



# **RD200 Serial USB RFID Reader Example Application Guide (For Visual Basic)**

**Document Version**

**0100**

**30 July 2009**

**Confidential**

The contents of this document are confidential and proprietary to SYRIS Technology Corp. and may not be reproduced, transmitted, published, or disclosed to others without prior authorization.

# Contents

0 Revision History .....	2
1 Introduction .....	3
2 File Needed .....	3
3 How to Use EasyPOD DLL .....	3
4 How to use the internal built function .....	4
4.1. Connect .....	4
4.2. Disconnect .....	4
4.3. WriteData .....	4
4.4. ReadData .....	5
4.5. ClearPODBuffer .....	5
5 Example Code .....	6

SYRIS CONFIDENTIAL

## 0 Revision History

Date	Version	Description
30-Jul-09	V01.00	Initial version

SYRIS CONFIDENTIAL

# 1 Introduction

This document provides a simply example application to describes how to use EasyPOD.DLL in the Visual Basic .NET environment for RD200 serial USB RFID Reader data transmission and reception.

## 2 File Needed

1. EasyPOD.dll
2. EasyPOD.vb

## 3 How to Use EasyPOD DLL

Install Visual Basic .net and use it to develop the application program, please follow the steps as below:

### **Debug Mode:**

1. Copy "EasyPOD.dll" to windows installation path. For example: if your operation system is Windows XP and use default installation path, please copy the file to "C:\Windows directory".
2. Create a new Project in Visual Basic, then go to the "Project" menu then choose "Add Module" to add the "EasyPOD.vb" into this new Project.

### **Execution Mode:**

1. Copy the "EasyPOD.dll" into the same directory where the execution file is located.

## 4 How to use the internal built function

The DLL functions are described as below:

### 4.1. Connect

<b>Function</b>	Public Declare Function ConnectPOD Lib "EasyPOD.DLL" (ByRef m_EasyPod As MW_EasyPOD, ByVal Index As Long) As Long
<b>Return</b>	ERROR_SUCCESS(0) means connection is successful otherwise connection fails.
<b>Parameter</b>	m_EasyPod - the reference of struct MW_EasyPOD. Index - device index of EasyPOD, Index start at 1.

### 4.2. Disconnect

<b>Function</b>	Public Declare Function DisconnectPOD Lib "EasyPOD.DLL" (ByRef m_EasyPod As MW_EasyPOD) As Long
<b>Return</b>	ERROR_SUCCESS(0) means disconnection is successful otherwise disconnection fails.
<b>Parameter</b>	m_EasyPod - the reference of struct MW_EasyPOD.

### 4.3. WriteData

<b>Function</b>	Public Declare Function WriteData Lib "EasyPOD.DLL" (ByRef m_EasyPod As MW_EasyPOD, lpString As Byte, ByVal lToWrite As Long, lWritten As Long) As Long
<b>Return</b>	ERROR_SUCCESS(0) means writing is successful otherwise writing fails.
<b>Parameter</b>	m_EasyPod - the reference of struct MW_EasyPOD. lpString - pointer to the buffer containing the data to write to the device. lToWrite - number of Bytes to write. lWritten - actual number of bytes written.

## 4.4. ReadData

<b>Function</b>	Public Declare Function ReadData Lib "EasyPOD.DLL" (ByRef m_EasyPod As MW_EasyPOD, lpString As Byte, ByVal lToRead As Long, lRead As Long) As Long
<b>Return</b>	ERROR_SUCCESS(0) means reading is successful otherwise reading fails.
<b>Parameter</b>	m_EasyPod, the reference of struct MW_EasyPOD. lpString, pointer to the buffer containing the data to read from the device. lToRead, number of bytes to read. lRead, actual number of bytes read.

## 4.5. ClearPODBuffer

<b>Function</b>	Public Declare Function ClearPODBuffer Lib "EasyPOD.DLL" (ByRef m_EasyPod As MW_EasyPOD) As Long
<b>Return</b>	ERROR_SUCCESS(0) means buffer clearing is successful otherwise the clearing fails.
<b>Parameter</b>	m_EasyPod, the reference of struct MW_EasyPOD.

## 5 Example Code

Following example code will tell you how to send the request command to get the response data from the reader.

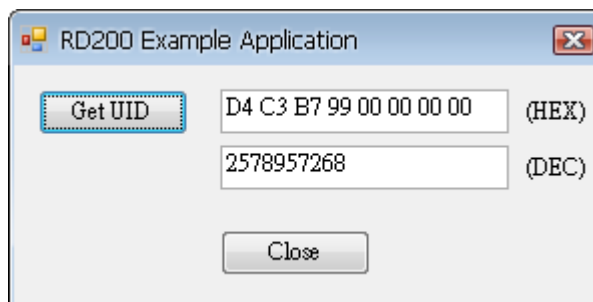


Figure 1-Example Application Form

### Module:

```

Module EasyPOD
    Public Declare Function ConnectPOD Lib "EasyPOD.DLL" (ByRef m_EasyPod As MW_EasyPOD,
        ByVal Index As Integer) As Integer
    Public Declare Function DisconnectPOD Lib "EasyPOD.DLL" (ByRef m_EasyPod As MW_EasyPOD)
As Integer
    Public Declare Function WriteData Lib "EasyPOD.DLL" (ByRef m_EasyPod As MW_EasyPOD,
        ByRef lpString As Byte, ByVal lToWrite As Integer, ByRef lWritten As Integer) As Integer
    Public Declare Function ReadData Lib "EasyPOD.DLL" (ByRef m_EasyPod As MW_EasyPOD, ByRef
        lpString As Byte, ByVal lToRead As Integer, ByRef lRead As Integer) As Integer
    Public Declare Function ClearPODBuffer Lib "EasyPOD.DLL" (ByRef m_EasyPod As MW_EasyPOD)
As Integer

    Public Structure MW_EasyPOD
        Dim VID As Integer
        Dim PID As Integer
        Dim ReadTimeOut As Integer
        Dim WriteTimeOut As Integer
        Dim Handle As Integer ' Do not modify this value, reserved for DLL
        Dim FeatureReportSize As Integer ' Do not modify this value, reserved for DLL
        Dim InputReportSize As Integer ' Do not modify this value, reserved for DLL
        Dim OutputReportSize As Integer ' Do not modify this value, reserved for DLL
    End Structure

    Function ByteToHex(ByVal sInput As Array)
        Dim s As String
        s = ""
        For i = 0 To 7
            s = s + Hex(sInput(i + 4)).PadLeft(2, Chr(&H30)) + " "
        Next i
        Return s
    End Function
End Module

```

**Main Form:**

Please refer to RD200 Protocol Manual for more detail about the request command.

```

Public Class frmMain

    Dim m_Pod As MW_EasyPOD
    Dim lResult As Long
    Dim lReturn As Long

    Private Sub btnGetUID_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnGetUID.Click

        Dim sRequest As Byte() = {&H2, &H2, &H1, &H1}      '{STX, LEN, CMD, DATA1, DATA2.....}'
        Dim sResponse() As Byte
        ReDim sResponse(20)
        Dim reqLen As Integer = sRequest.Length

        m_Pod.VID = &HE6A
        m_Pod.PID = &H317

        lResult = ConnectPOD(m_Pod, 1)
        lResult = ClearPODBuffer(m_Pod)

        If (lResult <> 0) Then 'Check the connectioin status
            MsgBox("Not connected yet", 16)
        Else
            m_Pod.ReadTimeout = 200
            m_Pod.WriteTimeout = 200

            'Send a request command to reader
            lResult = WriteData(m_Pod, sRequest(0), CLng(reqLen), lReturn)
            'Read the response data from reader
            lResult = ReadData(m_Pod, sResponse(0), 20, lReturn)

            txbGetUIDh.Text = ByteToHex(sResponse)           'Convert byte to HEX
            txbGetUIDd.Text = BitConverter.ToUInt32(sResponse, 4) 'Convert byte to DEC

        End If
        lReturn = DisconnectPOD(m_Pod)

    End Sub

    Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnClose.Click
        Close()
    End Sub

End Class

```