/**

Barcode Reader DLL User Guide 1.1

First Written in : 2012-04-18

**/

Content:

1. Development Environment

2. Usage

3. Execution

4. API Reference

5. Program Flow Sheet

6. Example for VS2005 C++.

1. Development Environment

(1) Install the provided SDK by executing the“MF2351_SDK.msi”.
(2) Execute Microsoft VS2005 C++.
(3) Select “MF2351_SDK device” as the target of your application.
(4) Add “BCR_DLL.lib” to the project
(5) Include “BCR_DLL.h” in your source files.

2. Usage

(1) Library header
: “BCR_DLL.h” is the library header file. To use the BCR library, include the header file in your source header files.
(2) BCR API
: Call the API functions provided by the BCR library.

3. Execution

(1) Copy the provided BCR_DLL.dll and MF2351DLL.dll to the device. The location can be“ \ Windows” directory or the same directory with the application which uses the library .
.
(2) Copy the application that uses the BCR library in the device and execute it.

4. API Reference

[1] CODEAPI UINT WINAPI BCROpen(HWND);

Description :

Turns on the BCR power, opens the serial port and sets the window handle to receive data.

Return Values :

BARCODE_FAIL : the BCR has been opened once .
BARCODE_OK : Success
BARCODE_ERROR_POWERON : Failed to turn on power

Parameter :

HWND: [in] the handle of the window to receive data

---

[2] CODEAPI UINT WINAPI BCRClose(void);

Description :

Turns off the BCR power, closes the serial port and releases the window handle.

Return Values :

BARCODE_FAIL : the BCR has been Closed .
BARCODE_OK : Success

---

[3] CODEAPI UINT WINAPI BCREnableSideKey (BOOL bEanble);

Description :

Enable/Disable the side key to trigger on. This should be set before BCROpen () executed. Default: enable the side key to trigger on.

Return Values :

BARCODE_OK : Success

---

[4] CODEAPI UINT WINAPI BCRBeepOn(BOOL bBeep);

Description :

Sets the beep sound to notify successful reading of the barcode data.

Return Value :

BARCODE_FAIL: the BCR has been Closed .

BARCODE_OK : Success

Parameter :

BOOL bBeep: [in] TRUE:Set the beep sound

FALSE :Clear the beep sound

---

[5] CODEAPI UINT WINAPI BCRRead(char *a_pData, int length);

CODEAPI UINT WINAPI BCRReadString(TCHAR *a_tData, int length);

Description :

Reads the barcode data.

Return Value :

BARCODE_FAIL : the BCR has been Closed .

UINT : the actual length of the data was read.

Parameters :

char *a_pData : [in, out] pointer to the buffer which is used for storing the data get from the card.

TCHAR *a_tData : [in, out] pointer to the buffer which is used for storing the data get from the card.**the buffer length of the user based on the actual situation to decide, the general is not too small.**

int length : [in] the length of buffer which is used for storing the data get from the card.

---

[6] CODEAPI UINT WINAPI BCRTriggerStart(void);

Description :

Start barcode scanning .

Return Value :

BARCODE_FAIL: the BCR has been Closed .

BARCODE_ERROR_POWEROFF: failed.

BARCODE_OK : Success

---

[7] CODEAPI UINT WINAPI BCRTriggerStop(void);

Description :

Stop barcode scanning .

Return Value :

0L : the BCR has been Closed .

BARCODE_ERROR_POWEROFF: failed.

BARCODE_OK : Succes

---

[8] CODEAPI UINT WINAPI BCRSetDefault(void);

Description :

Set the Barcode Scanner in Default situation. BCR models must be reopened after this command.

Return Value :

BARCODE_FAIL : the BCR has been Closed .

UINT : TRUE （Non-zero）:Success.

FALSE（zero）:Failed.

---

[9] CODEAPI UINT WINAPI BCRSetParameters(char paramNumber, char paramValue, BOOL bTemporary);

Description :

1. Support or not support one barcode type .
2. Support the length of barcode type.
3. Set trigger mode.
4. Transmit code ID charactor.
   The code ID character is inserted between the prefix character(if selected) and the decode symbol. The symbol code ID character are listed below
   A = UPC-A, UPC-E, UPC-E1, EAN-8, EAN-13
   B = Code 39, Code32
   C = Codabar
   D = Code 128, ISBT 128
   E = Code 93
   F = Interleaved 2 of 5
   G = Discrete 2 of 5
   J = MSI
   K = GS1-Databar
   L = Bookland EAN
   M = Trioptic Code 39
   N = Coupon Code
   R = GS1 Databar-14, GS1 Databar Limited, GS1 Databar Expanded
5. Set add prefix and suffix type
6. Add prefix、suffix

Return Value :

BARCODE_FAIL: the BCR has been Closed .

UINT : TRUE （Non-zero）:Success.

FALSE（zero）:Failed.

Parameters :

char paramNumber : [in]

```
1.set enable/disable barcode type
BARCODE_TYPE_CODE39            0x00
BARCODE_TYPE_CODABAR           0x07
BARCODE_TYPE_CODE128           0x08
BARCODE_TYPE_DISCRETE2OF5      0x05
BARCODE_TYPE_INTERLEAVED2OF5   0x06
BARCODE_TYPE_CODE93            0x09
BARCODE_TYPE_UPC_A             0x01
BARCODE_TYPE_UPC_E             0x02
BARCODE_TYPE_UPC_E1            0x0C
BARCODE_TYPE_EAN8              0x04
BARCODE_TYPE_EAN13             0x03
BARCODE_TYPE_MSIPLESSEY        0x0B
BARCODE_TYPE_EAN128            0x08
BARCODE_UCCEAN_128_CODE        0x0E
BARCODE_ISBT_128_CODE          0x54
BARCODE_TYPE_TRIOPTIC_CODE39   0x0D
BARCODE_TYPE_BOOKLAND_EAN      0x53
BARCODE_TYPE_COUPON_CODE       0x55
2.set barcode length
LENGTH_INTERLEAVED2OF5_CODE    0x16
LENGTH_DISCRETE2OF5_CODE       0x14
LENGTH_CODABAR_CODE            0x18
 3.set trigger mode
 BAR_SET_TRIGGER             0x8A
 4.add\delete transmit code ID charactor
 BAR_SET_CODEID              0x2D
 5.set adding prefix、suffix type
 BAR_SET_PRESUFFIX           0xEB
 6.add prefix、suffix
 SET_PREFIX                  0x69
 SET_SUFFIX1                 0x68
 SET_SUFFIX2                 0x6A
```

char paramValue : [in]

```
//1.set enable/disable barcode type
 BARCODE_ENABLE          0x01
 BARCODE_DISABLE         0x00
 2.set barcode length
 ANY_LENGTH              0x00
 3.set trigger mode
 TRIGGER_PULSE           0x02
 TRIGGER_HOST            0x08
```

```
TRIGGER_BLINK                0x07
TRIGGER_CONTINUOUS           0x04
TRIGGER_LEVEL                0x00
4.add\delete transmit code ID charactor
ENABLE_CODEID                0x02
DISABLE_CODEID               0x00
5.set adding prefix、suffix type
SET_MODE_PRESUF1SUF2         0x07
SET_MODE_NONE                0x00
SET_MODE_SUF1                0x01
SET_MODE_SUF2                0x02
SET_MODE_SUF1SUF2            0x03
SET_MODE_PRE                 0x04
SET_MODE_PRESUF1             0x05
SET_MODE_PRESUF2             0x06
6.add prefix、suffix
The char to be added
```

BOOL bTemporary : [in] TRUE （Temporary change - lost when power removed.）
FALSE （Permanent change.）

5. Application Flow Sheet：

(1)Read data.

step1: Open the Scaner

```
BCREnableSideKey(BOOL);
BCROpen(HWND);
```

Step2: Set trigger mode ： “Host”

```
CodeSetParameters(BAR_SET_TRIGGER, TRIGGER_HOST, TRUE);
```

Step3: Start barcode scanning

```
BCRTriggerStart(void);
```

Step4 : Wait message for barcode data received

```
WM_BARCODE_COMPLETE
```

Step5: Read the data

```
BCRReadString(…,…);
```

Step6: Stop barcode scanning

```
BCRTriggerStop();
```

Step7: Close Scanning Equipment

```
CodeClose();
```

(2)Set beep.

step1: Open the Scaner

```
BCROpen(HWND);
```

step2: Turn on the beep.

```
BCRtBeepOn(TRUE);
```

Turn off the beep.

BCRBeepOn(FALSE);

step3: Close Scaner

BCRClose();

(3) Enable or Disnable one Type barcode

step1: Open the Scaner

BCROpen(HWND);

step2: Enable or disnable one type

BCRSetParameters(...,...,FALSE);

step3: Close Scaner

BCRClose();

(4) Set trigger mode

step1: Open the Scaner

BCROpen(HWND);

step2: Set trigger mode

BCRSetParameters(BAR_SET_TRIGGER, TRIGGER_BLINK, FALSE); //Blink

BCRSetParameters(BAR_SET_TRIGGER, TRIGGER_PULSE, FALSE); //Pulse

step3: Close Scaner

BCRClose();

(5) Transmit code ID charactor

step1: Open the Scaner

BCROpen(HWND);

step2: Set transmit code ID charactor

add transmit code ID charactor

BCRSetParameters(BAR_SET_CODEID, ENABLE_CODEID, FALSE);

delete transmit code ID charactor

BCRSetParameters(BAR_SET_CODEID, DISABLE_CODEID, FALSE);

step3: Close Scaner

BCRClose();

(6) Add prefix、suffix

step1: Open the Scaner

BCROpen(HWND);

step2: Set the format type of adding prefix、suffix

BCRSetParameters(BAR_SET_PRESUFFIX, …, FALSE);

step3: Add prefix、suffix

BCRSetParameters(SET_PREFIX, char, FALSE);

BCRSetParameters(SET_SUFFIX1, char, FALSE);

BCRSetParameters(SET_SUFFIX2, char, FALSE);

step4: Close Scaner

BCRClose();

6. Example for eVC++ 4.0.

```
void CBCRTestProgramDlg::OnOpencode()
{
    BCREnableSideKey(true);//press the side key to trigger on
```

```
    BCROpen(GetSafeHwnd());
}

void CBCRTestProgramDlg::OnClosecode()
{
    BCRClose();
}

BOOL CBCRTestProgramDlg::PreTranslateMessage(MSG* pMsg)
{
    switch(pMsg->message)
    {
        case WM_BARCODE_MESSAGE:
            break;
        case WM_BARCODE_COMPLETE:
            {
                TCHAR pData[1024];
                memset(pData,0,1024);
                BCRReadString(pData,1024);
            }
            break;
    }
    return CDialog::PreTranslateMessage(pMsg);
}

void CBCRTestProgramDlg::OnSetdefault()
{
    if(!BCRSetDefault())
    {
        MessageBox(_T("SetDefault Invaild Setting"));
    }
}

void CBCRTestProgramDlg::OnCode93en()
{
    if(!BCRSetParameters(BARCODE_CODE_93_CODE,BARCODE_ENABLE,F
ALSE))
    {
        MessageBox(_T("SetParameters Invaild Setting"));
    }
    else
    {
        MessageBox(_T("Code93 Setting enable Complete"));
    }
```

```
}

void CBCRTestProgramDlg::OnCode93dis()
{
    if(!BCRSetParameters(BARCODE_CODE_93_CODE,BARCODE_DISABLE,F
ALSE))
    {
        MessageBox(_T("SetParameters Invaild Setting"));
    }
    else
    {
        MessageBox(_T("Code93 Setting disable Complete"));
    }
}

void CBCRTestProgramDlg::OnCodeInterleaved2of5_Length()
{

    if(!BCRSetParameters(LENGTH_INTERLEAVED2OF5_CODE,ANY_LENGT
    H,FALSE))
    {
        MessageBox(_T("SetParameters Invaild Setting"));
    }
    else
    {
        MessageBox(_T("CodeInterleaved2of5 Setting To Any Length Complete"));
    }
}


void CBCRTestProgramDlg::OnSetTriggerModeBlink()
{
   BCRSetParameters(BAR_SET_TRIGGER, TRIGGER_BLINK, FALSE);         //Blink
 //BCRSetParameters(BAR_SET_TRIGGER, TRIGGER_CONTINUOUS, FALSE); //Continuous
 //BCRSetParameters(BAR_SET_TRIGGER, TRIGGER_PULSE, FALSE);      //Pulse
 //BCRSetParameters(BAR_SET_TRIGGER, TRIGGER_HOST, FALSE);      // Host
}

void CBCRTestProgramDlg::OnClose()
{

    BCRClose();
```

```
    CDialog::OnClose();
}

void CBCRTestProgramDlg::OnOK()
{
    BCRClose();
    CDialog::OnOK();
}

void CBCRTestProgramDlg::OnTrion()
{
    CodeSetParameters(BAR_SET_TRIGGER, TRIGGER_HOST, TRUE);
    BCRTriggerStart();
}

void CBCRTestProgramDlg::OnTrioff()
{
    BCRTriggerStop();
}

void CBCRTestProgramDlg::OnBeepon()
{
    BCRSetBeep(TRUE);
}

void CBCRTestProgramDlg::OnBeepoff()
{
    BCRSetBeep(FALSE);
}
void CBCRTestProgramDlg::OnBnClickedBaddid()
{
    //add transmit code ID charactor
    CodeSetParameters(BAR_SET_CODEID, ENABLE_CODEID, FALSE);
}
void CBCRTestProgramDlg::OnBnClickedBdeleteid()
{
    //delete transmit code ID charactor
    CodeSetParameters(BAR_SET_CODEID, DISABLE_CODEID, FALSE);
}
void CBCRTestProgramDlg::AddPrefixAndSuffix()
{
  Char cSendCmmd[3] = {0};
 //Sample code 1.
    //Add prefix and suffix1
```

```
        cSendCmmd[0] = '%';
        cSendCmmd[1]= '#';
      // set adding prefix、suffix type
        CodeSetParameters(BAR_SET_PRESUFFIX, SET_MODE_PRESUF1, FALSE);
      // add prefix
        CodeSetParameters(SET_PREFIX, cSendCmmd[0], FALSE);
      // add suffix1
        CodeSetParameters(SET_SUFFIX1, cSendCmmd[1], FALSE);
//Sample code 2.
      //Add suffix1 and suffix2
        cSendCmmd[1] = '#';
        cSendCmmd[2]= '*';
      // set adding suffix1、suffix2 type
        CodeSetParameters(BAR_SET_PRESUFFIX, SET_MODE_SUF1SUF2, FALSE);
     // add suffix1
        CodeSetParameters(SET_SUFFIX1, cSendCmmd[1], FALSE);
      // add suffix2
        CodeSetParameters(SET_SUFFIX2, cSendCmmd[2], FALSE);
}
```