

**PRAVION**

**PM100 シリーズ  
アプリケーション開発  
スタートアップガイド  
(Microsoft Visual Studio 編)**

2012 年 03 月 27 日

**浜松東亜電機株式会社**

## 目次

1 . 開発環境の構築.....	3
1 - 1 . SDK のダウンロード .....	3
1 - 2 . SDK のインストール .....	3
2 . アプリケーション開発手法 (ライブラリ未使用) .....	4
2 - 1 . アプリケーションの作成 .....	4
2 - 1 - 1 . プロジェクトの種類選択について.....	4
2 - 1 - 2 . アプリケーションのコーディング.....	5
2 - 2 . デバッグ.....	6
2 - 2 - 1 . PC と PM100 との接続 .....	6
2 - 2 - 2 . デバッグの開始.....	7
3 . 開発したアプリケーションの PM100 へのコピー .....	8
3 - 1 . Microsoft Visual Studio デバッグを使用する方法.....	8
3 - 2 . Microsoft ActiveSync を使用する方 法 .....	8
3 - 3 . microSD カードを使用する方 法 .....	9
4 . アプリケーション開発手法 (ライブラリ使用の場合) .....	10
4 - 1 . ライブラリの種類とダウンロード方法.....	10
4 - 2 . SDK の使用方法 (BCD.net.dll) .....	10
4 - 2 - 1 . BCD.net.dll の、参照設定 .....	10
4 - 2 - 2 . アプリケーションのコーディング.....	10
4 - 3 . SDK の使用方法 (UnitApi および BCDCore.dll) .....	12
4 - 3 - 1 . 参照設定.....	12
4 - 3 - 2 . アプリケーションのコーディング.....	12
5 . インストーラの作成方法.....	15
5 - 1 . プロジェクトの追加.....	15
5 - 2 . ファイルの追加.....	16
5 - 3 . PM100 への配置 .....	18

当スタートアップガイドでは、Microsoft Visual Studio 操作方法および言語仕様についての説明は省略させていただいておりますので、ご了承ください。

Microsoft Visual Studio は、2005 Professional Edition または 2005 Standard Edition、2008 Professional Edition をご用意ください。

2005/2008 Express Edition、2008 Standard Edition、2010 全 Edition では、モバイル (WindowsCE) アプリケーション開発はサポートされていません。

また、Microsoft .Net Compact Framework 2.0 SP1 以降をご使用ください。

## 1 . 開発環境の構築

### 1 - 1 . SDK のダウンロード

PRAVION サイトより、PM100 簡易開発キットをダウンロードしてください。

<http://www.pravion.com/pda/download/index.html>

### 1 - 2 . SDK のインストール

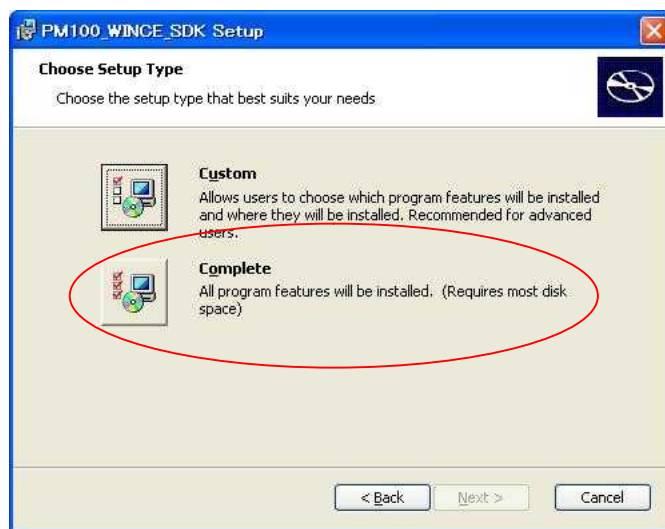
ダウンロードしたインストーラ ( PM100\_WINCE\_SDK.msi ) を実行してください。



事前に、Microsoft Visual Studio をインストールしておいてください

インストーラのメッセージに従って、インストールを実施してください。

なお、Setup Type の選択では、Complete を選択してください。



## 2 . アプリケーション開発手法 (ライブラリ未使用)

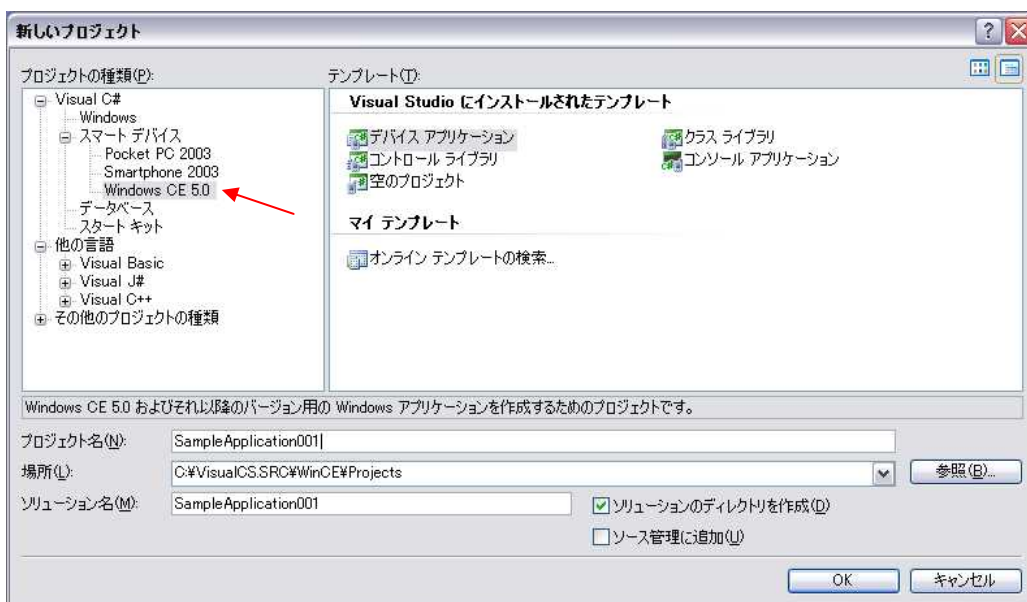
PC 上にてアプリケーションを開発し、PDA 上にて動作させるまでの手順を記載します。

### 2 - 1 . アプリケーションの作成

Microsoft Visual Studio にて、アプリケーションを開発いたします。  
ここでは、VisualStudio2005 の VB.NET にて開発する例を記載いたします。  
サンプルとして、PDA 画面上に「Hello World」を表示させ、ボタンをクリックすると「PM100」に表示を変更するアプリケーションを作成します。

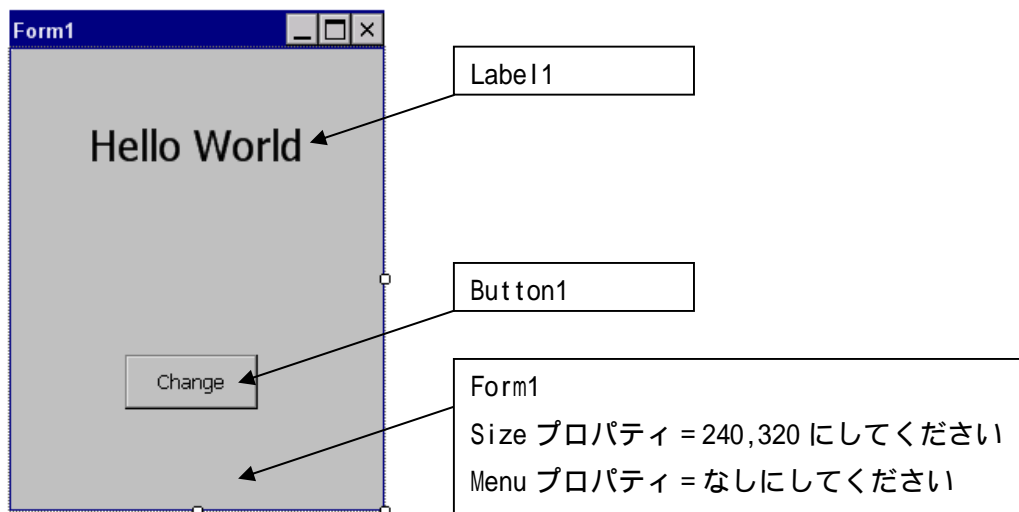
#### 2 - 1 - 1 . プロジェクトの種類選択について

「新しいプロジェクト」にて、プロジェクトの種類を選択する際は「Windows CE 5.0」を選択してください。



## 2 - 1 - 2 . アプリケーションのコーディング

ここでは、下記の様なアプリケーションを作成しました。



```

? Button1をクリックしたら、Label1の表示を変更する
? Button1がクリックされた時の処理
Public Class Form1
? Label1の表示を変更
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Label1.Text = "PM100"
    End Sub
End Class

```

## 2 - 2 . デバッグ

ここでは、PM100 を PC と接続し、デバッグを実施する方法について記載いたします。

### 2 - 2 - 1 . PC と PM100 との接続

まず、PC と PM100 を接続してください。

デバッグを実施する際は、USB を使用して接続します。



クレードル(100-HB)

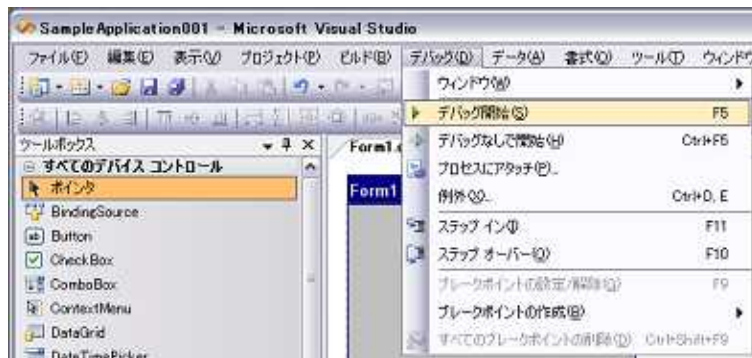
オプションの専用 SYNC ケーブル(100-USB)  
または、オプションのクレードル(100-HB) を  
使用して、接続してください。

PM100 タスクバーに下記アイコンが表示されれば、正常に接続されています。

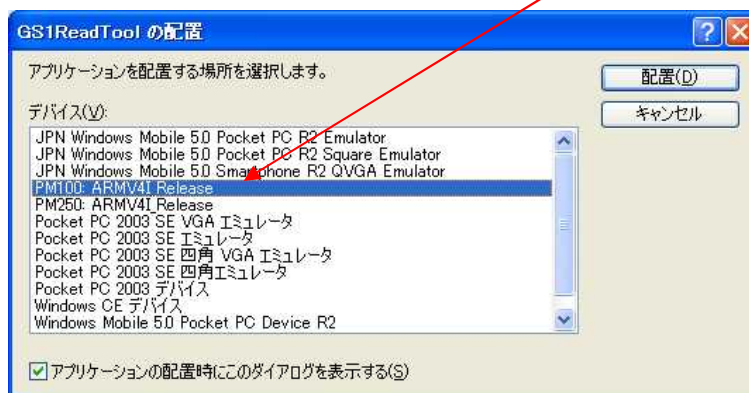


## 2 - 2 - 2 . デバッグの開始

Microsoft Visual Studio のメニューから「デバッグ開始」を選択してください。



アプリケーションを配置する場所には、PM100 を選択してください。



しばらくお待ちいただくと、PM100 画面上に、先ほど作成したアプリケーションが起動しています。



以上で、PM100 上でのデバッグを実施する事が出来ます。

デバッグ中にブレークポイントの指定が正常動作しない場合は、Microsoft .Net Compact Frame Work2.0 SP1 より古いバージョンを使用している可能性が有ります。

### 3 . 開発したアプリケーションの PM100 へのコピー

開発したアプリケーションを PM100 にコピーする方法には、下記の手法が有ります。

#### 3 - 1 . Microsoft Visual Studio デバッグを使用する方法

前項「2 - 2 - 2 . デバッグの開始」にて「アプリケーションの配置」が実施されます。

これによって、PM100 の ProgramFiles フォルダ内に、アプリケーションフォルダおよび実行ファイルが配置されます。

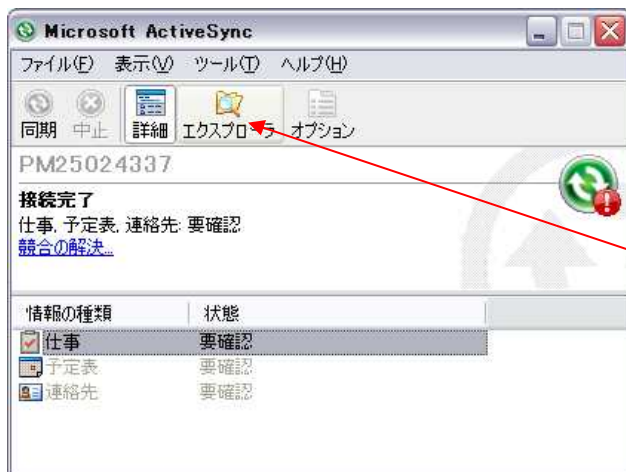
#### 3 - 2 . Microsoft ActiveSync または Microsoft Windows Mobile デバイスセンターを使用する方法

Microsoft Active Sync ( Windows XP ) または Microsoft Windows Mobile Device Center ( Windows Vista / 7 ) を使用する事により、開発したアプリケーションおよび各種ファイルを PM100 へコピーする事が出来ます。



PC と PM100 間は、USB にて接続してください。  
Bluetooth による接続には対応しておりません。

ActiveSync を使用する場合 ( Windows XP )



ActiveSync 画面上にて「エクスプローラ」を選択し、直接ファイルコピー操作を実施してください。

ActiveSync のインストール方法・操作方法については、割愛します。



Microsoft Windows Mobile デバイスセンターを使用する場合  
(Windows Vista / 7)



Microsoft Windows Mobile デバイスセンター画面上にて「ファイル管理」の「デバイスのコンテンツの参照」を選択すると、接続している PM250 のエクスプローラ画面に切り替わります。直接ファイルコピー操作を実施してください。

### 3 - 3 . microSD カードを使用する方法

PM100 には microSD カードスロットを搭載しています。こちらを使用し、ファイルをコピーする事が可能です。microSD カードスロットは、バッテリーの裏にあります。



## 4 . アプリケーション開発手法 (ライブラリ使用の場合)

ライブラリを利用したアプリケーションの開発手順を記載します。

### 4 - 1 . ライブラリの種類とダウンロード方法

ライブラリには「基本ライブラリ」「スキャナ開発用ライブラリ」の2種類があります。

PRAVION ウェブサイトより、ダウンロードしてください。

<http://www.pravion.com/pda/download/index.html>

### 4 - 2 . SDK の使用方法 (BCD.net.dll)

BCD.net.dll は.NET ユーザが容易に SDK を利用できるものです。

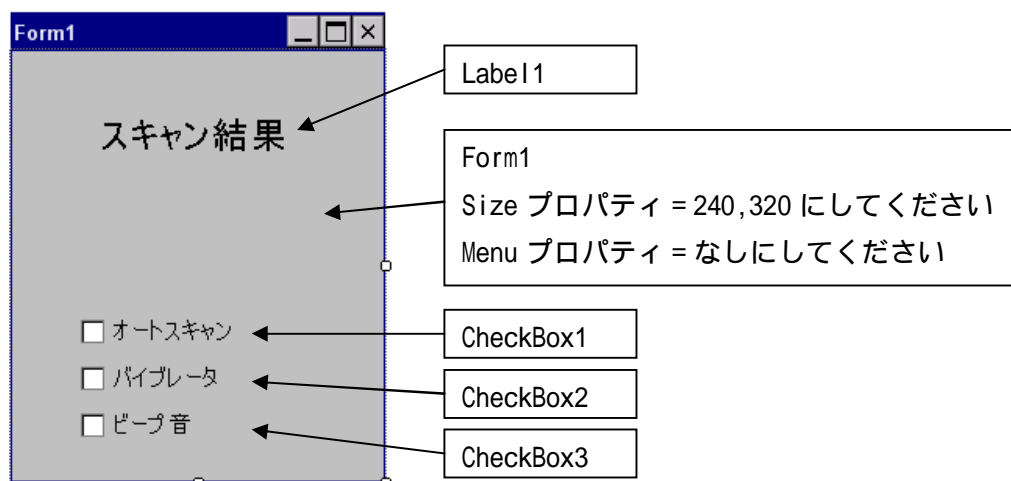
#### 4 - 2 - 1 . BCD.net.dll の、参照設定

Microsoft Visual Studio のメニューから「プロジェクトのプロパティ」を開いて参照に BCD.net を追加します。



#### 4 - 2 - 2 . アプリケーションのコーディング

BCD.net.dll を使ってスキャナを利用する簡単なアプリケーションを作成します



```

Imports BCD.net
Public Class Form1
    BCD.net.dll の DBScanner宣言
    Private WithEvents BScanner As DBScanner
    ' Form1のLoad
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Load
        ' BScanner作成
        BScanner = New DBScanner
        ' スキャナー追加に初期化
        BScanner.Enable(True)
        ' オートスキャン未使用に初期化
        BScanner.EnableAutoScan(False)
        CheckBox1.Checked = False
        ' ビープ音有効に初期化
        BScanner.EnableBeep(True)
        CheckBox2.Checked = True
        ' バイブレータ有効に初期化
        BScanner.EnableVibrator(True)
        CheckBox3.Checked = True
    End Sub
    ' Form1のClose
    Private Sub Form1_Closed(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Closed
        ' BScanner解放
        BScanner = Nothing
    End Sub
    ' DBScannerのOnScanned (スキャン完了)
    Private Sub BScanner_OnScanned(ByVal data As String, ByVal nLen As UInteger, ByVal barType As Integer, ByVal err As Long) Handles BScanner.OnScanned
        ' スキャン結果判定
        If err = 0 Then
            ' err = 0ならスキャン成功
            Label1.Text = data
        Else
            ' その他はスキャン失敗
            Label1.Text = "スキャン失敗"
        End If
    End Sub
    ' オートスキャン チェックボックス変化
    Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged
        BScanner.EnableAutoScan(CheckBox1.Checked)
    End Sub
    ' バイブレータ チェックボックス変化
    Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox2.CheckedChanged
        BScanner.EnableVibrator(CheckBox2.Checked)
    End Sub
    ' ビープ音 チェックボックス変化
    Private Sub CheckBox3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox3.CheckedChanged
        BScanner.EnableBeep(CheckBox3.Checked)
    End Sub
End Class

```

アプリケーションを実行し、SCAN ボタンでバーコードをスキャンするとスキャン結果に表示されます。



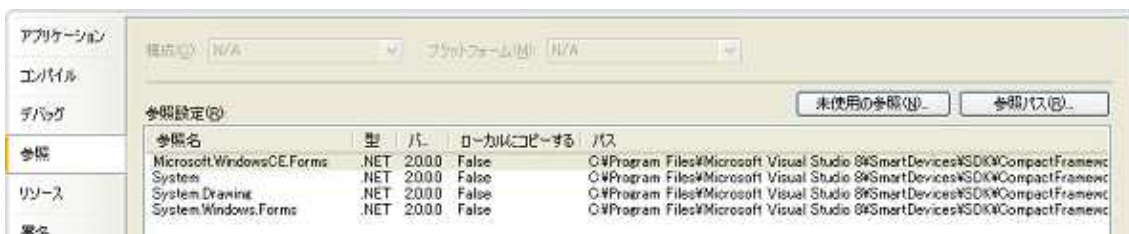
### 4 - 3 . SDK の使用方法 (UnitApi および BCDCore.dll)

BCD.net.dll は全ての関数には対応していません。

ここでは BCD.net.dll を使用せずに UnitApi.dll および BCDCore.dll を直接利用するアプリケーションを作成します。

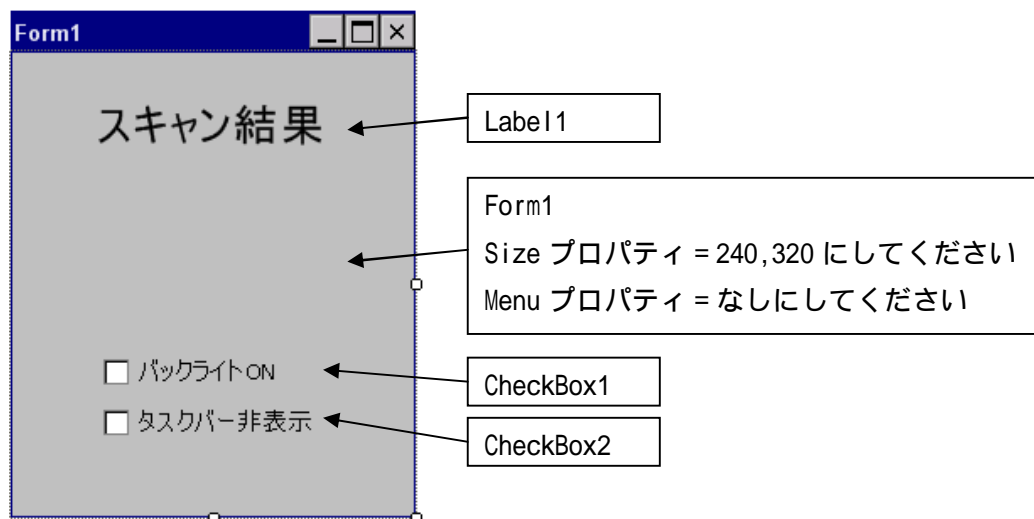
#### 4 - 3 - 1 . 参照設定

Microsoft.WindowsCE.Forms を利用するため、参照設定に追加します。



#### 4 - 3 - 2 . アプリケーションのコーディング

UnitApi.dll を使ってバックライトの ON/OFF、タスクバーの表示/非表示を切り替えます。また、BCDCore.dll を直接使用してスキャナを利用します。



```

' BCD.netを使わないサンプル
' =====
Public Class Form1
    ' EVENT定義
    Public WithEvents Scanner As New ScanEvent

    ' DLL関数定義
    ' BCDCore.dll
    Public Declare Function BcdOpen Lib "BCDCore" () As IntPtr
    Public Declare Sub BcdClose Lib "BCDCore" ()
    Public Declare Sub BcdIsEnabled Lib "BCDCore" (ByVal bEnable As Boolean)
    Public Declare Sub BcdSetResultType Lib "BCDCore" (ByVal dwType As Integer)
    Public Declare Sub BcdSetHWND Lib "BCDCore" (ByVal hWnd As IntPtr)
    Public Declare Function BcdGetString Lib "BCDCore" () As IntPtr
    Public Declare Function BcdGetLength Lib "BCDCore" () As Integer
    ' UnitApi.dll
    Public Declare Sub BacklightTurnOn Lib "UnitApi" (ByVal bOn As Boolean)
    Public Declare Sub HideTaskBar Lib "UnitApi" (ByVal bHide As Boolean)

    ' WindowMessage受信用クラス
    Private HandleWindow As New cHandleWindow

    ' =====
    ' FormのLoadの処理
    ' =====
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Load
        ' スキャナー初期化
        BcdOpen()
        BcdIsEnabled(True)
        BcdSetResultType(0)
        BcdSetHWND(HandleWindow.Handle)

        ' バックライトONで初期化
        BacklightTurnOn(True)
        CheckBox1.Checked = True
        ' タスクバー表示で初期化
        HideTaskBar(False)
        CheckBox2.Checked = False
    End Sub

    ' =====
    ' FormのCloseの処理
    ' =====
    Private Sub Form1_Closed(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Closed
        BcdClose()
    End Sub

    ' =====
    ' Scann完了イベント
    ' =====
    Private Sub OnScan(ByVal sender As Object, ByVal e As ScannedParam) Handles Scanner.Scanned
        If e.errCode = 0 Then
            ' スキャン文字列アドレス取得
            Dim pRet As IntPtr = BcdGetString()
            ' スキャン文字列
            Dim iLen As Integer = BcdGetLength()
            Dim scanBuff(iLen) As Char
            Dim sScanned As String = ""
            Dim i As Integer
            ' バッファへコピー
            System.Runtime.InteropServices.Marshal.Copy(pRet, scanBuff, 0, iLen)
            For i = 0 To iLen - 1
                sScanned += scanBuff(i)
            Next i
            Label1.Text = sScanned
        Else
            Label1.Text = "スキャン失敗"
        End If
    End Sub
End Class

```

```

'.....
' バックライト変更
'.....
Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged
    'バックライトを変更
    Backlight.TurnOn(CheckBox1.Checked)
End Sub
'.....
' タスクバー非表示変更
'.....
Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox2.CheckedChanged
    'タスクバー非表示を変更
    HideTaskBar(CheckBox2.Checked)
End Sub
End Class
'.....
' スキャン結果引継ぎクラス
'.....
Public Class ScannedParam
    Inherits EventArgs
    Public errCode As Integer
    Public Sub New(ByVal param As Integer)
        errCode = param
    End Sub
End Class
'.....
' スキャンイベント定義クラス
'.....
Public Class ScanEvent
    Public Event Scanned(ByVal sender As Object, ByVal e As ScannedParam)
    ' Scan用イベント送信要求
    Public Sub ScannedRequest(ByVal errCode As IntPtr)
        ' Scan完了イベント送信
        Dim args As New ScannedParam(errCode.ToInt32())
        RaiseEvent Scanned(Me, args)
    End Sub
End Class
'.....
' WindowMessage 変換クラス
'.....
Friend Class cHandleWindow
    Inherits Microsoft.WindowsCE.Forms.MessageWindow
    ' WindowMessage定義
    Public Const WM_USER As Integer = &H400
    Public Const WM_SCANNED As Integer = WM_USER + &H7777
    ' WndProcをOverrideしてWM_SCANNED受信
    Protected Overrides Sub WndProc(ByRef m As Microsoft.WindowsCE.Forms.Message)
        If m.Msg = WM_SCANNED Then
            ' Scan完了イベント送信要求
            Form1.Scanner.ScannedRequest(m.LParam)
        End If
        MyBase.WndProc(m)
    End Sub
End Class
'.....

```

SCAN ボタンにより 4 - 2 と同様にバーコードのスクランが行えます。  
 またタスクバー非表示をチェックするとタスクバーが隠れます。

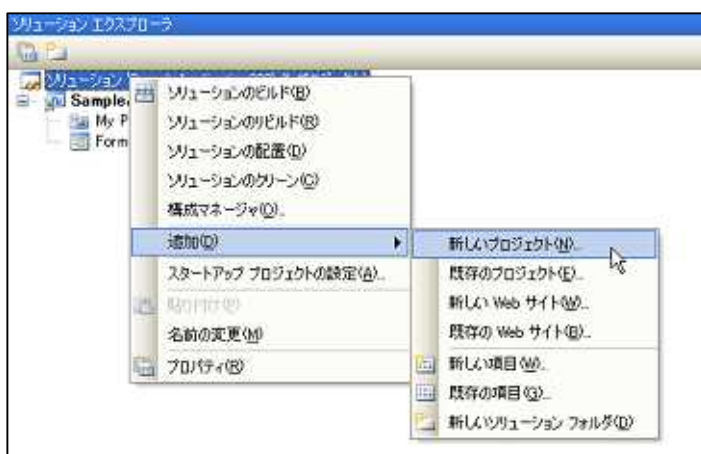


## 5 . インストーラの作成方法

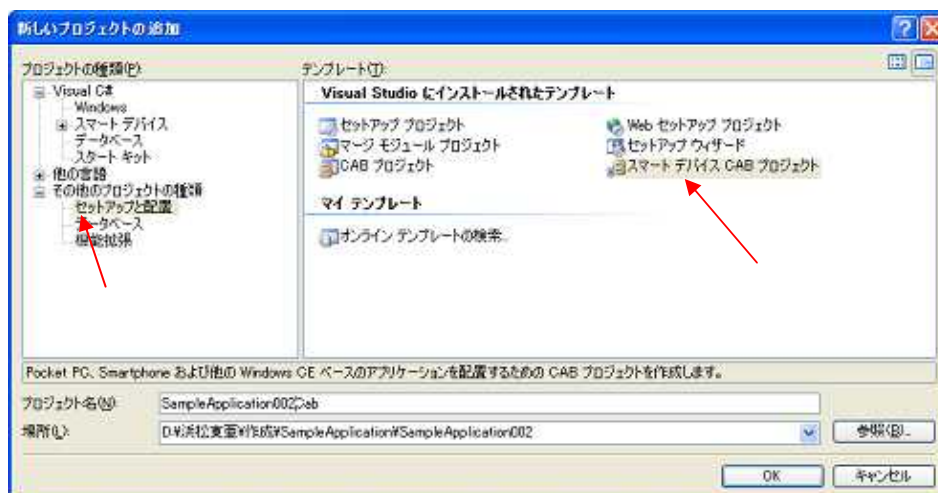
作成した EXE の他に DLL や INI ファイル、データファイルなどを一括して PM100 へ設定するには CAB ファイルを作成します。

### 5 - 1 . プロジェクトの追加

ソリューションにプロジェクトを追加します。

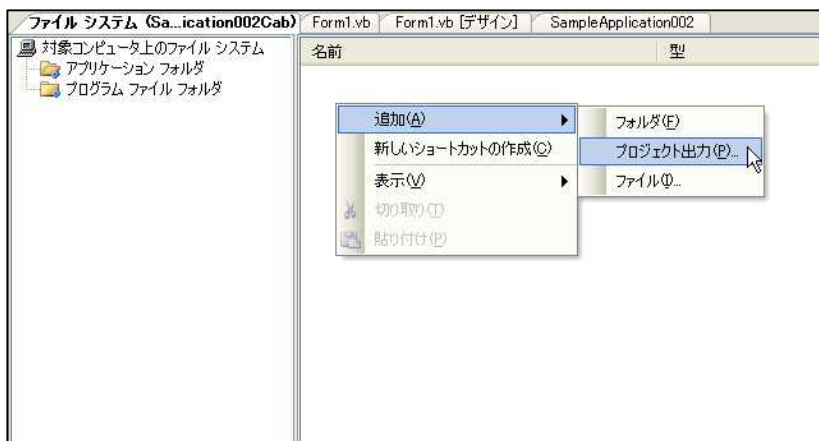


プロジェクトの種類にて「セットアップと配置」を選択し、テンプレートは「スマートデバイス CAB プロジェクト」を選択します。

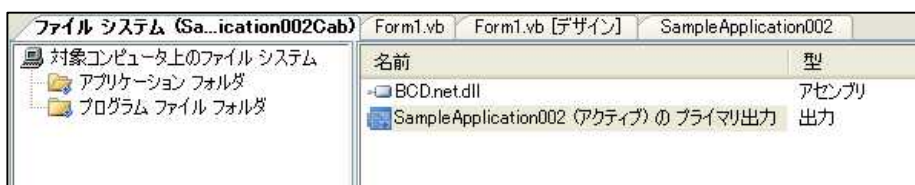
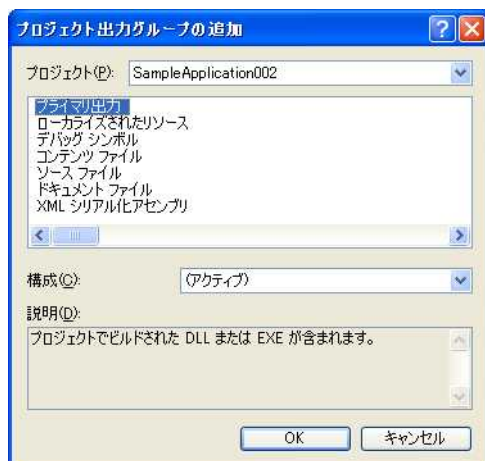


## 5 - 2 . ファイルの追加

「ファイルシステム」を表示し「アプリケーションフォルダ」を選択します。  
名前欄で右クリック 「追加」 「プロジェクト出力」を選択します。



プライマリ出力を選択するとプロジェクトの EXE と利用している DLL が自動的に設定されます。

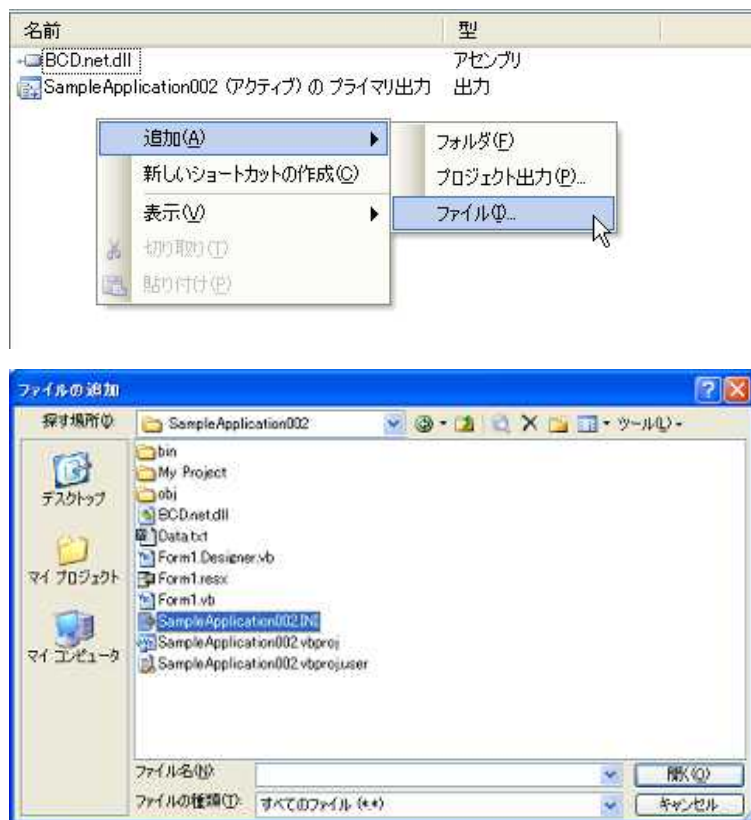


「アプリケーションフォルダ」を指定した場合、PM250 の ProgramFiles 配下にプロジェクト名のフォルダが作成され、そこに EXE と DLL が配置されます。

「プログラムファイルフォルダ」を指定した場合、PM250 の ProgramFiles 直下に配置されます。



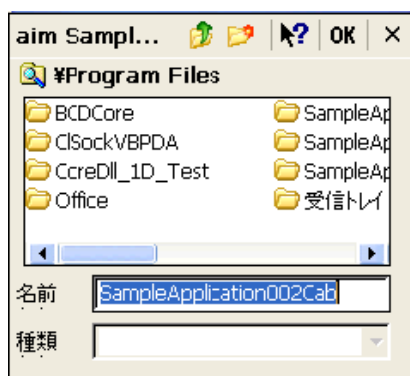
INI ファイルや DATA ファイルなどを指定する場合は、右クリック「追加」「ファイル」でファイルの追加ダイアログを表示して、追加したいファイルを指定します。



必要なファイルを全て設定したら、プロジェクトをビルドし CAB ファイルを作成します。

### 5 - 3 . PM100 への配置

3章で示した手法で CAB ファイルを PM100 の任意のフォルダへ配置します。  
CAB ファイルをダブルクリックすると下の画面が表示されます。



OK を押すと名前欄のディレクトリが作成され、そこに指定したプログラム等  
が配置されます。