.7 The LC-1000U CDC Debugger Software Description

A VB.NET demo program named "LC-1000U CDC Debugger V1.0.exe" is provided by INHAOS for example to use the LC-1000U CDC application. The following functions are implemented by this demo:

- Open device
- Configure Operations for LC-1000
- Set LC-1000 output RF carrier
- Binding two devices A and B, let them can transmit data each other
- Packet Mode transmission Test, under which the flow control mechanism is ignored and it does not guarantee the data reliability

◆ Flow control mode transmission test


LC-1000U CDC Debugger (Build:1.0.0.0)

Setting | Packet Mode Test | Flow Control Mode Test | Data Log

Common Settings

✓	Uart Rate	38400
✓	Sleep Time	1000
✓	Work Mode	Normal
✓	RF Power	0
☐	TX Address	
☐	Local Address	
✓	Save Selected to Device	

Device A
Port Name: COM40

Device B
Port Name: COM39

Figure 13 LC-1000U CDC Debugger Interface

9 Appendix – HID DLL interface functions

- **LC1000U_Open**

Open the LC-1000U HID Device

Prototype:

BOOL LC1000U_Open(char *Serial);

Parameter:

Serial ---- The USB serial number string of HID device, which can be obtained according LC1000U_GetAllDevice().

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_Close**

Close the LC-1000U HID Device.

Prototype:

BOOL LC1000U_Close();

Parameter:

None

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_IsOpen**

Check whether the LC-1000U HID Device is opened or not

Prototype:

BOOL LC1000U_IsOpen();

Parameter:

None

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_GetAllDevice**

Search for all the HID devices on the system

Prototype:

int LC1000U_GetAllDevice(char *rDeviceList, int MaxLen);

Parameter:

rDeviceList ---- The receive buffer for the device USB serial number list of LC-1000U, the item format of the list is like: 10 00 00 01, 10 00 00 02, … ,10 00 00 21, and each item in the list is separated by ‘,’.

MaxLen --- The max length of the receive buffer

Return:

The item count in the rDeviceList

- **LC1000U_WriteDataToDevice**

  Write data to device through HID interface

  Prototype:

  BOOL LC1000U_WriteDataToDevice(UINT8 *wData, int Offset, int Length);

  Parameter:

wData	---- The data buffer which will be write to device
Offset	---- The start position of data buffer
Length	---- The length of wData

  Return:

1	Operation Complete.
0	Operation Fail

- **LC1000U_ReceiveDataFromDevice**

  Read the received data in the receive buffer

  Prototype:

  int LC1000U_ReceiveDataFromDevice(UINT8 *rData, int Offset, int MaxLength);

  Parameter:

rData	---- The receive data buffer
Offset	---- The start position of receive data buffer
MaxLength	---- The max length of the rData

  Return:

  The data length received, that is the valid data length in rData.

- **LC1000U_GetBytesReceived**

  Get the data length received by DLL

  Prototype:

  int LC1000U_GetBytesReceived();

  Parameter:

  None

  Return:

  The valid data length in the receive buffer.

- **LC1000U_GetDeviceConfigMode**

  Get the device's current configure mode status

  Prototype:

  int CLC1000U_GetDeviceConfigMode();

  Parameter:

  None

  Return:

  The current configure mode status:

0x00	Mode switching in progress
0x01	NML Mode
0x02	Configure Mode

- **LC1000U_SetDeviceIntoConfigMode**

Set the LC-1000 to configure mode

Prototype:

BOOL LC1000U_SetDeviceIntoConfigMode();

Parameter:

None

Return:

1	Operation Complete.
0	Operation Fail

- **LC1000U_SetDeviceExitFromConfigMode**

Set the LC-1000 exit from configure mode

Prototype:

BOOL LC1000U_SetDeviceExitFromConfigMode();

Parameter:

None

Return:

1	Operation Complete.
0	Operation Fail

- **LC1000U_DiscardOutBuffer**

Discard the output buffer, after this command operated, all the data in the output buffer of the DLL and the output buffer of device will be cleared.

Prototype:

BOOL LC1000U_DiscardOutBuffer();

Parameter:

None

Return:

1	Operation Complete.
0	Operation Fail

- **LC1000U_DiscardInBuffer**

Discard the input buffer, after this command operated, all the data in the input buffer of the DLL and the input buffer of device will be cleared.

Prototype:

BOOL LC1000U_DiscardInBuffer();

Parameter:

None

Return:

1	Operation Complete.
0	Operation Fail

- **LC1000U_SetSleepTime**

Set the sleep time of LC-1000

Prototype:

BOOL LC1000U_SetSleepTime(UINT16 time, BOOL issave);

Parameter:

time ---- The time value, Value range: 20 to 65535

issave ---- 1 - save the setting to EEPROM; 0 - not save

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_SetTXAddr**

Set the TX address of LC-1000

Prototype:

BOOL LC1000U_SetTXAddr(BYTE *addr, BOOL issave);

Parameter:

addr ---- The address bytes buffer, Length = 4 bytes

issave ---- 1 - save the setting to EEPROM; 0 - not save

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_SetLocalAddr**

Set the local address of LC-1000

Prototype:

BOOL LC1000U_SetLocalAddr(BYTE *addr, BOOL issave);

Parameter:

addr ---- The address bytes buffer, Length = 4 bytes

issave ---- 1 - save the setting to EEPROM; 0 - not save

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_SetWorkMode**

Set the work mode of LC-1000

Prototype:

BOOL LC1000U_SetWorkMode(int mode, BOOL issave);

Parameter:

mode ---- The work mode of LC-1000,

0x00 NML Mode

0x01 PSM Mode

issave ---- 1 - save the setting to EEPROM; 0 - not save

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_SetRFPower**

Set the RF power of LC-1000

Prototype:

BOOL LC1000U_SetRFPower(int power, BOOL issave);

Parameter:

power ---- The power level of LC-1000, value range: 0x00 ~ 0x0F

0x00 – The maximum power output

0x0F – The minimum power output

issave ---- 1 - save the setting to EEPROM; 0 - not save

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_SetCarrierOut**

Set the LC-1000 output the carrier wave (only for test)

Prototype:

BOOL LC1000U_SetCarrierOut(int chn, int power);

Parameter:

chn ---- The frequency channel of RF, value range: 0 ~ 83

power ---- The power level of LC-1000, value range: 0x00 ~ 0x0F

0x00 – The maximum power output

0x0F – The minimum power output

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_GetSleepTime**

Get the sleep time setting of the LC-1000

Prototype:

BOOL LC1000U_GetSleepTime(UINT16 *time);

Parameter:

Time ---- The sleep time, Length = 1

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_GetTXAddr**

Get the TX address setting of the LC-1000

Prototype:

BOOL LC1000U_GetTXAddr(BYTE *addr);

Parameter:

Addr ---- The TX address, Length = 4

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_GetWorkMode**

Get the work mode setting of the LC-1000

Prototype:

BOOL LC1000U_GetWorkMode(BYTE *mode)

Parameter:

Mode --- The work mode of LC-1000, Length = 1

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_GetRFPower**

Get the RF Power setting of the LC-1000

Prototype:

BOOL LC1000U_GetRFPower(BYTE *power);

Parameter:

power --- The power level of LC-1000, Length = 1

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_GetLocalAddress**

Get the local address setting of the LC-1000

Prototype:

BOOL LC1000U_GetLocalAddress(BYTE *addr);

Parameter:

addr --- The local address of LC-1000, Length = 4

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_GetIDN**

Get the IDN of the LC-1000

Prototype:

BOOL LC1000U_GetIDN(BYTE *idn);

Parameter:

idn --- IDN buffer, Length = 7

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_GetVersion**

Get the version setting of the LC-1000

Prototype:

BOOL LC1000U_GetVersion(BYTE *version);

Parameter:

version --- IDN buffer, Length = 2

Return:

1 Operation Complete.

0 Operation Fail

- **LC1000U_Reset**

Reset the LC-1000. After reset, the LC-1000’s baud rate is set to default 57600bps. And the

Local address of LC-1000 reset to the UID bytes.

Prototype:

BOOL LC1000U_Reset();

Parameter:

None

Return:

1	Operation Complete.
0	Operation Fail

- **LC1000U_GetPinState**

Get the pin state of LC-1000

Prototype:

BOOL LC1000U_GetPinState(UINT8 *rPinState);

Parameter:

rPinState ---- Get the Pin State

Return:

1	Operation Complete.
0	Operation Fail

10 Parameters

No.	Parameter	Symbol	Unit	condition	Min	typical	Max	Note
1	Voltage Supply	VDD	V		4.75	5	5.25	
2	Current	IDD	mA	continued transmit RF carrier at 0dBm	TBD	35	TBD	Depending on mode
3	USB rate	FUSB	Mbps			12		USB2.0 full speed
4	RF output frequency	FOP	MHz		2400	---	2483	
5	Data rate	RFSK	bps			30K	38.4K	
6	RF Output power	PRF	dBm		-40		3	
7	Receive sensitivity	RXSENS	dBm	1E-3 BER sensitivity (1Mbps)	---	-90	---	
8	Receive sensitivity	RXSENS	dBm	1E-3 BER sensitivity (2Mbps)	---	-87	---	
9	Storage Temperature	STEMP	°C		-20		+80	
10	Temperature	TEMP	°C		+5		+45	

11 Revision History

Version	**Date**	**Author**	**Description**
V10	2013-5-29	Tony Tan	First released

Declare

Due to technical limitations and the reader's understanding , this document is for reference only. Our company makes no legal commitment or guarantee of the document. If you have any doubt, please feel free to contact our company or authorized service provider, thank you! (The source code of the example can be download form www.inhaos.com.See the website for more technical support

Copyright

About Us

INHAOS is a high-tech private limited company combined with electronic products, telecommunications equipment, computer peripheral equipment development and sales. Aiming to promote domestic IT technological progress, we develop a series of embedded product development kit. This kit comes from large quantities of commercial product. The user can use it directly for design and verification, also can quickly convert the design to production and collect new product design ideas .

We also can undertake the following services:

Electronic product design

Brand components acting

Embedded development kit，Circuit module

Contact Us: http://www.inhaos.com/about.php?aID=7

文件名:	UM-LC-1000U-V10-EN.docx
目录:	C:\Documents and Settings\TONY\My Documents
模板:	C:\Documents and Settings\TONY\Application Data\Microsoft\Templates\Normal.dotm
标题:	
主题:	
作者:	TONY
关键词:	
备注:	
创建日期:	2012-9-21 14:15:00
修订号:	3,675
上次保存日期:	2013-5-29 16:09:00
上次保存者:	TONY
编辑时间总计:	6,992 分钟
上次打印时间:	2013-5-29 16:15:00
打印最终结果	
页数:	39
字数:	5,928 (约)
字符数:	33,790 (约)