

# Лекція 13.

**Тема. Канали передачі та прийому інформації.**

**Розділ. Приклад реалізації розподіленої системи вимірювання.**

**План лекції:**

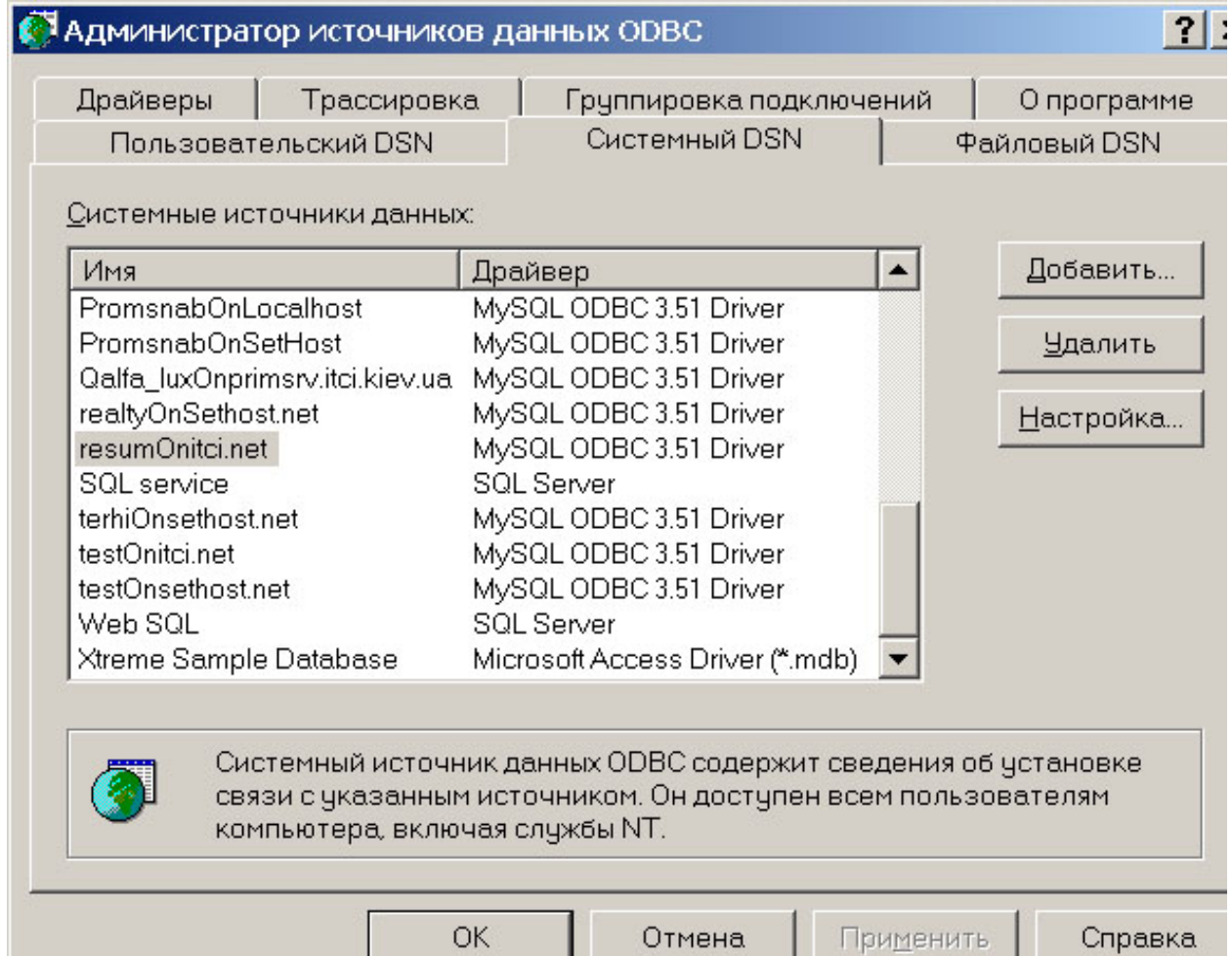
13.1. Система управління базою даних – універсальний інструмент з'єднання складових частин розподіленої системи вимірювання.

13.1.1. ODBC. Авторизація. DSN.

13.1.2. Вибір DSN та з'єднання.

13.1.3. Програмна реалізація протокольних-інтерфейсних функцій.

## Список ODBC (*Open Database Connectivity*) DSN (*Data Source Name*) джерел.



## Account mySQL

Connector/ODBC 3.51.12 - Configure Data Source Name

### Connector/ODBC

MySQL

Login | Connect Options | Advanced

Data Source Name: resumOnitci.net

Description: resum On itci.net

Server: itci.net

User: sergey

Password: \*

Database: resum

#### Connector/ODBC Configuration

This dialog is used to edit a Data Source Name (DSN).

Test | Diagnostics >> | Ok | Cancel | Help

## Опції mySQL

Connector/ODBC 3.51.12 - Configure Data Source Name

Connector/ODBC

MySQL

Login **Connect Options** Advanced

Port

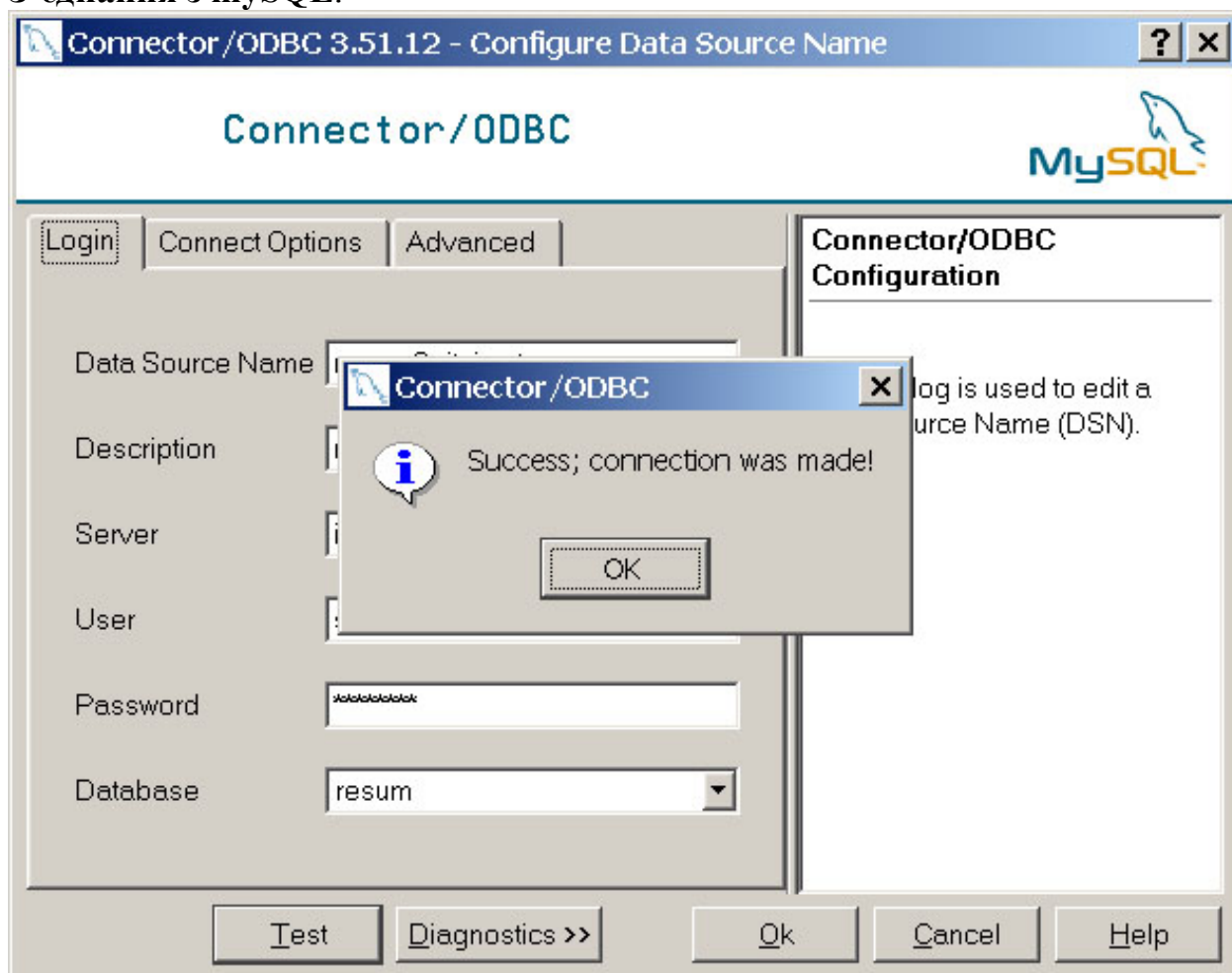
Socket

Initial Statement

**Connector/ODBC Configuration**

This dialog is used to edit a Data Source Name (DSN).

### З'єднання з MySQL.



## Зовнішній інтерфейс 1.

Диспетчер сервиса, samj

Файл Правка Сервис Просмотр/Изменение Управление Помощь

Авто  Удалять после  Сохранять события

13/12/12 10:07:10

Код стр чтения Win 1251  по умолчанию  насильно

Win 1251 koï8-r

! Своб Опл Пров Требования  
пров, опл, возвр, заявки

Аренда зданий	1437	0	0	0, 0, 0, 0
Продажа зданий	4036	0	0	0, 0, 0, 0
Аренда складов	4786	0	0	0, 0, 0, 0
Продажа складов	1930	0	0	0, 0, 0, 0
Аренда квартир	92933	1	0	0, 0, 0, 0
Продажа квартир	29268	2	2	0, 0, 0, 0
Аренда домов, дач	9373	0	0	0, 0, 0, 0
Продажа домов, дач	17319	0	0	0, 0, 0, 0
Аренда магазинов	6120	0	0	0, 0, 0, 0
Продажа магазинов	4700	0	0	0, 0, 0, 0
Аренда офисов	21102	0	0	0, 0, 0, 0
Продажа офисов	18470	0	0	0, 0, 0, 0
Аренда рест-в, кафе	1271	0	0	0, 0, 0, 0
Продажа рест-в, кафе	1537	0	0	0, 0, 0, 0
Аренда других	3304	0	0	0, 0, 0, 0
Продажа др помещей	4062	0	0	0, 0, 0, 0

Процесс:

SMTP: Имя сервера Кол-во  
mail.voliaca

Текущее

POP3: Имя сервера Кол-во  
pop3.sethost.net

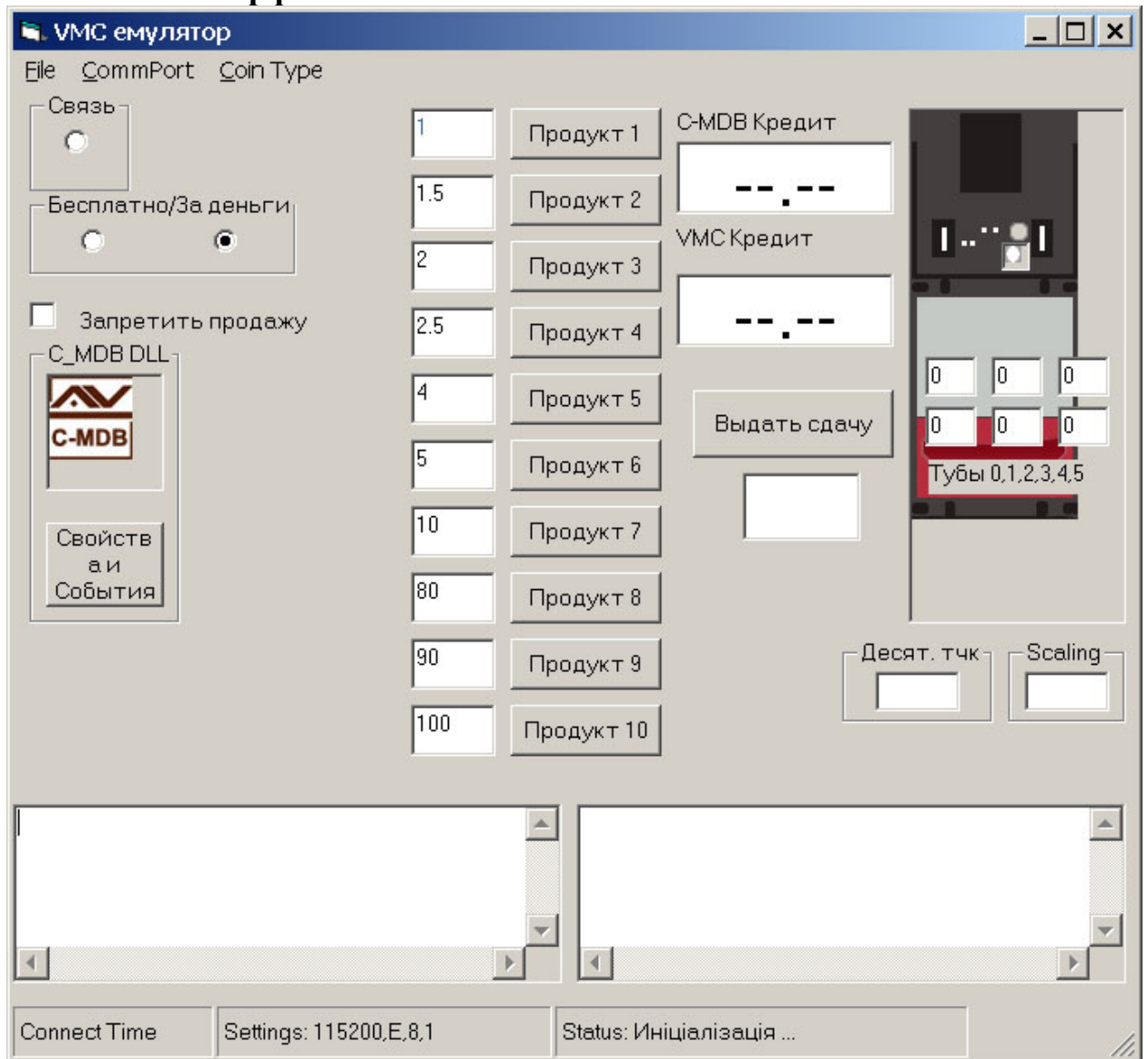
Имя польз-ля Текущее  
bpt

Тема

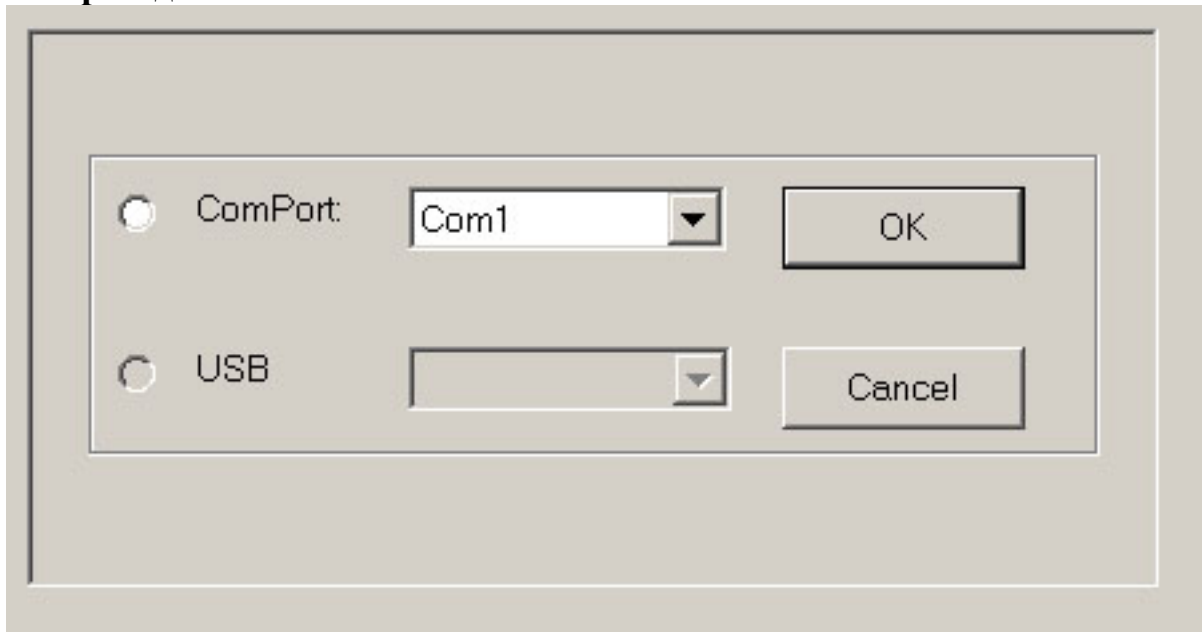
Содержание

Заголовок

## Зовнішній інтерфейс 2



### Вибір з'єднання.



### Доступ до MySQL.

*Public Sub ConnectMySQL(myblConnect As Boolean, strDSN As String, strUser As String, strPassword As String, strDatabase As String, Host As String)*

*Dim objCom As Command*

*Dim strcon As String*

*Dim k As Integer*

*'Dim strDSN As String*

*'Dim strUser As String*

*'Dim strPassword As String*

*'Dim strDatabase As String*

*myblConnect = False*

*' strDSN = frmRemoteSite.txtDSN*

*' strUser = frmRemoteSite.txtUser*

*' strPassword = frmRemoteSite.txtPassword2*

*' strDatabase = frmRemoteSite.txtDatabase*

*'myDatabase.Connect =*

*"ODBC;DSN=MySQL;UID=samj;PWD=juravv;DATABASE=resum"*

*' glbmyCon.ConnectionString = "DSN=" & strDSN & ";UID=" & strUser & ";PWD=" & strPassword & ";"*

*On Error GoTo ErrorHandler*

*Set objCom = CreateObject("ADODB.Command")*



```

objCom.CommandType = adCmdText

If Host = "remote" Then
    Set glbmyCon = New ADODB.Connection
    glbmyCon.ConnectionString = "DSN=" & strDSN & ";UID=" & strUser &
";PWD=" & strPassword & ";DATABASE=" & strDatabase
    glbmyCon.Open
    myblConnect = True

    objCom.ActiveConnection = glbmyCon

Else
    Set glbmyConLocal = New ADODB.Connection
    glbmyConLocal.ConnectionString = "DSN=" & strDSN & ";UID=" &
strUser & ";PWD=" & strPassword & ";DATABASE=" & strDatabase
    glbmyConLocal.Open
    myblConnect = True
    objCom.ActiveConnection = glbmyConLocal
End If
'Élèöèàèèçàöèÿ êîäèðîâîê
objCom.CommandText = "SET NAMES 'cp1251'"
objCom.Execute
objCom.CommandText = "set character set cp1251"
objCom.Execute
objCom.CommandText = "set character_set_client='cp1251'"
objCom.Execute
objCom.CommandText = "set character_set_results='cp1251'"
objCom.Execute
objCom.CommandText = "set collation_connection='cp1251_general_ci'"
objCom.Execute

Exit Sub
ErrorHandler:
Select Case Err.number
    Case -2147467259
        MsgBox Err.description

        Exit Sub
    Case Else
        MsgBox "Err " & Err & "."
        Exit Sub
End Select
End Sub

```

## Доступ до Access

```
Public Sub ConnectAccess(strpth As String, objCon As ADODB.Connection)
```

```
    Dim strcon As String
```

```
    Dim k As Integer
```

```
    'Set objCon = New ADODB.Connection
```

```
    strcon = "Provider=Microsoft.Jet.OLEDB.4.0;"
```

```
    strcon = strcon & "Data Source=" & strpth & ";"
```

```
    strcon = strcon & "Jet OLEDB:Engine Type=5;"
```

```
    strcon = strcon & "Persist Security Info=False"
```

```
    On Error GoTo ErrorHandler
```

```
    objCon.Open strcon
```

```
Exit Sub
```

```
ErrorHandler:      '
```

```
    Select Case Err.number
```

```
        Case -2147467259
```

```
            MsgBox "Помилка"
```

```
            Exit Sub
```

```
        Case Else
```

```
            MsgBox "Помилка: " & Err.description
```

```
            Resume Next
```

```
    End Select
```

```
Exit Sub
```

```
End Sub
```

## З'єднання з MySQL

```
Private Sub cmdNewConnect_Click()
```

```
Me.MousePointer = vbHourglass
```

```
If glbCon.State > 1 Then glbmyCon.Close
```

```
cmdConnect.Caption = ">>--/--<<"
```

```
Call ConnectMySQL(blConnect, txtDSN, txtUser, txtPasswordMy2, txtDatabase,  
"remote")
```

```
If blConnect = False Then
```

```
    MsgBox "Помилка"
```

```
    cmdConnect.Caption = ">>--/--<<"
```

```
Else
```

```
    MsgBox "Помилка"
```

```
    cmdConnect.Caption = ">>----<<"
```

```

cmdReplStart.Enabled = True
If blCheck = False Then
    If FormIsOpen("frmAccount") Then
        Rdispatcher.frmAccount.SetFocus
    Else
        Rdispatcher.frmAccount.Show
    End If
' frmAccount.Show
End If

```

```

End If
Me.MousePointer = vbDefault
End Sub

```

### **Формування SQLзапита.**

```

Public Sub UpdateFieldOnTable(objCon As ADODB.Connection, strTable As String, _
strFieldUpdate As String, strFieldUpdateValue As String, _
strFieldWHERE As String, strFieldWHEREValue As String, blLog As Boolean)
Dim strQuery As String

```

```

Set objCom = CreateObject("ADODB.Command")
If strFieldUpdateValue Like "*" & Chr(39) & "*" Then
    strFieldUpdateValue = replChr(strFieldUpdateValue, "'", "''")
End If

```

```

strQuery = "Update `" & strTable & "` SET " & strFieldUpdate & " = " & _
Chr(39) & strFieldUpdateValue & Chr(39) & _
" WHERE " & strFieldWHERE & " = " & strFieldWHEREValue & "'"

```

```

objCom.CommandType = adCmdText
objCom.ActiveConnection = objCon
objCom.CommandText = strQuery
On Error GoTo ErrorHandler
objCom.Execute
If blLog Then
    Call AddLogRepl(glbmyConLocal, strQuery)
End If

```

Exit Sub

ErrorHandler:

```

Select Case Err.number
Case 3464, -2147217913
    strQuery = "Update `" & strTable & "` SET " & strFieldUpdate & " = " & _
Chr(39) & strFieldUpdateValue & Chr(39) & _

```

```

        " WHERE " & strFieldWHERE & " = " & Val(strFieldWHEREValue)
objCom.CommandText = strQuery
objCom.Execute
If blLog Or FormIsOpen("frmEditTable") Then
    Call AddLogRepl(glbmyConLocal, strQuery)
End If

```

```

        Exit Sub
    Case Else
        MsgBox "Err " & Err & "."
    Exit Sub
End Select

```

End Sub

### **Визначення класу.**

Private Sub Class\_Initialize()

```

Set myComm = New MSCommLib.MSComm
'Set myComm = CreateObject("MSCommLib.MSComm")

```

```

myComm.Settings = "115200,E,8,1"
'myComm.Settings = "2400,N,8,1"
SaveSetting App.Title, "Vend_Status", "Exe_Status", exe_status_wait
SaveSetting App.Title, "Vend_Status", "Vend_Action", 0

```

```

myComm.RThreshold = 1
myComm.SThreshold = 1

```

```

mvarBVL_ACCEPTED = mvar_def_BVL_ACCEPTED
mvarBVL_STATUS = mvar_def_BVL_STATUS
mvarBVL_SETUP = mvar_def_BVL_SETUP
mvarBVL_STACKER = mvar_def_BVL_STACKER
mvarVMC_COMM = mvar_def_VMC_COMM
mvarVMC_STATUS_ANSWER = mvar_def_VMC_STATUS_ANSWER
mvarCHANGER_STATUS = mvar_def_CHANGER_STATUS
mvarCHANGER_TUBE_STATUS = mvar_def_CHANGER_TUBE_STATUS
mvarCHANGER_IDENTIFICATION =
mvar_def_CHANGER_IDENTIFICATION
mvarBVL_IDENTIFICATION = mvar_def_BVL_IDENTIFICATION
mvarCHANGER_SETUP = mvar_def_CHANGER_SETUP
mvarCHANGER_FULL_STATUS = mvar_def_CHANGER_FULL_STATUS
mvarCHANGER_SCALLING = mvar_def_CHANGER_SCALLING
mvarCHANGER_DECIMAL = mvar_def_CHANGER_DECIMAL

```

```

mvarBVL_CREDIT = mvar_def_BVL_CREDIT
mvarCHANGER_CREDIT = mvar_def_CHANGER_CREDIT
mvarBVL_SCALLING = mvar_def_BVL_SCALLING
mvarBVL_DECIMAL = mvar_def_BVL_DECIMAL
mvarBVL_STACKER_CAPACITY = mvar_def_BVL_STACKER_CAPACITY
mvarMASTER_SCALLING = mvar_def_MASTER_SCALLING
mvarMASTER_DECIMAL = mvar_def_MASTER_DECIMAL
mvarVMC_QUERY_DISPENSE = mvar_def_VMC_QUERY_DISPENSE
mvarMASTER_CREDIT = mvar_def_MASTER_CREDIT
mvarCHANGER_COIN_DEPOSITED =
mvar_def_CHANGER_COIN_DEPOSITED
mvarBVL_BILL_VALUES = mvar_def_BVL_BILL_VALUES
mvarCHANGER_COIN_VALUES = mvar_def_CHANGER_COIN_VALUES

```

*End Sub*

Обробка компорту.

*Private Sub myComm\_OnComm()*

```

Dim EVMsg$
Dim ERMsg$
Dim Buffer As Variant
Dim btTemp As Byte
Dim InBufferCountCurrent As Long
Dim strTemp As String
Dim strTmp As Variant
Dim strEXPANSION() As String
Dim iStatus As Integer

```

*Dim i, j As Double*

*On Error GoTo ErrorHandler*

*Select Case myComm.CommEvent*

*' Event messages.*

*Case comEvReceive                   ', comRxParity*

*SaveSetting App.Title, "C-MDB", "Vend\_Timeout", 0 '*

*InBufferCountCurrent = myComm.InBufferCount*

*If InBufferCountCurrent > 0 Then*

*Buffer = myComm.Input*

*strTemp = Buffer*

*'       Debug.Print strTemp*

*'       myComm.Output = Chr(&H2)*

```

'      myComm.Output = Chr(&H3)
'      myComm.Output = Chr(&H6)
'      myComm.Output = Chr(&H0)
'      myComm.Output = Chr(&HC2)
'      myComm.Output = Chr(&H82)
'      Exit Sub

btTemp = Val(Asc(Buffer) And &HF0)

If Hex(btTemp) = 30 Then
    Debug.Print "Éîàîäà= &H" & Hex(Asc(Buffer))

    Select Case Asc(Buffer)

        Case VMC_STATUS '
            SaveSetting App.Title, "C-MDB", "BVL_POLL_Status", 0
            SaveSetting App.Title, "Output", "last_out", Chr(&H0)
            Debug.Print "."
            Call EVENT_MDB_TEST("TEST IS OK")
            myComm.Output = Chr(VMC_STATUS_ANSWER) '&H0
            If Request_CHECK_CHANGER Then
                strTemp = mvarCHANGER_SETUP
                Call EVENT_CHANGER_SETUP(strTemp)
            End If
            Request_CHECK_CHANGER = False

        Case vmc_credit '
            'Debug.Print "VMC_QUERY_DISPENSE = " &
VMC_QUERY_DISPENSE
            vmc_credit_reply = VMC_QUERY_DISPENSE

            If vmc_credit_reply <> 254 Then
                Debug.Print Hex(vmc_credit_reply)
            End If

            myComm.Output = Chr(vmc_credit_reply) '
            Debug.Print "Îòâò àà Credit = " & Hex(vmc_credit_reply)

        Case vmc_vend
            Debug.Print "Vend, Êóîþðà " & Hex(Val(iStatus And &HF)) & "
äîëæíà áúòü ó÷òâíà"
            SaveSetting App.Title, "C-MDB", "Vend_Action", 1
            SaveSetting App.Title, "C-MDB", "Vend_Credit", 0
            SaveSetting App.Title, "Output", "last_out", Chr(&H0)
            myComm.Output = Chr(&H0)

```

```

Call EVENT_BVL_STACKED("OK")

Case vmc_audit
  Debug.Print "ÉÛàíää AUDIT"
  SEND_ADDRESS_COUNT = 0
  myComm.Output = Chr(&H5) '
  mymass = Split(VMC_DISPENSE, ",")

Case vmc_send_address
  Debug.Print "ÉÛàíää SEND_ADDRESS"
  myComm.Output = Chr(SEND_ADDRESS_COUNT)

Case vmc_send_data
  Debug.Print "ÉÛàíää SEND_DATA"
  myComm.Output = Chr(mymass(SEND_ADDRESS_COUNT))
  SEND_ADDRESS_COUNT = SEND_ADDRESS_COUNT + 1

Case vmc_identify
  Debug.Print Hex(Asc(Buffer))

Case vmc_accept_data
  Debug.Print "-----"
  word_receive = ""
  SaveSetting App.Title, "C-MDB", "Exe_Status",
exe_status_continue_receive
  SaveSetting App.Title, "Output", "last_out", Chr(&H0)
  myComm.Output = Chr(&H0) 'vmc_status_nothing

Case vmc_nak '
  Debug.Print "-- "
  strTemp = GetSetting(App.Title, "Output", "last_out", Chr(&H0))
  myComm.Output = strTemp

Case vmc_sync
  Debug.Print "-- --"
  Debug.Print word_receive
  strTemp = ""
  strTmp = ""
  For i = 1 To Len(word_receive)
    Debug.Print Hex(Asc(Mid(word_receive, i, 1)) And &HF) & " "
    Select Case i - (i \ 2) * 2
      Case 1
        btTemp = Val(Asc(Mid(word_receive, i, 1)) And &HF)
      Case 0
    End Select
  Next i

```

```

        btTemp = btTemp + 16 * Val(Asc(Mid(word_receive, i, 1)) And
&HF)
    If i < Len(word_receive) Then
        strTemp = strTemp & btTemp & ","
    Else
        strTemp = strTemp & btTemp
    End If
End Select
Next i
'btTemp = Val(strTemp)
mymass = Split(strTemp, ",")
btTemp = mymass(0)
strTemp = ""
For i = 1 To UBound(mymass)
    If i = UBound(mymass) Then
        strTemp = strTemp & mymass(i)
    Else
        strTemp = strTemp & mymass(i) & ","
    End If
Next i
'
' strTmp = Right(strTemp, Len(strTemp) - 3)
' If Left(strTmp, 1) = "," Then
'     strTmp = Right(strTemp, Len(strTemp) - 4)
' End If
Select Case btTemp

    Case p_BVL_SETUP     '= &H31
        SaveSetting App.Title, "C-MDB", "BVL_SETUP", strTemp
        Debug.Print "Êüàíàà MDB p_BVL_SETUP &H" & Hex(btTemp)
& " " & vbCrLf & "Íòââò" & vbCrLf & strTemp
        mvarBVL_SETUP = strTemp
        If Len(strTemp) = 0 Then
            myComm.Output = Chr(&H0)
            Call EVENT_BVL_SETUP(strTemp)
            Exit Sub
        End If
        mymass = Split(strTemp, ",")
        strTmp = ""
        For i = 11 To UBound(mymass)
            If i = UBound(mymass) Then
                strTmp = strTmp & mymass(i)
            Else
                strTmp = strTmp & mymass(i) & ","
            End If
        Next i

```



```

    Bill_Values = Split(strTmp, ",")
    mvarBVL_SCALLING = mymass(4)
    mvarBVL_DECIMAL = mymass(5)
    mvarBVL_STACKER_CAPACITY = mymass(6) * 256 +
mymass(7)

    mvarBVL_BILL_VALUES = strTmp
    Call EVENT_BVL_SETUP(strTmp)
    Case p_C_MDB_FLASH_REPORT '&HE1
        SaveSetting App.Title, "C-MDB",
"p_C_MDB_FLASH_REPORT", strTemp
        Debug.Print "ÉÛàíàà p_C_MDB_FLASH_REPORT &H" &
Hex(btTemp) & " " & vbCrLf & "Íòââò" & vbCrLf & strTemp
        '
        '
        mvarC_MDB_FLASH_REPORT = strTemp
        mymass = Split(strTemp, ",")
        Debug.Print "Àäðãñ FLASH 80H ... " & strTemp

    Case p_BVL_POLL ' = &H33
        SaveSetting App.Title, "C-MDB", "BVL_POLL_Status", 1
        SaveSetting App.Title, "C-MDB", "BVL_POLL", strTemp
        Debug.Print "ÉÛàíàà MDB p_BVL_POLL &H" & Hex(btTemp)
& " " & vbCrLf & "Íòââò" & vbCrLf & strTemp
        iStatus = Val(strTemp)
        mvarBVL_STATUS = iStatus
        If iStatus > 127 Then
            Select Case Val(iStatus And &HF0)
                Case &H80
                    Debug.Print " MDB p_BVL_POLL, " & Hex(Val(iStatus And
&HF)) & "--"

                    SaveSetting App.Title, "C-MDB", "BVL_CREDIT", strTemp '
                    i = Val(strTemp And &HF)
                    mvarBVL_ACCEPTED = Hex(Val(iStatus And &HF))
                    mvarBVL_CREDIT = Val(Bill_Values(i)) * BVL_SCALLING
                    Call EVENT_BVL_STACKED(strTemp)

                Case &H90
                    Debug.Print "ÉÛàíàà MDB p_BVL_POLL, " &
Hex(Val(iStatus And &HF)) & " -- Escrow"
                    'SaveSetting App.Title, "C-MDB", "BVL_CREDIT", strTemp '
                    'i = Val(strTemp And &HF)
                    'mvarBVL_ACCEPTED = Hex(Val(iStatus And &HF))
                    'mvarMASTER_CREDIT = Val(Bill_Values(i))
                    mvarBVL_ACCEPTED = Hex(Val(iStatus And &HF))
                    Call EVENT_BVL_ESCROW(Val(vmc_credit_reply))

                Case &HA0

```

```

        Debug.Print "ÊÛàíää MDB p_BVL_POLL, " &
Hex(Val(iStatus And &HF)) & "--"
        Call EVENT_BVL_RETURNED(Val(vmc_credit_reply))
        mvarBVL_ACCEPTED = iStatus

    Case &HB0
        Debug.Print " MDB"

    Case &HC0
        Debug.Print "ÊÛàíää MDB p_BVL_POLL, " &
Hex(Val(iStatus And &HF)) & " DISABLED BVL REJECTED"
        mvarBVL_ACCEPTED = iStatus

    End Select
Else
    Debug.Print "ÊÛàíää MDB p_BVL_POLL, STATUS " &
Hex(Val(strTemp))
    mvarBVL_STATUS = iStatus
    Call EVENT_BVL_FAIL(strTemp)
End If

Case p_BVL_ESCROW '= &H35
    SaveSetting App.Title, "C-MDB", "BVL_ESCROW", strTemp
    Debug.Print " MDB p_BVL_ESCROW &H" & Hex(btTemp) & "
" & vbCrLf & "--" & vbCrLf & strTemp

Case p_BVL_STACKER '= &H36
    SaveSetting App.Title, "C-MDB", "BVL_STACKER", strTemp
    Debug.Print "ÊÛàíää MDB p_BVL_STACKER &H" &
Hex(btTemp) & " " & vbCrLf & "--" & vbCrLf & strTemp
    mvarBVL_STACKER = strTemp
    'If Val(strTemp) > 127 Then
        Call EVENT_BVL_FULL(strTemp)
    'End If

Case p_BVL_EXPANSION_COMMAND
    mymass = Split(strTemp, ",")
    '
    btTemp = mymass(0)
    strTemp = ""
    For i = 1 To UBound(mymass)
        strTemp = strTemp & Chr(mymass(i))
    Next i
    SaveSetting App.Title, "C-MDB",
"BVL_EXPANSION_COMMAND", strTemp

```

```

        Debug.Print "MDB p_BVL_EXPANSION_COMMAND
&H" & Hex(btTemp) & " " & vbCrLf & "Iòââò" & vbCrLf & strTemp
        mvarBVL_IDENTIFICATION = strTemp
        Call EVENT_BVL_SETUP(strTemp)
'-----
    Case p_CHANGER_SETUP    '= &H9
        SaveSetting App.Title, "C-MDB", "CHANGER_SETUP", strTemp
        Debug.Print "MDB p_CHANGER_SETUP &H" & Hex(btTemp)
& " " & vbCrLf & "--" & vbCrLf & strTemp
        mvarCHANGER_SETUP = strTemp
        If Len(strTemp) = 0 Then
            myComm.Output = Chr(&H0)
            Call EVENT_CHANGER_SETUP(strTemp)
            Exit Sub
        End If
        mymass = Split(strTemp, ",")
        strTmp = ""
        For i = 7 To UBound(mymass)
            If i = UBound(mymass) Then
                strTmp = strTmp & mymass(i)
            Else
                strTmp = strTmp & mymass(i) & ","
            End If
        Next i
        Coin_Values = Split(strTmp, ",")
        mvarCHANGER_SCALLING = mymass(3)
        mvarCHANGER_DECIMAL = mymass(4)
        mvarCHANGER_COIN_VALUES = strTmp
        Call EVENT_CHANGER_SETUP(strTemp)

    Case p_CHANGER_POLL    '= &HB
        SaveSetting App.Title, "C-MDB", "CHANGER_POLL", strTemp
        Debug.Print "MDB p_CHANGER_POLL &H" & Hex(btTemp)
& " " & vbCrLf & "--" & vbCrLf & strTemp
        iStatus = Val(strTemp)
        If iStatus = 0 Then
            mvarCHANGER_STATUS = strTemp
            Call EVENT_CHANGER_fail(strTemp)
        ElseIf iStatus > 127 Then
            mvarCHANGER_STATUS = strTemp
            Call EVENT_CHANGER_fail(strTemp)
        ElseIf (iStatus < 128) And (iStatus > 63) Then
            mvarCHANGER_STATUS = 0
            Debug.Print "-- " & Val(iStatus And &HF) & " ";
            Select Case Val(iStatus And &H30)

```

```

        Case &H0
            Debug.Print " ---"
            i = Val(iStatus And &HF)
            mvarCHANGER_COIN_DEPOSITED = i
            mvarCHANGER_CREDIT = Coin_Values(i) *
CHANGER_SCALLING
            Call EVENT_CHANGER_ACCEPTED(strTemp)

        Case &H10
            Debug.Print " ---"
            i = Val(iStatus And &HF)
            mvarCHANGER_COIN_DEPOSITED = i
            mvarCHANGER_CREDIT = Coin_Values(i) *
CHANGER_SCALLING
            Call EVENT_CHANGER_ACCEPTED(strTemp)
        Case &H30
            Debug.Print " ---"
        End Select
    '
    Debug.Print " MDB p_CHANGER_POLL, -- " &
(Val(strTmp))
    'Call EVENT_CHANGER_fail(strTemp)
Else
    mvarCHANGER_STATUS = strTemp
    Call EVENT_CHANGER_fail(strTemp)
End If

Case p_CHANGER_TUBE_STATUS '= &HA
    SaveSetting App.Title, "C-MDB", "CHANGER_TUBE_STATUS",
strTemp
    'Debug.Print " MDB p_CHANGER_TUBE_STATUS &H" &
Hex(btTemp) & " " & vbCrLf & "--" & vbCrLf & strTemp
    mymass = Split(strTemp, ",")
    mvarCHANGER_TUBE_STATUS = mymass(2) & "," &
mymass(3) & "," & mymass(4) & "," & mymass(5) & "," & mymass(6)
    mvarCHANGER_FULL_STATUS = mymass(0) & "," &
mymass(1)

    If mymass(1) > 0 Then
        Call EVENT_CHANGER_tube_full(strTemp)
    End If

Case p_changer_coin_VALUES '= &HC

Case p_changer_dispense '= &HD

Case p_CHANGER_EXPANSION_COMMAND

```

```

        SaveSetting App.Title, "C-MDB",
"CHANGER_EXPANSION_COMMAND", strTemp
        btTemp = mymass(0)
        'strTemp = mymass(1)
        Select Case Val(mymass(1))
        Case 0
            mymass = Split(strTemp, ",")
            strTemp = ""
            For i = 1 To UBound(mymass)
                strTemp = strTemp & Chr(mymass(i))
            Next i
            '
            mvarchanger_optional_features = mymass(30) & "," &
mymass(31) & "," & mymass(32) & "," & mymass(33)
            mvarCHANGER_IDENTIFICATION = strTemp
            Call EVENT_CHANGER_SETUP(strTemp)

            Debug.Print " MDB p_CHANGER_EXPANSION_COMMAND
&H" & Hex(btTemp) & Hex(mymass(1)) & " " & btTemp & vbCrLf & "--" &
vbCrLf & strTemp

        Case 1

        Case 5

        End Select

        Case p_VENDING_CREDIT_DISPLAY '= &HE0
            SaveSetting App.Title, "C-MDB", "VENDING_CREDIT",
strTemp
            Debug.Print "ÊÛàïäà p_VENDING_CREDIT_DISPLAY &H" &
Hex(btTemp) & " Çià÷âièà " & strTemp
            mymass = Split(strTemp, ",")
            mvarMASTER_SCALLING = mymass(2)
            mvarMASTER_DECIMAL = Val(mymass(3) And &HF0) \ 16
            mvarMASTER_CREDIT = (mymass(0) + mymass(1) * 256) *
mvarMASTER_SCALLING

            Call EVENT_MASTER_CREDIT(strTemp)
        Case Else
            Debug.Print " MDB" & Hex(btTemp) & "H"
        End Select
        myComm.Output = Chr(&H0)

        Case Else
            Debug.Print " VMC" & Hex(Asc(Buffer)) '

```

```
SaveSetting App.Title, "Output", "last_out", Chr(&H0)
myComm.Output = Chr(&H0)
End Select
```

```
ElseIf Hex(btTemp) = 20 Then
Debug.Print "Äàííûâ = &H" & Hex(Asc(Buffer))
word_receive = word_receive & Buffer
SaveSetting App.Title, "Output", "last_out", Chr(&H0)
myComm.Output = Chr(&H0)
Exit Sub
End If
```

```
End If
```

```
Case comEvSend
```

```
If Asc(GetSetting(App.Title, "Output", "last_out", "p")) = "0" Then
Debug.Print " &H0"
```

```
Else
```

```
Debug.Print "-- " & GetSetting(App.Title, "Output", "last_out", "p")
End If
```

```
Case comEvCTS
```

```
EVMsg$ = "Change in CTS Detected"
```

```
Case comEvDSR
```

```
EVMsg$ = "Change in DSR Detected"
```

```
Case comEvCD
```

```
EVMsg$ = "Change in CD Detected"
```

```
Case comEvRing
```

```
EVMsg$ = "The Phone is Ringing"
```

```
Case comEvEOF
```

```
EVMsg$ = "End of File Detected"
```

```
' Error messages.
```

```
Case comBreak
```

```
ERMsg$ = "Break Received"
```

```
Case comCDTO
```

```
ERMsg$ = "Carrier Detect Timeout"
```

```
Case comCTSTO
```

```
ERMsg$ = "CTS Timeout"
```

*Case comDCB*  
*ERMsg\$ = "Error retrieving DCB"*

*Case comDSRTO*  
*ERMsg\$ = "DSR Timeout"*

*Case comFrame*  
*ERMsg\$ = "Framing Error"*

*Case comOverrun*  
*ERMsg\$ = "Overrun Error"*

*Case comRxOver*  
*ERMsg\$ = "Receive Buffer Overflow"*

*Case comRxParity*  
*ERMsg\$ = "Parity Error"*

*Case comTxFull*  
*ERMsg\$ = "Transmit Buffer Full"*

*Case Else*  
*ERMsg\$ = "Unknown error or event"*

*End Select*

*If Len(EVMsg\$) Then*

*'Debug.Print EVMsg\$*

*ElseIf Len(ERMsg\$) Then*

*Buffer = myComm.Input*

*Debug.Print "-- " & ERMsg\$ & ", -- "*

*' SaveSetting App.Title, "Output", "last\_out", Chr(&HFF)*

*myComm.Output = Chr(&HFF) 'PNAK*

*End If*

*EVMsg\$ = ""*

*Exit Sub*

*ErrorHandler:*

*mvarCHANGER\_CREDIT = 0*

*mvarBVL\_CREDIT = 0*

*mvarMASTER\_CREDIT = 0*

*Call EVENT\_CHANGER\_ACCEPTED(strTemp)*

*End Sub*