AN3954
Application note

Developing your own Visual Basic or C/C++ application on a CR95HF demonstration board

Introduction

This application note explains how to develop an application using either Visual Basic or C/C++ language to drive a CR95HF demonstration board from a host computer.

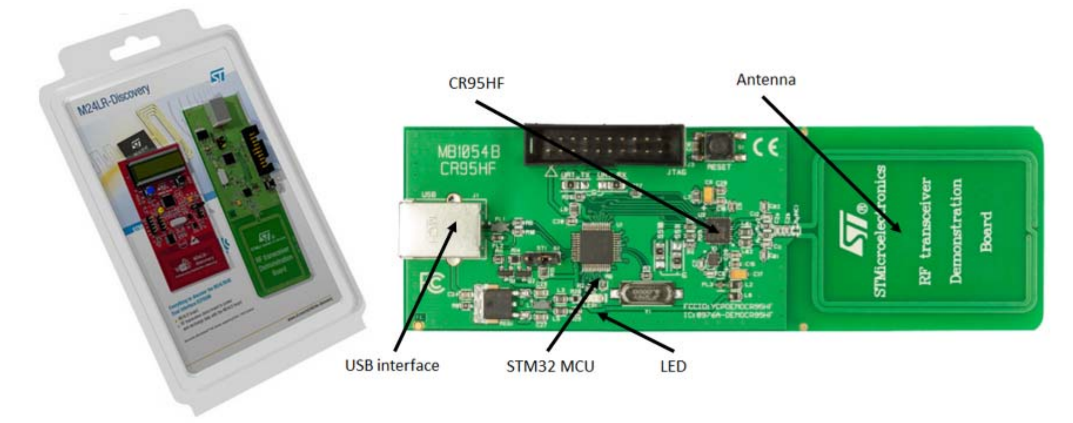
CR95HF demonstration board is a kit that allows the user to evaluate the performance of STMicroelectronics CR95HF, a 13.56 MHz multiprotocol contactless transceiver. It includes a ready-to-use board to interface with the CR95HF host PC demonstration software through an USB interface. The CR95HF demonstration board is included in the M24LR-DISCOVERY kit (see *Figure 1*).

The board is powered through the USB port, and no external power supply is required. It is made up of a CR95HF, a 48 mm x 34 mm, 13.56 MHz inductive etched antenna and its associated tuning components. The CR95HF communicates with the STM32F103CB 32-bit core microcontroller via the SPI bus.

A dynamic link library (DLL) file is used by the host computer to manage several functions and to communicate with both the STM32 MCU and the CR95HF

Source code examples described in this application note and additional demonstrations can be found in the package STSW-95HF002, which can be downloaded from *www.st.com*. Some basic examples are available in C/C++, additional examples are available in a Visual Basic 6 project, among them NDEF message management on Type 4A and Type 5, ST25DV user interface and FTM demonstrations. DLL file is also available in this package.

Figure 1. CR95HF demonstration board embedded in the M24LR-Discovery kit



For more information, refer to the following documents, available on *www.st.com*:

- M24LR-DISCOVERY databrief
- CR95HF datasheet
- User manual UM1084 “CR95HF development software”
- STSW-95HF002 source code example

Contents

List of tables

List of figures

1 Getting started

This application note and the source code example have been written to help users to easily develop their applications. This document note contains function descriptions and some use cases.

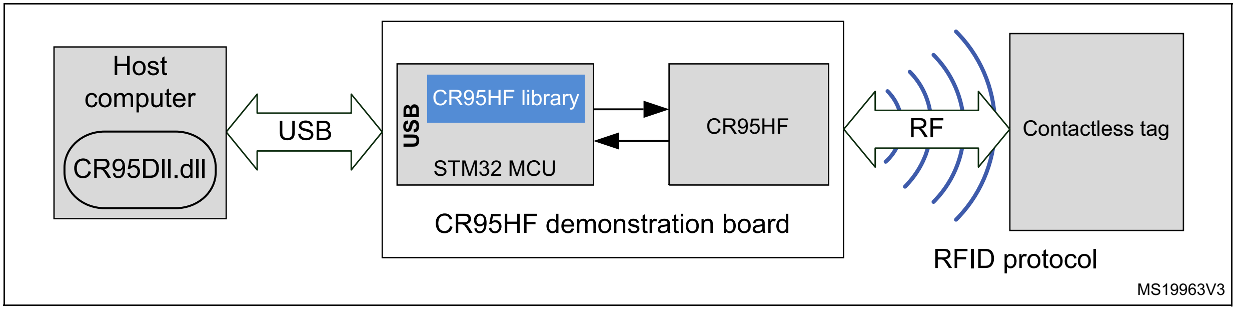
To improve the use of CR95HF demonstration board and the communication with tags, it is recommended to download and use the CR95HF development software (STSW-95HF001) available on *www.st.com.* This software (and not this application note) can be used as a reference to improve and test the CR95HF demonstration board functions and the communication with tags, in various RF protocols.

1.1 Connecting the board to the computer

As shown in *Figure 2*, the CR95HF demonstration board is connected to the host computer through its USB port, no particular driver is needed. The HID USB driver (already included in all Windows operating systems) is used for communication.

Through the USB port of the computer, the user can drive the board, communicate with the STM32 MCU, manage the CR95HF IC and send requests to an RF tag.

Figure 2. Typical application block diagram



1.2 Using the DLL system file

A DLL file (CR95HF.dll) is provided to drive the CR95HF demonstration board from your computer, copy this file to the system folder of your computer (C:/Windows/system32/).

This DLL contains several functions to enable communication with the board.

Using this DLL, the host computer is able to manage several functions and communicate with the STM32 MCU and the CR95HF IC.

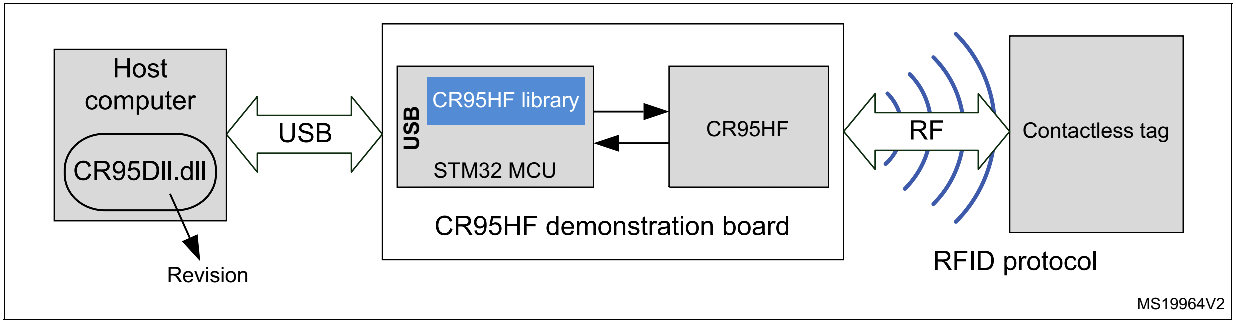
1.2.1 Overview of DLL functions

The *CR95HFDll_GetDLLrev* function reads the revision number of the DLL file. If the DLL file is not correctly installed in the system folder of the host computer, this function returns an error message (see *Figure 3*).

The *CR95HFDLL_USBconnect* function (available from DLL revision 0.5 and higher) is required to connect the CR95HF demonstration board to the host computer and communicate with it. The computer first checks if a CR95HF demonstration board is connected to the USB port. If so, it assigns a USB handle to the board (see *Figure 3*).

The *CR95HFDLL_USBhandlecheck* function can be used to verify if the CR95HF demonstration board is connected to the USB port of the computer. It is available from DLL revision 0.6 and higher. This function sends a request to the CR95HF demonstration board and waits for an acknowledge. An error code is sent if the communication with the board has been lost. In this case, the CR95HFDLL_USBconnect can be used to reconnect the board to the host computer.

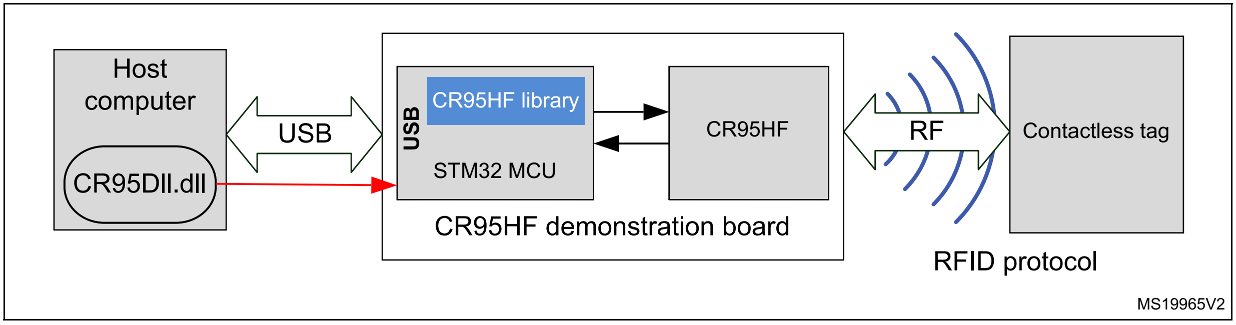
Figure 3. Function CR95HFDLL_USBconnect



A set of functions are available to drive the STM32 MCU (see *Figure 4*):

- *CR95HFDll_Echo*: verifies that the CR95HF demonstration board is connected to the USB port of the computer.
- *CR95HFDLL_getMCUrev*: reads the firmware revision of the STM32 MCU.
- *CR95HFDLL_getHardwareVersion*: reads the hardware version of the boards (available on latest MCU revision boards / introduced from dll revision 0.9).
- *CR95HFDLL_getInterfacePinState*: verifies the communication path between the STM32 MCU and the CR95HF IC (either SPI or UART).

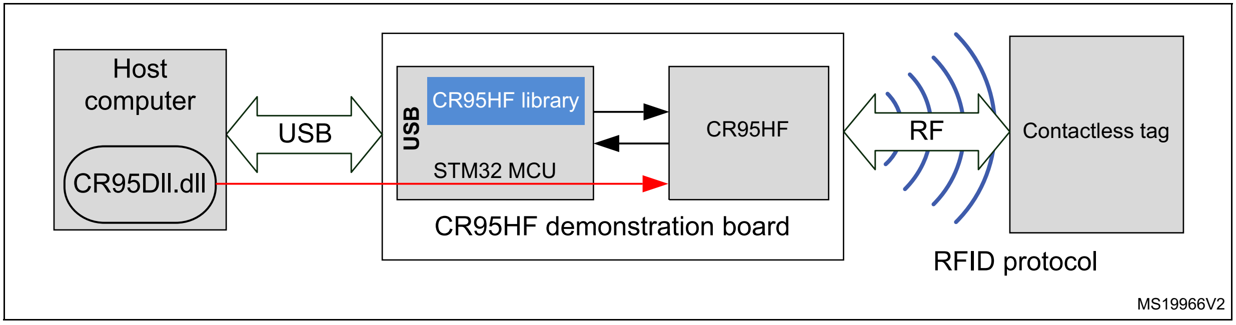
Figure 4. Driving the MCU on the CR95HF demonstration board from the PC



Other functions can be used to send instructions to the CR95HF IC (see *Figure 5*):

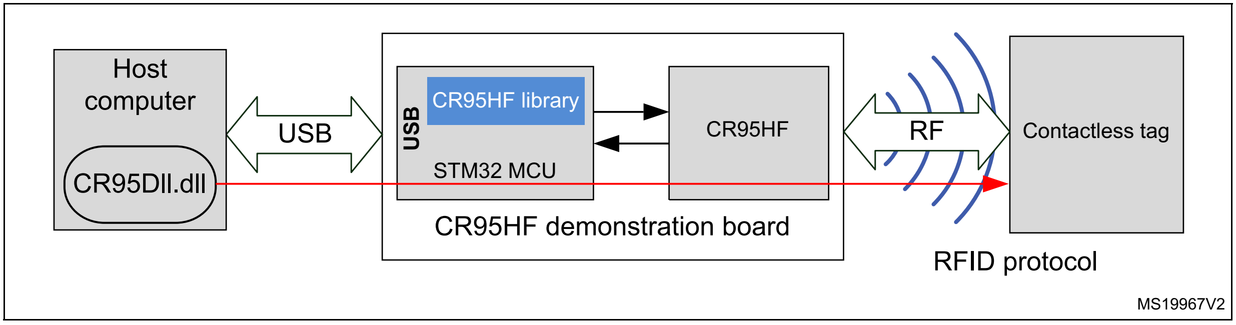
- *CR95HFDll_Idn*: gets the IDN of the CR95HF IC.
- *CR95HFDll_Select*: configures the required RF protocol for the CR95HF IC (IS0 15693, ISO 14443B …) and starts the RF field on the antenna.
- *CR95HFDll_STCmd*: sends commands to the CR95HF IC (refer to datasheet for a list of possible commands).
- *CR95HFDll_FieldOff*: stops the RF field of the CR95HF IC.
- *CR95HFDll_ResetSPI*: resets the SPI interface of the CR95HF IC.
- *CR95HFDll_SendIRQPulse*: sends an interruption pulse on the IRQ pin of the CR95HF IC to wake it up. This interrupt function can be used when the communication between the STM32 MCU and the CR95HF demonstration board is performed in SPI mode.
- *CR95HFDll_SendNSSPulse*: sends an interrupt pulse to the NSS pin of the CR95HF IC to wake it up. This interrupt function can be used when the communication between the STM32 MCU and the CR95HF demonstration board is performed in UART mode.
- *CR95HFDll_Polling_Reading*: launches a polling sequence and reads the CR95HF answer buffer to retrieve the answer when it is not available in the FIFO answer buffer (e.g. after an IDLE command)

Figure 5. Driving the CR95HF from the PC



Finally, the *CR95HFDll_SendReceive* function sends an RF frame to any tag present in the RF field and receives its answer according to the RF protocol previously defined using the Select command (see *Figure 6*).

Figure 6. Sending frames directly to the tag



2 Description of functions

This section describes the various functions required to evaluate the CR95HF demonstration board.

2.1 Functions to check DLL installation and USB connection

2.1.1 CR95HFDll_GetDLLrev

This function detects if the *CR95HF.dll* file has been installed in the system folder of the host PC system, reads its header and sends back its revision number.

This is used by software to ensure that the DLL is correctly installed and that it is up to date.

If no DLL file is found, it returns an error message.

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDll_GetDLLrev Lib
"CR95HF.dll" (ByVal mystring As String) As Long
```

Prototype:

```
Dim lngStatus As Long
Dim strAnswer  As String * 256
lngStatus = CR95HFDll_GetDLLrev(strAnswer)
```

Input parameter: None

Output parameter: strAnswer: Revision of the DLL installed on your PC system.
Answer example: "0.4"

Returned value: lngStatus: 0

Source code example

```
Private Sub cmdGetDllRev_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long
lngStatus = CR95HFDll_GetDLLrev(strAnswer)
If (lngStatus = 0) Then
     txtDllRev.Text = strAnswer
Else
     txtDllRev.Text = "Error = no Dll found"
End If
End Sub
```

C/C++ language

Declaration: `__declspec(dllimport) int __stdcall CR95HFDll_GetDLLrev(char* stringreply);`

Prototype:

```
int iresult;
char strAnswer[50]="";
iresult = CR95HFDll_GetDLLrev(strAnswer);
```

Input parameter: None

Output parameter: strAnswer: Revision of the DLL installed on your PC system.
Answer example: "0.4"

Returned value: Iresult: 0

Source code example

```
int Get_Dll_rev (void)
{
     char strAnswer[50]="";
     int iresult;
     char entry3;
     printf("\n\n\n\n");
     iresult = CR95HFDll_GetDLLrev(strAnswer);
     printf("\nGet Dll revision using CR95HFDll_GetDLLrev
     function:\n");
     printf("\n --> request : CR95HFDll_GetDLLrev(strAnswer)");
     printf("\n    <-- answer  : ");
     if (iresult == 0)
          printf("%s = DLL version\n\n\r", strAnswer);
     else
          printf("%s = Error : no communication with MCU\n\n\r",
          strAnswer);
     printf("\n\n\n\n");
     printf("\npress any key to continue");
     printf("\n");
     scanf("%s", &entry3);
     return iresult;
}
```

2.1.2 CR95HFDLL_USBconnect

This function can be used to connect the CR95HF demonstration board to the computer USB port. The computer then assigns a USB handle to the board.

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDLL_USBconnect
Lib"CR95HF.dll" () As Long
```

Prototype:

```
Dim lngStatus As Long
lngStatus = CR95HFDLL_USBconnect()
```

Input parameter: None

Output parameter: None

Returned value: lngStatus:

0: the CR95HF demonstration board is connected and ready to be used

1: the CR95HF demonstration board is not connected

Source code example

```
Private Sub cmdUSBconnect_Click()
Dim lngStatus As Long
lngStatus = CR95HFDLL_USBconnect()
If (lngStatus = 0) Then
     txtUSBconnectAnswer.Text="CR95HF demonstration board connected
     and ready to be used"
Else
     txtUSBconnectAnswer.Text = "CR95HF demonstration board not
     connected"
End If
End Sub
```

C/C++ language

Declaration:

```
__declspec(dllexport) int  __stdcall
CR95HFDLL_USBconnect(void);
```

Prototype:

```
int iresult;
iresult= CR95HFDLL_USBconnect ();
```

Input parameter: None

Output parameter: None

Returned value: iresult:

0: the CR95HF demonstration board is connected and ready to be used

1: the CR95HF demonstration board is not connected

Source code example

```
int DEMO_CR95HF_A_USBconnect (void)
{
  int iresult;
  char entry3;
     printf("\n\n\n\n");
```

```
           iresult= CR95HFDLL_USBconnect ();
           printf("\nCR95HFDll_USBconnect function:\n");
           printf("\n    --> request : CR95HFDll_USBconnect()");
           printf("\n    <-- answer  : ");
           if (iresult == 0)
              printf("CR95HF demonstration board is connected and ready to be
              used\n\n\r");
           else
              printf("Error : CR95HF demonstration board not ready to be
              used\n\n\r");
           printf("\n\n\n\n");
           printf("\npress any key to continue");
           printf("\n");
           scanf("%s", &entry3);
           return iresult;
      }
```

2.1.3 CR95HFDLL_USBhandlecheck

This function can be used to verify if the CR95HF demonstration board is correctly connected to the USB port of the computer and if the USB Handle assigned by the CR95HFDLL_USBconnect is always valid.

Visual Basic language

Declaration: `Public Declare Function CR95HFDLL_USBhandlecheck Lib "CR95HF.dll" () As Long`

Prototype:

```
Dim lngStatus as long
lngStatus = CR95HFDLL_USBhandlecheck()
```

Input parameter: None

Output parameter: None

Returned value: lngStatus:

0: the USB handle is valid

5: the USB handle is invalid and the CR95HF demonstration board not connected

Source code example

```
Private Sub cmdUSBhandlecheck_Click()
Dim lngStatus As Long
txtUSBhandlecheckAnswer.Text = ""
lngStatus = CR95HFDLL_USBhandlecheck()
If (lngStatus = 0) Then
    txtUSBhandlecheckAnswer.Text = "CR95HF demonstration board
    connected"
ElseIf (lngStatus = 5) Then
    txtUSBhandlecheckAnswer.Text = "CR95HF demonstration board not
    connected (use CR95HFDLL_USBconnect to connect it)"
End If

End Sub
```

C/C++ language

Declaration:

```
__declspec(dllexport) int  __stdcall
CR95HFDLL_USBhandcheck(void);
```

Prototype:

```
int iresult;
iresult= CR95HFDLL_USBhandcheck();
```

Input parameter: None

Output parameter: None

Returned value: iresult:

0: the USB handle is valid
5: the USB handle is invalid and the CR95HF demonstration board not connected

Source code example

```
int DEMO_CR95HF_A_USBhandlecheck (void)
{
  int iresult;
  char entry3;
  printf("\n\n\n\n");
  iresult= CR95HFDLL_USBhandlecheck ();
  printf("\nCR95HFDll_USBhandlecheck function:\n");
  printf("\n    --> request : CR95HFDll_USBhandlecheck()");
  printf("\n    <-- answer  : ");
  if (iresult == 0)
printf("DEMO-CR95HF-A is well connected\n\n\r");
  else
printf("Error : CR95HF demonstration board not connected (use
CR95HFDLL_USBConnect)\n\n\r");
  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);
  return iresult;
}
```

2.2 Functions to communicate with the STM32 MCU

2.2.1 CR95HFDll_Echo

This function sends a USB request to the STM32 MCU that executes an Echo request on the CR95HF. The STM32 MCU sends back the answer of the CR95HF, if possible, or returns an error code ‘1’ if there is no answer.

Note: The Echo request uses the EchoCode command to perform a serial interface echo. For more information, refer to the CR95HF transceiver datasheet.

Visual Basic language

Declaration: `Public Declare Function CR95HFDll_Echo Lib "CR95HF.dll" (ByVal mystring As String) As Long`

Prototype:

```
Dim lngStatus as long
Dim strAnswer  As String * 256
lngStatus = CR95HFDll_Echo(strAnswer)
```

Input parameter: None

Output parameter: strAnswer: The CR95HF IC answer to the Echo request is “5500” if there is no error.
Answer example: “5500”

Returned value: lngStatus:
0: No error
5: CR95HF demonstration board not connected

Source code example

```
Private Sub cmdEcho_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_Echo(strAnswer)

If (lngStatus = 0) Then
    txtEchoAnswer.Text = strAnswer
Else
    txtEchoAnswer.Text = "No communication with CR95HF demonstration
     board"
End If

End Sub
```

C/C++ language

Declaration: `__declspec(dllimport) int  __stdcall CR95HFDll_Echo(char *Stringreply);`

Prototype:

```
int iresult;
char strAnswer[50]="";
iresult= CR95HFDll_Echo (strAnswer);
```

Input parameter: None

Output parameter: strAnswer: The CR95HF IC answer to the Echo request is “5500” if there is no error.
Answer example: “5500”

Returned value: iresult:
0: No error
5: CR95HF demonstration board not connected

Source code example

```
int Echo (void)
{
  char strAnswer[50]="";
  int iresult;
  char entry3;
```

```
    printf("\n\n\n\n");

    iresult= CR95HFDll_Echo (strAnswer);

    printf("\nEcho using CR95HFDll_Echo function:\n");
    printf("\n    --> request : CR95HFDll_Echo(strAnswer)");
    printf("\n    <-- answer  : ");

    if (iresult == 0)
            printf("%s = Echo command answer\n\n\r", strAnswer);
    else
            printf("%s = No answer from MCU\n\n\r", strAnswer);

    printf("\n\n\n\n");
    printf("\npress any key to continue");
    printf("\n");
    scanf("%s", &entry3);

    return iresult;
  }
```

2.2.2 CR95HFDLL_getMCUrev

This function sends a USB request to the STM32 MCU on the CR95HF demonstration board that sends back the revision number of its firmware.

Visual Basic language

Declaration: `Public Declare Function CR95HFDLL_getMCUrev Lib "CR95HF.dll" (ByVal mystring As String) As Long`

Prototype: `status = CR95HFDLL_getMCUrev(strAnswer)`

Input parameter: None

Output parameter: strAnswer: Firmware revision of the STM32 MCU on CR95HF demonstration board.
Answer example: “0003010300”

Where:
00: Status byte (see *Appendix A* for error codes)
03: Size of answer (in bytes)
010300: Revision 1.3.0

Returned value: lngStatus:
0: No error
5: CR95HF demonstration board not connected

Source code example

```
Private Sub cmdGetMCURev_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDLL_getMCUrev(strAnswer)

If (lngStatus = 0) Then
    txtMCUrev.Text = strAnswer
```

```
Else
    txtMCUrev.Text = "Error = no communication with STM32 MCU"
End If

End Sub
```

C/C++ language

Declaration:

```
__declspec(dllexport) int  __stdcall
CR95HFDLL_getMCUrev(char* stringreply);
```

Prototype

```
int iresult;
char strAnswer[50]="";
iresult = CR95HFDLL_getMCUrev(strAnswer);
```

Input parameter: None

Output parameter: strAnswer: Firmware revision of the STM32 MCU on CR95HF demonstration board.
Answer example: “0003010300”

Where:
00: Status byte (see *Appendix A* for error codes)
03: Size of answer (in bytes)
010300: Revision 1.3.0

Returned value: iresult:
0: No error
5: CR95HF demonstration board not connected

Source code example

```
int Get_MCU_rev (void)
{
  char strAnswer[50]="";
  int iresult;
  char entry3;

  printf("\n\n\n\n");

iresult = CR95HFDLL_getMCUrev(strAnswer);

  printf("\nGet MCU revision using CR95HFDLL_getMCUrev function:\n");
  printf("\n    --> request : CR95HFDLL_getMCUrev(strAnswer)");
  printf("\n    <-- answer  : ");

  if (iresult == 0)
          printf("%s = MCU version\n\n\r", strAnswer);
  else
          printf("%s = Error : no communication with MCU\n\n\r",
strAnswer);

  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

  return iresult;
}
```

2.2.3 CR95HFDLL_getHardwareVersion

This function sends a USB request to the STM32 MCU on the CR95HF demonstration board that sends back the hardware revision of the board.

Note: This function is available since DLL revision 0.9 and higher.
The hardware revision number is available since CR95HF demonstration board with MCU revision 2.1 (produced after 2012-10).

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDLL_getHardwareVersion Lib
"CR95HF.dll" (ByVal mystring As String) As Long
```

Prototype

```
lngStatus = CR95HFDLL_getHardwareVersion(strAnswer);
```

Input parameter: None

Output parameter: strAnswer: Hardware revision of the CR95HF demonstration board.
Answer example: "00074D423938335F41"

Where:
00: Status byte (see *Appendix A* for error codes)
03: Size of answer (in bytes)
4D423938335F41: Hardware revision (ASCII = MB983_A)

Answer example: "FE00"
Error return: CR95HF demonstration board doesn't contain Hardware revision (too old)

Returned value: lngStatus:
0: No error
5: CR95HF demonstration board not connected

Source code example

```
Private Sub cmdGetHardwareRev_Click()
Dim strAnswer As String * 256
Dim lngStatus As Long
Dim i As Integer

txtHardwareRevAscii.Text = ""

lngStatus = CR95HFDLL_getHardwareVersion(strAnswer)

If (lngStatus = 0) Then
    If (Mid(strAnswer, 1, 2) = "00") Then
        txtHardwareRev.Text = strAnswer
        For i = 0 To Mid(strAnswer, 3, 2) - 1
            txtHardwareRevAscii.Text = txtHardwareRevAscii.Text &
            Chr(CLng("&h" & Mid(strAnswer, (i * 2) + 5, 2)))
        Next i
    Else
        txtHardwareRev.Text = "Hardware doesn't contain Hardware
        Revision"
    End If
Else
    txtHardwareRev.Text = "Error = no communication with STM32 MCU"
End If

End Sub
```

C/C++ language

Declaration: `__declspec(dllexport) int __stdcall CR95HFDLL_getHardwareVersion(char* StringReply);`

Prototype `iresult = CR95HFDLL_getHardwareVersion(strAnswer);`

Input parameter: None

Output parameter: strAnswer: Hardware revision of the CR95HF demonstration board

Answer example: "00074D423938335F41"

Where:
00: Status byte (see *Appendix A* for error codes)
03: Size of answer (in bytes)
4D423938335F41: Hardware revision (ASCII = MB983_A)

Answer example: "FE00"
Error return: CR95HF demonstration board doesn't contain Hardware revision (too old)

Returned value: lngStatus:
0: No error
5: CR95HF demonstration board not connected

Source code example

```
int Get_Hardware_rev (void)
{
 char strAnswer[50]="";
 int iresult;
 char entry3;

 printf("\n\n\n\n");

 iresult = CR95HFDLL_getHardwareVersion(strAnswer);

 printf("\nGet Hardware revision using CR95HFDLL_getHardwareVersion
 function:\n");
 printf("\n    --> request : CR95HFDLL_getHardwarerev(strAnswer)");
 printf("\n    <-- answer  : ");

 if (iresult == 0)
      printf("%s = Hardware revision\n\n\r", strAnswer);
   else
      printf("%s = Error : no communication with MCU\n\n\r", strAnswer);

 printf("\n\n\n\n");
 printf("\npress any key to continue");
 printf("\n");
 scanf("%s", &entry3);

 return iresult;
}
```

2.2.4 CR95HFDLL_getInterfacePinState

This function verifies the communication path between the STM32 MCU and the CR95HF IC (either SPI or UART).

The STM32 MCU checks which communication configuration is selected on the CR95HF demonstration board. Two solder bridges are available to select the SPI or UART configuration. By default, the board is delivered in the SPI configuration.

Visual Basic language

Declaration: `Public Declare Function CR95HFDLL_getMCUrev Lib "CR95HF.dll" (ByVal mystring As String) As Long`

Prototype: `status = CR95HFDLL_getMCUrev(strAnswer)`

Input parameter: None

Output parameter: strAnswer: Interface Pin state.
Answer example: “00010X”

Where:

00: Status byte (see *Appendix A* for error codes)

01: Size of answer (in bytes)
0X: Communication mode

With X:
0: Communication in UART mode
1: Communication in SPI Mode

Returned value: lngStatus:
0: No error
5: CR95HF demonstration board not connected

Source code example

```
Private Sub cmdGetInterfacePinState_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

chkInterfacePinState(0).Value = vbUnchecked
chkInterfacePinState(1).Value = vbUnchecked

lngStatus = CR95HFDLL_getInterfacePinState(strAnswer)

If (lngStatus = 0) Then
    txtGetInterfacePinStateAnswer.Text = strAnswer
    If (Mid(strAnswer, 6, 1) = "0") Then
        chkInterfacePinState(0).Value = vbChecked
        chkInterfacePinState(1).Value = vbUnchecked
    ElseIf (Mid(strAnswer, 6, 1) = "1") Then
        chkInterfacePinState(0).Value = vbUnchecked
        chkInterfacePinState(1).Value = vbChecked
    Else
        DoEvents
        'error
    End If
Else
    txtGetInterfacePinStateAnswer.Text = "No answer from the DEMO-
CR95HF-A board"
End If

End Sub
```

C/C++ language

Declaration:

```
__declspec(dllexport) int  __stdcall
CR95HFDLL_getInterfacePinState(char* stringreply);
Prototypeint iresult;
char strAnswer[50]="";
iresult = CR95HFDLL_getInterfacePinState(strAnswer);
```

Input parameter: None

Output parameter: strAnswer: Interface Pin state.
Answer example: “00010X”

Where:
00: Status byte (see *Appendix A* for error codes)
01: Size of answer (in bytes)
0X: Communication mode

With X:
0: Communication in UART mode
1: Communication in SPI Mode

Returned value: iresult:
0: No error
5: CR95HF demonstration board not connected

Source code example

```
int Get_Interface_Pin_State (void)
{
  int iresult;
  char strAnswer[50]="";
  char entry3;

  printf("\n\n\n\n");

  iresult = CR95HFDLL_getInterfacePinState(strAnswer);

  printf("\nGet Interface Pin State CR95HFDLL_getInterfacePinState 
function:\n");
  printf("\n    --> request : 
CR95HFDLL_getInterfacePinState(strAnswer)");
  printf("\n    <-- answer  : ");

  if (iresult == 0)
  {
       printf("%s = Get Interface Pin State successfully\n\n\r", 
strAnswer);
       if (strAnswer[5] == '0')
          printf("\n        %c = communication in UART 
mode",strAnswer[3]);
       else
          printf("\n        %c = communication in SPI 
mode",strAnswer[3]);
  }
  else
       printf("\n        %s = No answer from CR95HF demonstration board 
\n\n\r", strAnswer);

  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

  return iresult;
}
```

2.3 Functions to communicate with the CR95HF IC

2.3.1 CR95HFDll_Idn

This function sends a USB request to the STM32 MCU that requests the IDN of the CR95HF IC. The STM32 MCU sends back the answer of the CR95HF containing the IDN value (ASCII codes), if possible, or returns an error code ‘1’ if there is no answer.

Note: The IDN command requests brief information about the CR95HF and its firmware version. For more information, refer to the CR95HF transceiver datasheet.

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDll_Idn Lib "CR95HF.dll"
(ByVal mystring As String) As Long
```

Prototype:

```
Dim strAnswer  As String * 256
Dim lngStatus As Long
status = CR95HFDll_Idn(strAnswer)
```

Input parameter: None

Output parameter: strAnswer: IDN of the CR95HF IC (if no error)
Answer example: “000F4E4643204653324A4153543000A998”

Where:
00: Status byte (see annexe 2 for error codes)
0F: Size of answer (in bytes)
4E4643204653324A41535430: ASCII transcription of the CR95HF IDN (in this example, IDN = “NFC FS2JAST0”)
00: protocol status
A998: CRC value

Returned value: lngStatus:
0: No error
5: CR95HF demonstration board not connected

Source code example

```
Private Sub cmdIDnRequest_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_Idn(strAnswer)

If (lngStatus = 0) Then
    txtIdnAnswer.Text = strAnswer
Else
    txtIdnAnswer.Text = "Idn not sent back by the CR95HF IC"
End If

End Sub
```

C/C++ language

Declaration:

```
__declspec(dllimport) int  __stdcall CR95HFDll_Idn(char
*Stringreply);
Prototype int iresult;
char strAnswer[50]="";
iresult= CR95HFDll_Idn (strAnswer);
```

Input parameter: None

Output parameter: strAnswer: IDN of the CR95HF IC (if no error)
Answer example: “000F4E4643204653324A4153543000A998”

Where:

00: Status byte (see annexe 2 for error codes)
0F: Size of answer (in bytes)
4E4643204653324A41535430: ASCII transcription
of the CR95HF IDN (in this example,
IDN = “NFC FS2JAST0”)
00: protocol status
A998: CRC value

Returned value: iresult:

0: No error
5: CR95HF demonstration board not connected

Source code example

```
int Idn (void)
{
  int iresult;
  char strAnswer[50]="";
  char entry3;

  printf("\n\n\n\n");

  iresult= CR95HFDll_Idn (strAnswer);

  printf("\nIdn using CR95HFDll_Idn function:\n");
  printf("\n    --> request : CR95HFDll_Idn(strAnswer)");
  printf("\n    <-- answer  : ");

  if (iresult == 0)
          printf("%s = Idn command response\n\n\r", strAnswer);
  else
          printf("%s = Error : no Idn returned\n\n\r", strAnswer);

  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

  return iresult;
}
```

2.3.2 CR95HFDll_Select

This function sends a USB request to the STM32 MCU that prepares the CR95HF for communication by executing a Select request containing the selected RF parameters to the the CR95HF IC. The STM32 MCU sends back the answer of the CR95HF, if possible, or returns an error code ‘1’ if there is no answer. In addition to selecting the correct RF communication parameters, this function activates the RF field. This is necessary at the start of communications if the RF field has been previously switched off.

The parameters of this function indicate the selected RF communication protocol (ISO 15693, ISO 14443A, ISO 14443B, FELICA) as defined in the CR95HF datasheet.

Note: The Select request uses the Protocol Select command to select the RF communication protocol and specify certain protocol-related parameters. For more information, refer to the CR95HF transceiver datasheet.

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDll_Select Lib
"CR95HF.dll" (ByVal mycmdstring As String, ByVal
mystring As String) As Long
```

Prototype:

```
Dim lngStatus As Long
Dim strConfiguration As String
Dim strSelectMsg  As String * 256
lngStatus = CR95HFDll_Select(strConfiguration,
strSelectMsg)
```

Input parameter:

strConfiguration: Selected RF communication protocol and certain protocol-related parameters. (This configuration is used for SendReceive requests.)

ISO15693 HighDataRate 10% One subcarrier = “010D”:
Where:
“01” is the ISO 15693 configuration
“0D” are the parameters

ISO14443A = "02000180":
Where:
“02” is the ISO 14443A configuration
“000180” are the parameters

Note: The CR95HF datasheet recommends to complete the ISO14443A protocol configuration:
- by optimizing the synchro between digital & analog inputs (sends CR95HFDLL_STCMD with "01 09043A005804")
- by modifying modulation & gain (send CR95HFDLL_STCMD with "01 0904680101D1").

ISO14443B = "03010180":
Where:
“03” is the ISO 14443B configuration
“010180” are the parameters

Note: The CR95HF datasheet recommends to complete the ISO144443B protocol configuration by modifying the ARC_B value to increase the Demodulator Gain (send CR95HFDLL_STCMD with "01 090468010130").

Output parameter:

strSelectMsg: The CR95HF demonstration board sends back an answer if the CR95HF is configured correctly and the RF field is on.
Answer example: “0000”

Where:
“00” is the status byte (see *Appendix A* for error codes)
“00” is the size of the answer

Returned value:

lngStatus:
0: No error
5: CR95HF demonstration board not connected
2: Empty argument error
3: Command parameter error

Source code example

```
 'Private Sub cmdSelect_ISO15693_protocol_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_Select("010D", strAnswer)

If (lngStatus = 0) Then
    txtISO15693_SelectAnswer.Text = strAnswer
Else
    txtISO15693_SelectAnswer.Text = " Select protocol ISO15693 not
executed"
End If

End Sub


Private Sub cmdSelect_ISO14443B_protocol_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_Select("03010180", strAnswer)

If (lngStatus = 0) Then
    txtISO14443B_SelectAnswer.Text = strAnswer
Else
    txtISO14443B_SelectAnswer.Text = "Select protocol ISO14443B not
executed"
End If

End Sub
```

C/C++ language

Declaration:

```
__declspec(dllimport) int  __stdcall
CR95HFDll_Select(char *StringCmd, char *Stringreply);
```

Prototype:

```
int iresult;
char strRequest[50]="";
char strAnswer[50]="";
iresult = CR95HFDll_Select(strRequest ,strAnswer);
```

Input parameter: strConfiguration: Selected RF communication protocol and certain protocol-related parameters. (This configuration is used for SendReceive requests.)

ISO15693 HighDataRate 10% One subcarrier = "010D":
Where:
“01” is the ISO 15693 configuration
“0D” are the parameters

ISO14443A = "02000180":
Where:
“02” is the ISO 14443A configuration
“00180” are the parameters

Note: TheCR95HF datasheet recommends to complete the ISO14443A protocol configuration:

- by optimizing the synchro between digital & analog inputs (sends CR95HFDLL_STCMD with "01 09043A005804")

- by modifying modulation & gain (send CR95HFDLL_STCMD with "01 0904680101D1")

ISO14443B = "03010180":
Where:
"03" is the ISO 14443B configuration
"010180" are the parameters

Note: The CR95HF datasheet recommends to complete the ISO144443B protocol configuration by modifying the ARC_B value to increase the Demodulator Gain (send CR95HFDLL_STCMD with "01 090468010130").

Output parameter: strSelectMsg: The CR95HF sends back an answer if the CR95HF demonstration board is configured correctly and the RF field is on.
Answer example: "0000"

Where:
"00" is the status byte (see *Appendix A* for error codes)
"00" is the size of the answer

Returned value: iresult:
0: No error
5: CR95HF demonstration board not connected
2: Empty argument error
3: Command parameter error

Source code example

```
int Select_ISO15693 (void)
{
  int iresult;
  char strRequest[50]="";
  char strAnswer[50]="";

  char entry3;

  printf("\n\n\n\n");

  strcpy(strRequest,"010D");

  iresult = CR95HFDll_Select(strRequest ,strAnswer);

  printf("\nProtocol select using CR95HFDll_Select function:\n");
  printf("\n    --> request : CR95HFDll_Select(%s, strAnswer)",
strRequest);
  printf("\n    <-- answer  : ");

  if (iresult == 0)
          printf("%s = ISO15693 protocol selected\n\n\r", strAnswer);
  else
          printf("%s = Select command error\n\n\r", strAnswer);
  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

  return iresult;
}

int Select_ISO14443B (void)
{
  char strRequest[50]="";
  char strAnswer[50]="";
  int iresult;
  char entry3;

  printf("\n\n\n\n");

  strcpy(strRequest,"03010180");

  iresult = CR95HFDll_Select(strRequest ,strAnswer);

  printf("\nISO14443B protocol select using CR95HFDll_Select
function:\n");
  printf("\n    --> request : CR95HFDll_Select(%s,
strAnswer)",strRequest);
  printf("\n    <-- answer  : ");

  if (iresult == 0)
          printf("%s = ISO14443B protocol selected\n\n\r", strAnswer);
  else
          printf("%s = Select command error\n\n\r", strAnswer);

  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

  return iresult;
}
```

2.3.3 CR95HFDll_STCmd

This function is used to send any request to the CR95HF IC. The STM32 MCU receives the frame contained in the request and sends it directly to the CR95HF IC. The STM32 MCU sends back the answer from the CR95HF IC to the PC through the USB port. The frame has to be formatted according to the CR95HF datasheet.

This function can be defined as a “Transparent Mode” command.

The CR95HF formatted frame has to be sent through this CR95HFDll_STCmd function preceded by “01” data as the header byte.

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDll_STCmd Lib "CR95HF.dll"
(ByVal mycmdstring As String, ByVal mystring As String)
As Long
```

Prototype:

```
Dim lngStatus As Long
Dim strMCUrequest As String
Dim strCR95HFanswer As String * 256
lngStatus = CR95HFDll_STCmd(strMCUrequest,
strCR95HFanswer)
```

Input parameter:

strMCUrequest: request to be sent by the STM32 MCU to the CR95HF IC, preceded by “01”.

Example 1: Send RdReg (Read Wakeup Register) command to read CR95HF wakeup flags: “010803620100”

Where:
- “01” is the transparent command header byte
- “0803620100” is the request to be sent to the CR95HF IC
  - “08” is the RdReg command
  - “03” is the request length
  - “62” is the wakeup register value
  - “01” is the register size
  - “00” ST reserved

Example 2: Select ISO 15693 protocol: “010202010D”

Where:
- “01” is the transparent command header byte
- “0202010D” is the Protocol Select ISO 15693 frame
  - “02” is the Protocol Select command
  - “02” is the request length
  - “010D” are the Protocol Select command parameters

Example 3: IDLE mode:
“01070E0B21007801180020606064743F01”

Where:
- “01” is the transparent command header byte
- “01070E0B21007801180020606064743F01” is the Idle frame for Tag Calibration

Output parameter:

strCR95HFanswer: Answer of the CR95HF IC

Example 1: Answer to RdReg command = “000102”

Where:
- “00” is the status byte (see *Appendix A* for error codes)
- “01” is the size of the answer
- “02” is the register value

Example 2: Answer to Select ISO 15693 protocol = “0000”

Where:
- “00” is the status byte (see *Appendix A* for error codes)
- “00” is the size of the answer

Example 3: Answer to Idle command = “0000”

Where:

"00" is the status byte (see *Appendix A* for error codes)
"00" is the size of the answer

Returned value: lngStatus:

0: No error
5: CR95HF demonstration board not connected
2: Empty argument error

Source code example

```
Private Sub cmdSTCmd_RdReg_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_STCmd("010803620100", strAnswer)

If (lngStatus = 0) Then
    txtSTCmd_RdReg.Text = strAnswer
Else
    txtSTCmd_RdReg.Text = "Read register not executed properly"
End If

End Sub


Private Sub cmdSTCmd_ISO15693_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_STCmd("010202010D", strAnswer)

If (lngStatus = 0) Then
    txtSTCmd_ISO15693.Text = strAnswer
Else
    txtSTCmd_ISO15693.Text = " ISO15693 protocol not selected"
End If

End Sub


Private Sub cmdSTCmd_Idle_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_STCmd("01070E0B21007801180020606064743F01",
strAnswer)

If (lngStatus = 0) Then
    txtSTCmd_Idle.Text = strAnswer
Else
    txtSTCmd_Idle.Text = " No Idn returned from CR95HF IC "
End If

End Sub
```

C/C++ language

Declaration:

```
__declspec(dllimport) int  __stdcall
CR95HFDll_STCmd(char *StringCmd, char *StringReply);
```

Prototype:

```
int iresult;
char strRequest[50]="";
char strAnswer[50]="";
iresult = CR95HFDll_STCmd(strRequest ,strAnswer);
```

Input parameter: strMCUrequest: Request to be sent by the STM32 MCU to the CR95HF IC following the Datasheet.

Example 1: Send RdReg (Read Wakeup Register) command to read CR95HF wakeup flags: “010803620100”

Where:
“01” is the transparent command header byte
“0803620100” is the request to be sent to the CR95HF IC
“08” is the RdReg command
“03” is the request length
“62” is the wakeup register value
“01” is the register size
“00” ST reserved

Example 2: Select ISO 15693 protocol: “010202010D”

Where:
“01” is the transparent command header byte
“0202010D” is the Protocol Select ISO 15693 frame
“02” is the Protocol Select command
“02” is the request length
“010D” are the Protocol Select command parameters

Example 3: IDLE mode:
“01070E0B21007801180020606064743F01”

Where:
“01” is the transparent command header byte
“01070E0B21007801180020606064743F01” is the Idle frame for Tag Calibration

Output parameter: strCR95HFanswer: Answer of the CR95HF IC

Example 1: Answer to RdReg command = “000102”

Where:
“00” is the status byte (see *Appendix A* for error codes)
“01” is the size of the answer
“02” is the register value

Example 2: Answer to Select ISO 15693 protocol = “0000”

Where:
“00” is the status byte (see *Appendix A* for error codes)
“00” is the size of the answer

Example 3: Answer to Idle command = “0000”

Where:
“00” is the status byte (see *Appendix A* for error codes)
“00” is the size of the answer

Returned value: iresult:
0: No error
5: CR95HF demonstration board not connected
2: Empty argument error

Source code example

```
int STCmd_RdReg (void)
{
  int iresult;
  char strRequest[50]="";
  char strAnswer[50]="";
  char entry3;

  printf("\n\n\n\n");

  strcpy(strRequest,"010803620100");

  iresult = CR95HFDll_STCmd(strRequest ,strAnswer);

  printf("\nRead register using CR95HFDll_STCmd function:\n");
  printf("\n    --> request : CR95HFDll_STCmd(%s,
strAnswer)",strRequest);
  printf("\n    <-- answer  : ");

  if (iresult == 0)
          printf("%s = Read Register answer\n\n\r", strAnswer);
  else
          printf("%s = No answer from CR95HF IC\n\n\r", strAnswer);

  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

 return iresult;
}



int STCmd_ISO15693 (void)
{
  char strRequest[50]="";
  char strAnswer[50]="";
  int iresult;
  char entry3;

  printf("\n\n\n\n");

  strcpy(strRequest,"010202010D");

  iresult = CR95HFDll_STCmd(strRequest ,strAnswer);
  printf("\nISO15693 protocol select using CR95HFDll_STCmd
function:\n");
  printf("\n    --> request : CR95HFDll_STCmd(%s,
strAnswer)",strRequest);
  printf("\n    <-- answer  : ");

  if (iresult == 0)
          printf("%s = ISO15693 protocol selected successfully\n\n\r",
strAnswer);
  else
          printf("%s = Protocol not selected\n\n\r", strAnswer);

  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

  return iresult;
}



int STCmd_Idle (void)
{
  char strRequest[50]="";
  char strAnswer[50]="";
  int iresult;
  char entry3;

  printf("\n\n\n\n");

  strcpy(strRequest,"01070E0B21007801180020606064743F01");

  iresult = CR95HFDll_STCmd(strRequest ,strAnswer);

  printf("\nIdle using CR95HFDll_STCmd function:\n");
  printf("\n    --> request : CR95HFDll_STCmd(%s,
strAnswer)",strRequest);
  printf("\n    <-- answer  : ");

  if (iresult == 0)
          printf("%s = Idle answer\n\n\r", strAnswer);
  else
          printf("%s = No answer from CR95HF IC\n\n\r", strAnswer);

  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

  return iresult;
}
```

2.3.4 CR95HFDll_FieldOff

This function sends a USB request to the STM32 MCU to switch off the CR95HF RF Field. The STM32 MCU sends back the answer of the CR95HF, if possible, or returns an error code ‘1’ if there is no answer.

Note: To perform this RF Field Off command, the STM32 MCU sends a Protocol Select command with parameters ‘0000’. For more information, refer to the CR95HF transceiver datasheet.

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDll_FieldOff Lib
"CR95HF.dll" (ByVal mystring As String) As Long
```

Prototype:

```
Dim lngStatus As Long
Dim strAnswer  As String * 256
status = CR95HFDll_FieldOff(strAnswer)
```

Input parameter: None

Output parameter: strAnswer: The CR95HF demonstration board sends back an answer and the RF Field is switched off.

Answer example: “0000”
Where:
“00” is the status byte (see *Appendix A* for error codes)
“00” is the size of the answer

Returned value: lngStatus:
0: No error
5: CR95HF demonstration board not connected

Source code example

```
Private Sub cmdFieldOff_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_FieldOff(strAnswer)

If (lngStatus = 0) Then
    txtFieldOffAnswer.Text = strAnswer
Else
    txtFieldOffAnswer.Text = "Field Off request not executed"
End If

End Sub
```

C/C++ language

Declaration:

```
__declspec(dllimport) int  __stdcall
CR95HFDll_FieldOff(char *StringReply);
```

Prototype:

```
int iresult;
char strAnswer[50]="";
iresult= CR95HFDll_FieldOff (strAnswer);
```

Input parameter: None

Output parameter: strAnswer: The CR95HF demonstration board sends back an answer and the RF Field is switched off.

Answer example: “0000”
Where:
“00” is the status byte (see *Appendix A* for error codes)
“00” is the size of the answer

Returned value: iresult:

0: No error
5: CR95HF demonstration board not connected

Source code example

```
int FieldOff (void)
{
  int iresult;
  char strAnswer[50]="";
  char entry3;

  printf("\n\n\n\n");

  iresult= CR95HFDll_FieldOff (strAnswer);

  printf("\nField Off using CR95HFDll_FieldOff function:\n");
  printf("\n    --> request : CR95HFDll_FieldOff(strAnswer)");
  printf("\n    <-- answer  : ");

  if (iresult == 0)
          printf("%s = RF Field Off ok\n\n\r", strAnswer);
  else
          printf("%s = Error : RF Field Off command error\n\n\r", 
strAnswer);

  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

  return iresult;
}
```

2.3.5 CR95HFDll_ResetSPI

This function resets the CR95HF IC in case of a problem. This function only resets the CR95HF IC and not the STM32 MCU.

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDll_ResetSPI Lib 
"CR95HF.dll" (ByVal mystring As String) As Long
```

Prototype:

```
Dim lngStatus As Long
Dim strAnswer  As String * 256
lngStatus = CR95HFDll_ResetSPI(strAnswer)
```

Input parameter: None

Output parameter: strAnswer: The CR95HF demonstration board sends back an answer if the SPI has been correctly reset.

Answer example: “0000”

Where:

“00” is the status byte (see *Appendix A* for error codes)
“00” is the size of the answer

Returned value: lngStatus:

0: No error
5: CR95HF demonstration board not connected

Source code example

```
Private Sub cmdResetSPI_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_ResetSPI(strAnswer)

If (lngStatus = 0) Then
    txtResetSPIAnswer.Text = strAnswer
Else
    txtResetSPIAnswer.Text = "No answer from the DEMO-CR95HF-A board"
End If

End Sub
```

C/C++ language

Declaration:

```
__declspec(dllexport) int  __stdcall
CR95HFDll_ResetSPI(char* stringreply);
```

Prototype:

```
int iresult;
char strAnswer[50]="";
iresult = CR95HFDll_ResetSPI(strAnswer);
```

Input parameter: None

Output parameter: strAnswer: The CR95HF demonstration board sends back an answer if the SPI has been correctly reset.

Answer example: “0000”

Where:

“00” is the status byte (see *Appendix A* for error codes)
“00” is the size of the answer

Returned value: iresult:

0: No error
5: CR95HF demonstration board not connected

Source code example

```
int ResetSPI (void)
{
  int iresult;
  char strAnswer[50]="";
  char entry3;

  printf("\n\n\n\n");

  iresult = CR95HFDll_ResetSPI(strAnswer);

  printf("\nReset SPI using CR95HFDll_ResetSPI function:\n");
  printf("\n    --> request : CR95HFDll_ResetSPI(strAnswer)");
  printf("\n    <-- answer  : ");

  if (iresult == 0)
```

```
            printf("%s = Reset SPI successfully\n\n\r", strAnswer);
    else
            printf("%s = No answer from CR95HF demonstration 
board\n\n\r", strAnswer);

    printf("\n\n\n\n");
    printf("\npress any key to continue");
    printf("\n");
    scanf("%s", &entry3);

    return iresult;
}
```

2.3.6 CR95HFDll_SendIRQPulse

This function must be used when the CR95HF demonstration board is configured in SPI mode (communication between the STM32 MCU and the CR95HF IC). The interrupt pulse is sent to the CR95HF IRQ pin.

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDll_SendIRQPulse Lib
"CR95HF.dll" (ByVal mystring As String) As Long
```

Prototype:

```
Dim lngStatus As Long
Dim strAnswer  As String * 256
lngStatus = CR95HFDll_SendIRQPulse(strAnswer)
```

Input parameter: None

Output parameter: strAnswer: The CR95HF demonstration board sends back an answer if the IRQ pulse is correctly sent.

Answer example: “0000”

Where:

“00” is the status byte (see *Appendix A* for error codes)
“00” is the size of the answer

Returned value: lngStatus:

0: No error
5: CR95HF demonstration board not connected

Source code example

```
Private Sub cmdSendIRQPulse_Click()
Dim strAnswer  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_SendIRQPulse(strAnswer)

If (lngStatus = 0) Then
    txtSendIRQPulseAnswer.Text = strAnswer
Else
    txtSendIRQPulseAnswer.Text = "No answer from the CR95HF 
demonstration board"
End If

End Sub
```

C/C++ language

Declaration: `__declspec(dllexport) int __stdcall CR95HFDll_SendIRQPulse(char* stringreply);`

Prototype:

```
int iresult;
char strAnswer[50]="";
iresult = CR95HFDll_SendIRQPulse(strAnswer);
```

Input parameter: None

Output parameter: strAnswer: The CR95HF demonstration board sends back an answer if the IRQ Pulse is correctly sent.

Answer example: “0000”

Where:
“00” is the status byte (see *Appendix A* for error codes)
“00” is the size of the answer

Returned value: iresult:
0: No error
5: CR95HF demonstration board not connected

Source code example

```
int Send_IRQ_Pulse (void)
{
  int iresult;
  char strAnswer[50]="";
  char entry3;

  printf("\n\n\n\n");

  //iresult = CR95HFDll_SendIRQPulse(strAnswer);

  printf("\nSend IRQ Pulse using CR95HFDll_SendIRQPulse function:\n");
  printf("\n    --> request : CR95HFDll_SendIRQPulse(strAnswer)");
  printf("\n    <-- answer  : ");

  if (iresult == 0)
          printf("%s = Send IRQ pulse successfully\n\n\r", strAnswer);
  else
          printf("%s = No answer from CR95HF demonstration 
board\n\n\r", strAnswer);

  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

  return iresult;
}
```

2.3.7 CR95HFDll_Polling_Reading

This function polls the CR95HF SPI communication wires. When the CR95HF is ready, it sends back the data stored in its FIFO answer buffer.

This function can be used after an IDLE command to make sure that the answer buffer has been emptied and avoid receiving an erroneous answer on the next request.

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDll_Polling_Reading Lib
"CR95HF.dll" (ByVal mystring As String) As Long
```

Prototype:

```
Dim strAnswer As String * 256
Dim lngStatus As Long
lngStatus = CR95HFDll_Polling_Reading(strAnswer)
```

Input parameter: None

Output parameter: CR95HF answer stored in FIFO answer buffer.

Returned value: lngStatus:

0: No error

5: CR95HF demonstration board not connected

Source code example

```
Private Sub cmdPollingReading_Click()
Dim strAnswer As String * 256
Dim lngStatus As Long
lngStatus = CR95HFDll_Polling_Reading(strAnswer)
If (lngStatus = 0) Then
    txtPollingReadingAnswer.Text = strAnswer
Else
    txtPollingReadingAnswer.Text = "No answer from the
    CR95HF demonstration board"
End If
End Sub
```

C/C++ language

Declaration:

```
__declspec(dllexport) int  __stdcall
CR95HFDll_Polling_Reading(char* StringReply);
```

Prototype:

```
int iresult;
char strAnswer[50]="";
iresult = CR95HFDll_Polling_Reading(strAnswer);
```

Input parameter: None

Output parameter: CR95HF answer stored in FIFO answer buffer.

Returned value: iresult:

0: No error
5: CR95HF demonstration board not connected

Source code example

```
int Set_Polling_Reading_config (void)
{
  int iresult;
  char strAnswer[50]="";
  char entry3;
  printf("\n\n\n\n");
  iresult = CR95HFDll_Polling_Reading(strAnswer);
  printf("\nSend IRQ Pulse using CR95HFDll_Polling_Reading
        function:\n");
        printf("\n    --> request : CR95HFDll_Polling_Reading(strAnswer)");
        printf("\n    <-- answer  : ");
        if (iresult == 0)
           printf("%s = Set Polling Reading successfully\n\n\r", strAnswer);
        else
           printf("%s = No answer from CR95HF demonstration board\n\n\r",
strAnswer);
        printf("\n\n\n\n");
        printf("\npress any key to continue");
        printf("\n");
        scanf("%s", &entry3);
        return iresult;
      }
```

2.3.8 CR95HFDll_SendNSSPulse

This function sends an interrupt to wake up the CR95HF IC. It can be used when the CR95HF is configured in UART mode (communication between the STM32 MCU and the CR95HF IC). The interrupt pulse is sent to the CR95HF NSS pin.

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDll_SendNSSPulse Lib
"CR95HF.dll" (ByVal mystring As String) As Long
```

Prototype:

```
Dim strAnswer  As String * 256
Dim lngStatus As Long
lngStatus = CR95HFDll_SendNSSPulse(strAnswer)
```

Input parameter: None

Output parameter: strAnswer. The CR95HF demonstration board sends back an answer if the NSS pulse is correctly sent.

Answer example: “0000”

Where:
“00” is the status byte (see Appendix A for error codes)
“00” is the size of the answer

Returned value: lngStatus:

0: No error
5: CR95HF demonstration board not connected

Source code example

```
    Private Sub cmdSendNSSPulse_Click()
    Dim strAnswer As String * 256
    Dim lngStatus As Long
    lngStatus = CR95HFDll_SendNSSPulse(strAnswer)
    If (lngStatus = 0) Then
        txtNSSpulseAnswer.Text = strAnswer
    Else
        txtNSSpulseAnswer.Text = "No answer from the CR95HF demonstration
board"
    End If
    End Sub
```

C/C++ language

Declaration:

```
__declspec(dllexport) int  __stdcall
CR95HFDll_SendNSSPulse(char* StringReply);
```

Prototype:

```
int iresult;
char strAnswer[50]="";
iresult = CR95HFDll_SendNSSPulse(strAnswer);
```

Input parameter: None

Output parameter: strAnswer: The CR95HFdemonstration board sends back an answer if the NSS pulse is correctly sent.
Example of answer: “0000”
where
“00” is the status byte (see Appendix A for error codes)
“00” is the size of the answer

Returned value: iresult:

0: No error
5: CR95HF demonstration board not connected

Source code example

```
    int Send_NSS_Pulse (void)
    {
      int iresult;
      char strAnswer[50]="";
      char entry3;
      printf("\n\n\n\n");
      iresult = CR95HFDll_SendNSSPulse(strAnswer);
      printf("\nSend IRQ Pulse using CR95HFDll_SendNSSPulse function:\n");
      printf("\n    --> request : CR95HFDll_SendNSSPulse(strAnswer)");
      printf("\n    <-- answer  : ");
      if (iresult == 0)
              printf("%s = Send NSS pulse successfully\n\n\r", strAnswer);
      else
              printf("%s = No answer from CR95HF demonstration
board\n\n\r", strAnswer);
      printf("\n\n\n\n");
      printf("\npress any key to continue");
      printf("\n");
      scanf("%s", &entry3);
      return iresult;
    }
```

2.4 Functions to communicate with an RF tag

2.4.1 CR95HFDll_SendReceive

This function sends a USB request to the STM32 MCU that executes a SendRecv command with data to the CR95HF IC. The STM32 MCU sends back the answer of the CR95HF, if possible, or returns an error code ‘1’ if there is no answer.

Note: *The request uses the SendRecv command to send data using previously selected protocol and to receive the tag response. For more information, refer to the CR95HF transceiver datasheet.*

Visual Basic language

Declaration:

```
Public Declare Function CR95HFDll_SendReceive Lib
"CR95HF.dll" (ByVal mycmdstring As String, ByVal
mystring As String) As Long
```

Prototype:

```
Dim lngStatus As Long
Dim strCR95HFrequest As String
Dim strTagResponse  As String * 256
lngStatus = CR95HFDll_SendReceive(strCR95HFrequest,
strTagResponse  )
```

Input parameter: strCR95HFrequest: The RF Request to be sent by the CR95HF IC to the Tag (with previously selected ISO format).

ISO 15693 Inventory example: “260100”

Where:
“260100” is the ISO 15693 Inventory command.

ISO 14443B Initiate example: "0600"

Where:
“0600” is the ISO 14443B Initiate command.

Output parameter: strTagResponse: The Tag answer if the CR95HF has received an answer from the Tag in the field; otherwise, an error code (see *Appendix A*).

ISO 15693 Inventory example:
“800D00FF5AB914F6414C02E0385D00”

Where:
“80” is the status byte (see *Appendix A* for error codes)
“0D” is the length of entire data field
“00FF5AB914F6414C02E0” is the data received from the tag
“385D” is the original received CRC value
“00” is the protocol error status

ISO 14443B Initiate example: “80046C125900”

Where:
“80” is the status byte (see *Appendix A* for error codes)
“04” is the length of the entire data field
“6C” is the data received from the tag

“1259” is the original received CRC value
“00” is the protocol error status

Returned value: lngStatus:
0: No error
5: CR95HF demonstration board not connected
4: Communication error

Source code example

```
Private Sub cmdSendReceive_ISO15693_Inventory_Click()
Dim strTagResponse  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_SendReceive("260100", strTagResponse)

If (lngStatus = 0) Then
    txtISO15693_Inventory_Answer.Text = strTagResponse
Else
    txtISO15693_Inventory_Answer.Text = "No answer from the Tag"
End If

End Sub


Private Sub cmdSendReceive_ISO14443B_Initiate_Click()
Dim strTagResponse  As String * 256
Dim lngStatus As Long

lngStatus = CR95HFDll_SendReceive("0600", strTagResponse)

If (lngStatus = 0) Then
    txtISO14443B_Initiate_Answer.Text = strTagResponse
Else
    txtISO14443B_Initiate_Answer.Text = "No answer from the Tag"
End If

End Sub
```

C/C++ language

Declaration:

```
__declspec(dllimport) int  __stdcall
CR95HFDll_SendReceive(char *StringCmd, char
*StringReply);
```

Prototype:

```
int iresult;
char strRequest[50]="";
char strTagAnswer[50]="";
iresult=CR95HFDll_SendReceive(strRequest,strTagAnswer);
```

Input parameter: strCR95HFrequest: The RF Request to be sent by the CR95HF IC to the Tag (with previously selected ISO format).

ISO 15693 Inventory example: “260100”

Where:
“260100” is the ISO 15693 Inventory command.

ISO 14443B Initiate example: "0600"

Where:
"0600" is the ISO 14443B Initiate command.

Output parameter: strTagResponse: The Tag answer if the CR95HF has received an answer from the Tag in the field; otherwise, an error code (see *Appendix A*).

ISO 15693 Inventory example: "800D00FF5AB914F6414C02E0385D00"

Where:
"80" is the status byte (see *Appendix A* for error codes)
"0D" is the length of the entire data field
"00FF5AB914F6414C02E0" is the data received from the tag
"385D" is the original received CRC value
"00" is the protocol error status

ISO 14443B Initiate example: "80046C125900"

Where:
"80" is the status byte (see *Appendix A* for error codes)
"04" is the length of the entire data field
"6C" is the data received from the tag
"1259" is the original received CRC value
"00" is the protocol error status

Returned value: iresult:
0: No error
5: CR95HF demonstration board not connected
4: Communication error

Source code example

```
int Send_ISO15693_Inventory (void)
{
  int iresult;
  char strRequest[50]="";
  char strTagAnswer[50]="";
  char entry3;

  printf("\n\n\n\n");

  strcpy(strRequest,"260100");

  iresult=CR95HFDll_SendReceive(strRequest,strTagAnswer);

  printf("\nISO15695 inventory using CR95HFDll_SendReceive
function:\n");
  printf("\n    --> request : CR95HFDll_SendReceive(%s,
strTagAnswer)",strRequest);
  printf("\n    <-- answer  : ");

  if ((strTagAnswer[0] == '8') & (strTagAnswer[1] == '0')) //CR95HF Tag
answer OK
  {
     printf("%s = Tag answer",strTagAnswer);
  }
  else
  {
     printf("%s = No tag answer received\n");
  }

  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

  return iresult;
}


int Send_ISO14443B_Initiate (void)
{
  char strRequest[50]="";
  char strTagAnswer[50]="";
  int iresult;
  char entry3;

  printf("\n\n\n\n");

  strcpy(strRequest,"0600");

  iresult=CR95HFDll_SendReceive(strRequest,strTagAnswer);

  printf("\nISO14443B Initiate using CR95HFDll_SendReceive
function:\n");
  printf("\n    --> request : CR95HFDll_SendReceive(%s,
strTagAnswer)",strRequest);
  printf("\n    <-- answer  : ");

  if ((strTagAnswer[0] == '8') & (strTagAnswer[1] == '0')) //CR95HF Tag
answer OK
  {
     printf("%s = Tag answer",strTagAnswer);
  }
  else
  {
     printf("No tag answer received\n");
     printf("CR95HF answer : %s",strTagAnswer);
  }
  printf("\n\n\n\n");
  printf("\npress any key to continue");
  printf("\n");
  scanf("%s", &entry3);

  return iresult;
}
```

2.4.2 SendReceive Request and Response maximum data size

Since DLL revision 1.2 and CR95HF demonstration board firmware revision 3.6.6 (STSW-M24LR007), the CR95HFDll_SendReceive function can send requests up to 256 bytes and receive responses up to 528 bytes. This is internally managed by the CR95HF IC using Long frames (refer to the dedicated section in the CR95HF datasheet).

The management of HID USB frames is limited to 64 bytes. The communication between the computer and the CR95HF demonstration board is managed jointly by HID USB, while DLL and the STM32 microcontroller manage long frames.

Each request frame coming from the PC software is cut in several 64 bytes parts by the CR95HFDll then sent to the CR95HF demonstration board through the HID USB. The STM32 microcontroller then restores the initial frame and send the request to the CR95HF.

The same is done for the response and management by STM32 firmware and DLL.

*Note: strRequest[50] and strTagAnswer[50] / strTagResponse as string * 256 variables must be changed, respectively, with strRequest[1024] and strTagAnswer[1024] / strTagResponse as String * 1024.*

The CR95HF can also manage long frames on requests to increase Request maximum size up to 512 bytes (as Long frame support). This requires a patch to be sent to the device, hence a new function has been added to the firmware. To activate this mode, user has to send "02 A8" hexadecimal code using CR95HFDll_STCmd function. The expected response to this function is "00 02 03 97".

Visual Basic language

```
Private Sub SetCR95HFLongFrame_Click()
Dim strAnswer As String * 1204
Dim lngStatus As Long
lngStatus = CR95HFDll_STCmd("02A8", strAnswer)
End Sub
```

C/C++ language

```
int SetCR95HFLongFrame (void)
{
int iresult;
char strRequest[1024]="";
char strAnswer[1024]="";
strcpy(strRequest,"02A8");
iresult = CR95HFDll_STCmd(strRequest ,strAnswer);
return iresult;
}
```

Appendix A Error codes

Table 1. List of error codes

Error code	Description
0000	Answer OK
8000	Answer OK
8200	Invalid command length
8300	Invalid protocol
8600	Communication error
8700	Frame wait time out OR no tag
8800	Invalid Start Of Frame
8900	Receive buffer overflow (too many bytes received)
8A00	Framing error (start bit = 0, stop bit = 1)
8B00	EGT time out (for ISOIEC 14443-B)
8C00	Invalid length. Used in Felica, when field length < 3
8D00	CRC error (Used in Felica protocol)
8E00	Reception lost without EOF received
8F00	No field
FD00	Time out - no answer from Tag detected by the CR95HF IC
FE00	Unknown error

Revision history

Table 2. Document revision history

Date	Revision	Changes
25-Jul-2011	1	Initial release.
16-Dec-2011	2	Added functions available in new DLL release revision 0.5: *Section 2.1.2: CR95HFDLL_USBconnect*, *Section 2.1.3: CR95HFDLL_USBhandlecheck*, *Section 2.3.7: CR95HFDll_Polling_Reading*, and *Section 2.3.8: CR95HFDll_SendNSSPulse*. Updated returned value in C/C++ for *Section 2.1.1: CR95HFDll_GetDLLrev*, *Section 2.2.2: CR95HFDLL_getMCUrev*, *Section 2.2.3: CR95HFDLL_getHardwareVersion*, *Section 2.3.1: CR95HFDll_Idn*, *Section 2.3.2: CR95HFDll_Select*, *Section 2.3.3: CR95HFDll_STCmd*, *Section 2.3.4: CR95HFDll_FieldOff*, *Section 2.3.5: CR95HFDll_ResetSPI*, *Section 2.3.6: CR95HFDll_SendIRQPulse*, *Section 2.4.1: CR95HFDll_SendReceive*. Updated returned value in Visual basic and C/C++ for *Section 2.2.1: CR95HFDll_Echo*.
09-Oct-2012	3	Added text to *Section 1: Getting started*. Added *Section 2.2.3: CR95HFDLL_getHardwareVersion*. Updated Input parameter (in *Section 2.3.2* and *Section 2.3.3*), Output parameter (in *Section 2.3.1*), Returned value (in *Section 2.1.3* and *Section 2.2.2*) and some values in Source code example of *Section 2.3.2* and *Section 2.3.3*. Added ‘8F00’ to *Table 1: List of error codes*. Fixed typos.
18-Jan-2017	4	Updated document title, *Introduction* and figure on cover page. Replaced DEMO-CR95HF-A demonstration board with CR95HF demonstration board across the whole document, including figures. Updated *Section 1: Getting started* and captions of figures *2* to *6*. Added *Section 2.4.2: SendReceive Request and Response maximum data size*.