

84
400
21

LEBANESE AMERICAN UNIVERSITY

RP
23
C.1

EYE2SEE
A MOBILE SURVEILLANCE SYSTEM FOR
POCKET PC

SAMI M. SOUKI

NOT TO
CIRCULATE


AUGUST 2004

eye2see
A mobile surveillance system for Pocket PC

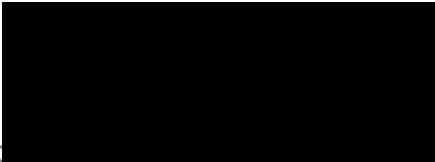
Sami Souki

Project

Submitted in fulfillment of the requirements for the degree of
Master of Science in Computer Science
at the Lebanese American University
Beirut, Lebanon
August 2004



Dr. Ramzi Haraty (Advisor)
Associate Professor of Computer Science
Lebanese American University



Dr. Abdul Nasser Kassar
Associate Professor
Beirut Arab University

ACKNOWLEDGMENT

I would like to seize the opportunity to express my sincere gratitude and appreciation to Dr. Ramzi Haraty, my advisor, who gave me the chance and lead me through accomplishing this project. Many thanks to the Lebanese American University who helped me by providing me with all the needed resources.

Thank you...

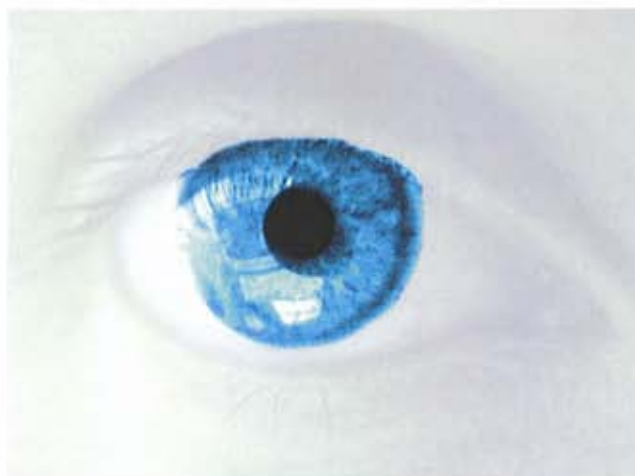
Lebanese American University

Version

1

MOBILE TECHNOLOGY

eye2see



Be aware... Someone out there might be watching you...

MS Project Report

MOBILE TECHNOLOGY

eye2see Project Report

© Sami Souki
Lebanese American University
Phone 961-3-924159 • Email: ssouki@lynx.net.lb

Table of Contents

CHAPTER 1

Introduction	2
Proposed Idea	4
Scope	4
Report Organization	5

CHAPTER 2

Surveillance Systems	6
Resources	7

CHAPTER 3

First Approach	8
COM Object	8
Some Problems	9
Second Approach	11
Main Algorithms	11

CHAPTER 4

Sample Run	12
Source Code	12
Setup	13

CHAPTER 5

Conclusion	14
Future Work	14
References	14

Introduction

As technology progresses and evolves, the trend is targeting mobile environment and clone all current aspects of technology into mobile platforms. The need to control and access information from anywhere and through wireless fashion is rising. Eye2see is an excellent example on utilizing mobile technology.

Mobile technology is the most important and impressive offspring of technology. Below is a list of some new aspects of mobile technology:

- **Bluetooth**

Bluetooth is a standard developed by a group of electronics manufacturers that allows any sort of electronic equipment -- from computers and cell phones to keyboards and headphones -- to make its own connections, without wires, cables or any direct action from a user. Bluetooth is intended to be a standard that works at two levels:

- It provides agreement at the physical level -- Bluetooth is a radio-frequency standard.
- It also provides agreement at the next level up, where products have to agree on when bits are sent, how many will be sent at a time and how the parties in a conversation can be sure that the message received is the same as the message sent.



- LANs and WANs

LAN (Local Area Network) is a network that links computers, printers and other devices located in an office, a building or even a campus. A WAN (Wide Area Network) is a system that extends for greater distances and is used to connect LANs together.

- Smart phones



A Smart Phone is a mobile phone that can send and receive voice, data and video. This phone generation has various features like connecting to the Internet for web access and receiving emails. The Smart Phone consolidates the typical mobile phone with the Personal Digital Assistant (PDA) and digital camera, and thus creating a powerful communicating Mobile Intelligent Device (MID).

- PDA

Personal Digital Assistant, or PDA, a remarkable, tiny, fully functional computer that you can hold in one hand. A PDA can do more than hold address books and calendar events. PDAs are the one of the fastest selling consumer devices in history. These devices receive information, process information, and communicate with other computers and PDAs.



One of the most important things involved with successful mobile work and telecommunicating is good communication. There are a wide variety of tools and technology available today to keep us connected no matter where we are. The good news is that these tools and services seem to keep getting less expensive every day. With the technology available today, the issue of mobile computing being difficult to accomplish is a thing of the past.

Proposed Idea

After examining all the information mentioned before, one could foresee that a project tackling mobile technology is a precious and interesting task to do. The initial proposed idea was to develop an application for a smart phone running the Symbian operating system. An application could be a system utility, game or a graphical one. After some consultations, a more interesting proposal has emerged. The issue of controlling a remote object using a PDA was more appealing and genuine. The object to be controlled is a cam and not a simple web cam. It should have a tilt, pan and zoom features. To summarize in few words, the application to be developed is a mobile surveillance system that allows a mobile user to access a cam installed at a specific target place and to manipulate its movement and monitor.

The Scope

The scope of the project and the features that are included in the application are listed below:

- Capability to retrieve frames from the cam at an acceptable time delay.
- Capability to control the tilt / pan / zoom features of the cam (4 basic directions, home repositioning, 4x zoom).
- Capturing frames at will and store the captured images into the PDA memory.
- Ability to browse captured images.
- Schedule future tasks to run invisibly in the background and save image frames into storage card memory.

Report Organization

Chapter 1

This chapter presents an introduction about mobile technology and describes the main idea of the project stating its scope.

Chapter 2

This chapter describes surveillance systems and states the resources used to implement this project.

Chapter 3

This chapter describes the actual implementation work done and summarizes the algorithms used.

Chapter 4

A sample run and the actual source code are included in this chapter along with setup tips.

Chapter 5

This chapter includes the conclusion and suggestions for future work.

Surveillance Systems

Surveillance systems are widely spread applications one can find a large variety of similar systems. Some online websites offer such services but the most famous ones are the ones that are packaged with specific cameras. Once the cam is installed and the specified application allows the user to access the cam from any PC connected to the internet.

What is unique in this project is that it offers a solution that was never provided before at least in normal surveillance systems. As mentioned before, surveillance systems are meant to be used with PCs or notebooks but not with handhelds or PDAs.

This proposed application will be a breakthrough in the surveillance systems world. Monitoring our home while we are driving, watching a special place to assure its security from a mobile location are examples of the offered services. Figure 1 illustrates this claim by showing an example of a similar application that is going to be launched on July 1st, 2004.

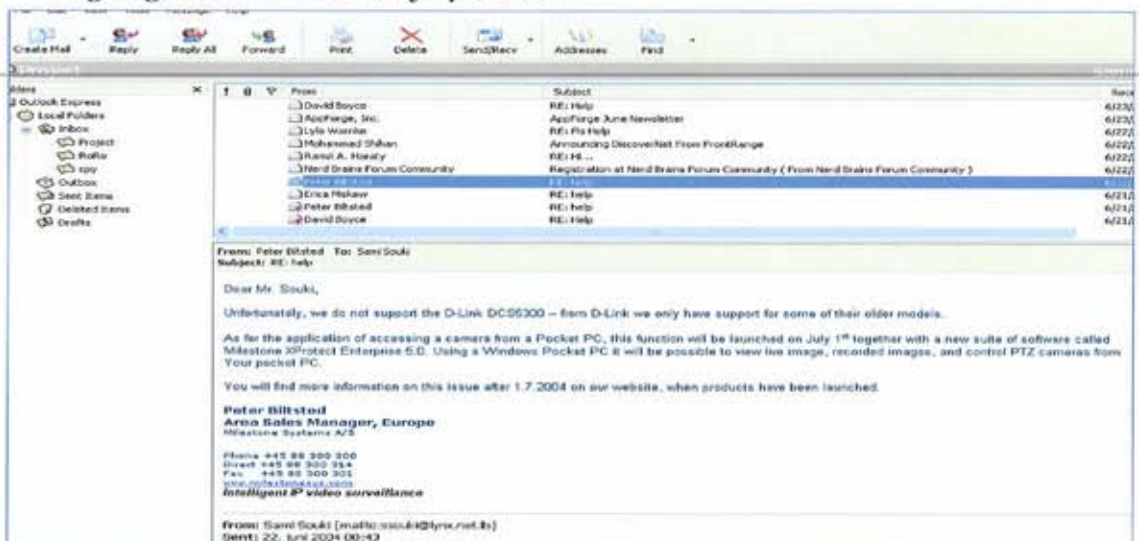


Figure 1 – Milestone Systems application

Figure 2 shows that the manufacturer has no plans for similar applications.

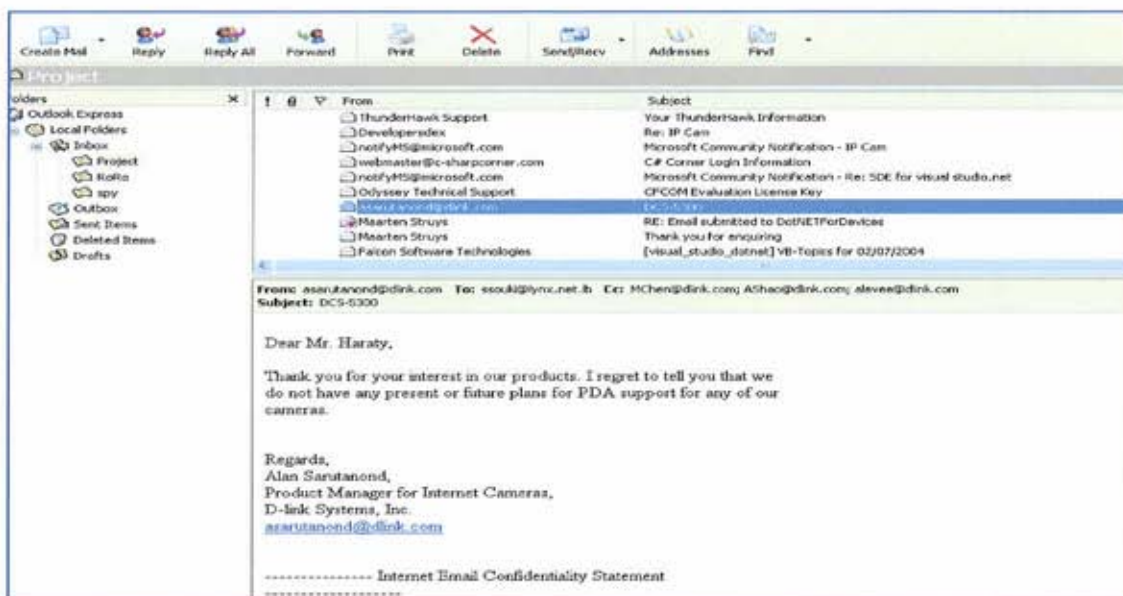


Figure 2 – D-Link Support

Resources

1. DCS 5300

The cam that will be used is a DCS 5300 manufactured by D-Link.

The D-Link *SecuriCam* **DCS-5300** Internet Camera is a full featured surveillance system that connects to an Ethernet, Fast Ethernet or broadband Internet connection to provide remote high-quality video and audio. The **DCS-5300** is the latest product added to the D-Link internet camera line. The camera features a motorized pan and tilt function found on more expensive cameras. This function allows the viewing area of the camera to extend 270 degrees side-to-side and 90 degrees up and down.



2. I-mate PPC

I-Mate pocket PC, the I-Mate PPC is a fully integrated voice enabled GSM/GPRS portable personal computing device that runs the phone edition of windows Mobile 2003 and features an SDIO slot to be used for memory cards.



3. Visual studio .NET 2003 Architect Edition

First Approach

The DCS 5300 cam is a stand alone camera that has its own operating system and ROM built in. It has an Ethernet interface, once plugged into a network it configures itself as an independent component that can be accessed through a web interface. In other words, the cam can be accessed through its IP address which will invoke an active-X control built in the cam memory (ROM). The only piece of code provided by the camera documentation is the following:

```
document.write("<OBJECT ID=\"VaCtrl\" WIDTH=362 HEIGHT=306");
document.write(" CLASSID=CLSID:A93B47FD-9BF6-4DA8-97FC-9270B9D64A6C");
document.write("
CODEBASE=\"http://xxx.xxx.xxx.xxx:5004/plugin/h263ctrl.cab#version=1,7,0,5\">");
document.write("<PARAM NAME=\"Url\" VALUE=\"http://xxx.xxx.xxx.xxx:5004/cgi-
bin/video.vam\">");
document.write("<PARAM NAME=\"VSize\" VALUE=\"SIF\">");
document.write("<PARAM NAME=\"RemoteIP\" VALUE=\"xxx.xxx.xxx.xxx:5004\">");
document.write("<PARAM NAME=\"RemotePort\" VALUE=\"5001\">");
document.write("</OBJECT>");
```

This characteristic implies that the cam acts as a COM (component object module) object. Therefore, the first approach to tackle this application was to use the cam as a COM object.

COM Object

Components developed using Microsoft's COM provide a way by which two objects in different object spaces or networks, could talk together by calling each other's methods. This excellent technology forces the operating system to see applications as objects.

COM forces the OS to act as a central registry for objects. The OS takes the responsibility of creating objects when they are required, deleting them when they are not, and handling communications between them, be it in the same or different processes or machines. One major advantage of this mechanism is versioning. If the COM object ever changes to a new version, the applications that use that object need not be recompiled.

The wonderful thing about COM components is that they are never linked to any application. The only thing that an application may know about a COM object is what functions it may or may not support. In fact, the object model is so flexible that applications can query the COM object at run-time as to what functionality it provides.

Some Problems

Now looking at the features provided by the pocket PC, one discovers that Javascript and active-X are not supported neither by the operating system nor by the internet browser installed.

As for the DCS 5300 cam support, nothing was available to aid in solving this problem. A Software Development Kit (SDK) was an essential requirement which was not provided by the manufacturer as shown below in figure 3.

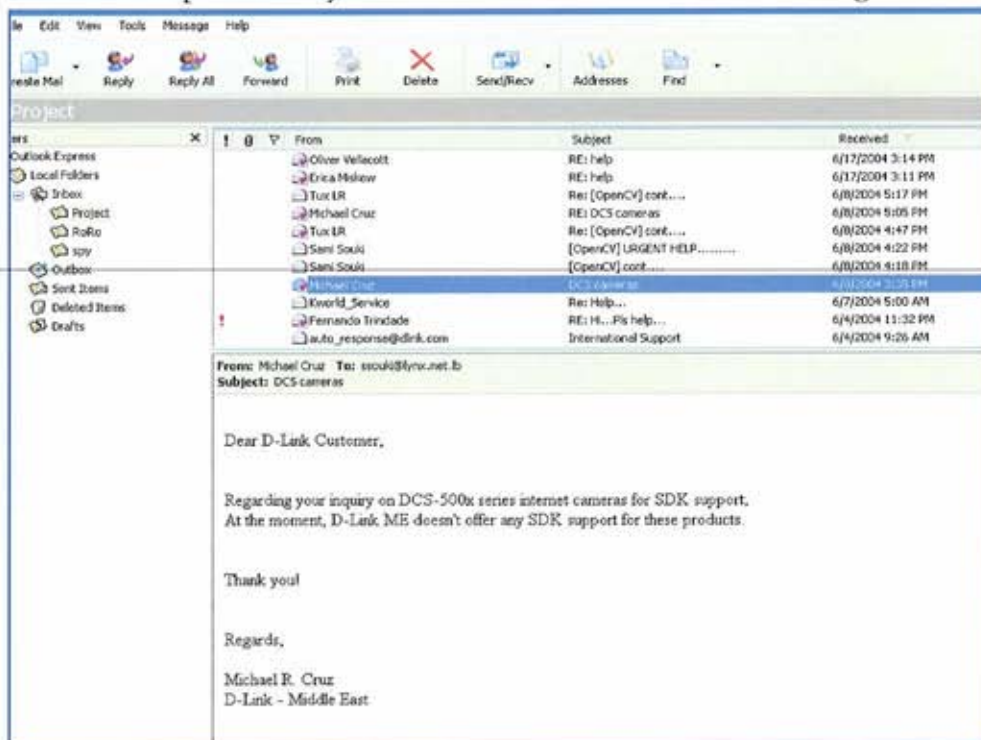


Figure 3 – D-Link SDK for DCS-5300 cam

Second Approach

A second approach was considered to solve the problems mentioned above. In this approach, it was decided to use the Visual Studio .NET 2003 (Architect Edition) to develop an application that is capable to establish a connection between the cam and the Pocket PC.

Visual Studio .NET 2003 includes integrated support for the Microsoft .NET Compact Framework. Using the Microsoft Windows Forms designer, Microsoft Visual Basic and Microsoft Visual C# developers, one can easily build, debug, and deploy powerful applications for the Pocket PC, Pocket PC Phone Edition, and other smart devices powered by the .NET Compact Framework. Integrated emulation enables developers to program and debug applications.

The actual and complete source code of the eye2see application is included in Chapter 4.

Main Algorithms

The features previously listed in the scope section are described below:

- *Capability to retrieve frames from the cam at an acceptable time delay.*
The main idea was to establish a tunnel (connection) between the eye2see program and the cam by addressing its IP address and then using timers to regulate the frame rate. A **webrequest** was used and the data retrieved from the cam was encapsulated into a stream variable. This variable (stream) was later on converted into a bitmap image to be displayed in the main form.
- *Capability to control the tilt / pan / zoom features of the cam (4 basic directions, home repositioning, 4x zoom).*
The same approach as in the first feature was used. A **webrequest** was adopted to send a request to the cam. Five parameters were used to summarize the Five directions listed. The zoom feature was programmed and implemented on one side mainly on the Pocket PC. In other words, the zoom action was simulated by manipulating the image frame retrieved from the cam.

- *Capturing frames at will and store the captured images into the PDA memory.*

This procedure was developed using Visual C# .NET. Converting a bitmap variable into a bitmap image file to reside on storage memory was not a straight forward task. The C# module, the variable bitmap had to be read bit by bit and stored in a two dimensional matrix and then be written to a file which constitutes the bitmap image.

- *Ability to browse captured images.*

A form was created to enable the user to browse the captured frames. This procedure scans the target directory and loads the names of the captured images into a combo list and displays the chosen picture upon selection.

- *Schedule future tasks to run invisibly in the background and save image frames into storage card memory.*

The scheduling issue was somehow a challenging task because the schedule task must reside in memory (in the background) until it is triggered when the target time is met. Therefore, this module was developed separately. It is installed automatically with the main package and is called from within the eye2see main program but it remains running in an invisible mode until it finishes the capture. The captured frames will be stored on the storage card if found if not they will be stored in a specific folder.

Sample Run

Source Code

Setup



```

Imports OpenNETCF.Diagnostics
Imports System.IO
Imports System.Net

Public Class Form1
    Inherits System.Windows.Forms.Form
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        MyBase.Dispose(disposing)
    End Sub

    'NOTE: The following procedure is required by the Windows Form
Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents Timer1 As System.Windows.Forms.Timer
    Private Sub InitializeComponent()
        Dim resources As System.Resources.ResourceManager = New
System.Resources.ResourceManager(GetType(Form1))
        Me.PictureBox1 = New System.Windows.Forms.PictureBox
        Me.Timer1 = New System.Windows.Forms.Timer
        '
        'PictureBox1
        '
        Me.PictureBox1.Image =
CType(resources.GetObject("PictureBox1.Image"), System.Drawing.Image)
        Me.PictureBox1.Size = New System.Drawing.Size(240, 320)
        Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        '
        'Timer1
        '
        Me.Timer1.Interval = 1
        Me.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.Fixed3D
        '
        'Form1
        '
        Me.Controls.Add(Me.PictureBox1)
        Me.MaximizeBox = False
        Me.MinimizeBox = False
        Me.Text = "eye2see"
        Me.WindowState = System.Windows.Forms.FormWindowState.Maximized

    End Sub

```

```
#End Region
    Public Shared delay As Integer
    Public Shared picname As String
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim p As New Process

        If Not File.Exists("\program files\schedule\schedule.exe") Then
            p.StartInfo.FileName = "wceload.exe"
            p.StartInfo.Arguments = """\program
files\eye2see\schedule_PPC.ARM.CAB""
            p.Start()
            p.Dispose()
        End If

        Timer1.Interval = 3000
        Timer1.Enabled = True
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        Dim f2 As New form2
        Timer1.Enabled = False
        delay = 100
        picname = "captured"

        f2.Show()
    End Sub
End Class
```



```
Imports System
Imports System.IO
Imports System.Net
```

```
Public Module module1
```

```
    Public counter As Integer
    Public v As Integer
    Public ip As String
```

```
    Public Sub capturetofile(ByVal p As Image)
        Dim path As String
        path = "\my documents\eye2see\" + Trim(Form1.picname) +
Trim(Str(counter)) + ".jpg"
        While File.Exists(path)
            counter = counter + 1
            path = "\my documents\eye2see\" + Trim(Form1.picname) +
Trim(Str(counter)) + ".jpg"
        End While
        SaveThisImage.BitmapSaver.SaveBitmapToFile(p, path)
        counter = counter + 1
    End Sub
```

```
    Public Sub getframe(ByRef p As Bitmap)
```

```
        Dim request As WebRequest = WebRequest.Create("http://" + ip +
"/cgi-bin/video.jpg")
```

```
        Dim stream As Stream =
request.GetResponse().GetResponseStream()
        p = New Bitmap(stream)
        stream.Close()
```

```
    End Sub
```

```
    Public Sub movecamup()
```

```
        Dim request As WebRequest = WebRequest.Create("http://" + ip +
"/cgi-bin/camctrl.cgi?move=up")
```

```
        Dim stream As Stream =
request.GetResponse().GetResponseStream()
        stream.Close()
```

```
    End Sub
```

```
    Public Sub movecamdown()
```

```
        Dim request As WebRequest = WebRequest.Create("http://" + ip +
"/cgi-bin/camctrl.cgi?move=down")
```

```
        Dim stream As Stream =
request.GetResponse().GetResponseStream()
        stream.Close()
```

```
    End Sub
```

```
    Public Sub movecamleft()
```

```
        Dim request As WebRequest = WebRequest.Create("http://" + ip +
"/cgi-bin/camctrl.cgi?move=left")
```

```

        Dim stream As Stream =
request.GetResponse().GetResponseStream()
        stream.Close()

    End Sub
    Public Sub movecamright()

        Dim request As WebRequest = WebRequest.Create("http://" + ip +
"/cgi-bin/camctrl.cgi?move=right")

        Dim stream As Stream =
request.GetResponse().GetResponseStream()
        stream.Close()

    End Sub
    Public Sub movecamhome()

        Dim request As WebRequest = WebRequest.Create("http://" + ip +
"/cgi-bin/camctrl.cgi?move=home")

        Dim stream As Stream =
request.GetResponse().GetResponseStream()
        stream.Close()

    End Sub
End Module
Public Class form2
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        MyBase.Dispose(disposing)
    End Sub

    'NOTE: The following procedure is required by the Windows Form
Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents Timer2 As System.Windows.Forms.Timer
    Friend WithEvents home As System.Windows.Forms.PictureBox
    Friend WithEvents up As System.Windows.Forms.PictureBox
    Friend WithEvents move As System.Windows.Forms.Label
    Friend WithEvents leftt As System.Windows.Forms.PictureBox
    Friend WithEvents rightt As System.Windows.Forms.PictureBox
    Friend WithEvents down As System.Windows.Forms.PictureBox

```

```

Friend WithEvents PictureBox2 As System.Windows.Forms.PictureBox
Friend WithEvents PictureBox3 As System.Windows.Forms.PictureBox
Friend WithEvents PictureBox4 As System.Windows.Forms.PictureBox
Friend WithEvents PictureBox5 As System.Windows.Forms.PictureBox
Friend WithEvents TrackBar1 As System.Windows.Forms.TrackBar
Friend WithEvents Panell1 As System.Windows.Forms.Panel
Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
<System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
    Dim resources As System.Resources.ResourceManager = New
System.Resources.ResourceManager(GetType(form2))
    Me.Timer2 = New System.Windows.Forms.Timer
    Me.leftt = New System.Windows.Forms.PictureBox
    Me.home = New System.Windows.Forms.PictureBox
    Me.rightt = New System.Windows.Forms.PictureBox
    Me.up = New System.Windows.Forms.PictureBox
    Me.down = New System.Windows.Forms.PictureBox
    Me.move = New System.Windows.Forms.Label
    Me.PictureBox2 = New System.Windows.Forms.PictureBox
    Me.PictureBox3 = New System.Windows.Forms.PictureBox
    Me.PictureBox4 = New System.Windows.Forms.PictureBox
    Me.PictureBox5 = New System.Windows.Forms.PictureBox
    Me.TrackBar1 = New System.Windows.Forms.TrackBar
    Me.Panell1 = New System.Windows.Forms.Panel
    Me.PictureBox1 = New System.Windows.Forms.PictureBox
    '
    'Timer2
    '
    '
    'leftt
    '
    Me.leftt.Image = CType(resources.GetObject("leftt.Image"),
System.Drawing.Image)
    Me.leftt.Location = New System.Drawing.Point(80, 240)
    Me.leftt.Size = New System.Drawing.Size(24, 32)
    Me.leftt.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'home
    '
    Me.home.Image = CType(resources.GetObject("home.Image"),
System.Drawing.Image)
    Me.home.Location = New System.Drawing.Point(104, 240)
    Me.home.Size = New System.Drawing.Size(32, 32)
    Me.home.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'rightt
    '
    Me.rightt.Image = CType(resources.GetObject("rightt.Image"),
System.Drawing.Image)
    Me.rightt.Location = New System.Drawing.Point(136, 240)
    Me.rightt.Size = New System.Drawing.Size(24, 32)
    Me.rightt.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'up
    '
    Me.up.Image = CType(resources.GetObject("up.Image"),
System.Drawing.Image)

```



```

        Me.up.Location = New System.Drawing.Point(104, 216)
        Me.up.Size = New System.Drawing.Size(32, 24)
        Me.up.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'down
    '
        Me.down.Image = CType(resources.GetObject("down.Image"),
System.Drawing.Image)
        Me.down.Location = New System.Drawing.Point(104, 272)
        Me.down.Size = New System.Drawing.Size(32, 24)
        Me.down.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'move
    '
        Me.move.Font = New System.Drawing.Font("Tahoma", 13.0!,
System.Drawing.FontStyle.Bold)
        Me.move.Location = New System.Drawing.Point(88, 296)
        Me.move.Size = New System.Drawing.Size(64, 20)
        Me.move.Text = "Move"
        Me.move.TextAlign = System.Drawing.ContentAlignment.TopCenter
    '
    'PictureBox2
    '
        Me.PictureBox2.Image =
CType(resources.GetObject("PictureBox2.Image"), System.Drawing.Image)
        Me.PictureBox2.Location = New System.Drawing.Point(8, 280)
        Me.PictureBox2.Size = New System.Drawing.Size(56, 32)
        Me.PictureBox2.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'PictureBox3
    '
        Me.PictureBox3.Image =
CType(resources.GetObject("PictureBox3.Image"), System.Drawing.Image)
        Me.PictureBox3.Location = New System.Drawing.Point(8, 240)
        Me.PictureBox3.Size = New System.Drawing.Size(56, 32)
        Me.PictureBox3.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'PictureBox4
    '
        Me.PictureBox4.Image =
CType(resources.GetObject("PictureBox4.Image"), System.Drawing.Image)
        Me.PictureBox4.Location = New System.Drawing.Point(176, 280)
        Me.PictureBox4.Size = New System.Drawing.Size(56, 32)
        Me.PictureBox4.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'PictureBox5
    '
        Me.PictureBox5.Image =
CType(resources.GetObject("PictureBox5.Image"), System.Drawing.Image)
        Me.PictureBox5.Location = New System.Drawing.Point(176, 240)
        Me.PictureBox5.Size = New System.Drawing.Size(56, 32)
        Me.PictureBox5.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'TrackBar1

```



```

    Me.TrackBar1.LargeChange = 4
    Me.TrackBar1.Location = New System.Drawing.Point(0, 216)
    Me.TrackBar1.Maximum = 20
    Me.TrackBar1.Size = New System.Drawing.Size(96, 45)
    Me.TrackBar1.SmallChange = 4
    Me.TrackBar1.TickFrequency = 4
    ' Panell
    Me.Panell.Controls.Add(Me.PictureBox1)
    Me.Panell.Size = New System.Drawing.Size(245, 216)
    ' PictureBox1
    Me.PictureBox1.Size = New System.Drawing.Size(245, 216)
    Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    Me.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.Fixed3D
    ' form2
    Me.BackColor = System.Drawing.Color.LightSkyBlue
    Me.ClientSize = New System.Drawing.Size(240, 316)
    Me.Controls.Add(Me.Panell)
    Me.Controls.Add(Me.TrackBar1)
    Me.Controls.Add(Me.PictureBox5)
    Me.Controls.Add(Me.PictureBox4)
    Me.Controls.Add(Me.PictureBox3)
    Me.Controls.Add(Me.PictureBox2)
    Me.Controls.Add(Me.move)
    Me.Controls.Add(Me.down)
    Me.Controls.Add(Me.up)
    Me.Controls.Add(Me.rightt)
    Me.Controls.Add(Me.home)
    Me.Controls.Add(Me.leftt)
    Me.Text = "eye2see"
    Me.WindowState = System.Windows.Forms.FormWindowState.Maximized

End Sub

```

#End Region

```

Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

```

```

    Dim sr As StreamReader = New StreamReader("\program
files\eye2see\counter.txt")
    Dim line As String

```

```

    Dim path As String = "\my documents\eye2see"
    v = 1
    ip = "212.36.193.4"

```

```

    Dim message, title, defaultValue As String
    Dim myValue As Object
    Dim done As Boolean
    done = False

```

```

While Not done
    message = "Enter the IP of the Cam" ' Set prompt.
    title = "IP Setting" ' Set title.
    defaultValue = "212.36.193.4" ' Set default value.
    ' Display message, title, and default value.
    myValue = InputBox(message, title, defaultValue, 10, 10)
    ip = myValue
    If ip = "" Then
        ip = defaultValue

    End If
    Try
        getframe(PictureBox1.Image)
        done = True
    Catch ex As Exception
        MsgBox("Incorrect IP")
    End Try
End While

```

```

If Not Directory.Exists(path) Then
    ' Try to create the directory.
    Dim di As DirectoryInfo = Directory.CreateDirectory(path)
End If

```

```

line = sr.ReadToEnd
sr.Close()

counter = Int(line)

```

```

Timer2.Interval = Form1.delay
Timer2.Enabled = True

'getframe(PictureBox1.Image)

```

End Sub

```

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
    Dim pic As Bitmap

    getframe(PictureBox1.Image)

```

End Sub

```

Private Sub home_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles home.Click
    move.Text = "HOME"
    movecamhome()
End Sub

```

```

Private Sub rightt_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles rightt.Click

    move.Text = "RIGHT"
    movecamright()

End Sub
Private Sub leftt_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles leftt.Click

    move.Text = "LEFT"
    movecamleft()

End Sub
Private Sub up_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles up.Click

    move.Text = "UP"
    movecamup()

End Sub
Private Sub down_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles down.Click

    move.Text = "DOWN"
    movecamdown()

End Sub

Private Sub PictureBox4_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles PictureBox4.Click
    Dim res As String

    res = MsgBox("Are you sure you want to exit?",
MsgBoxStyle.OKCancel)
    If res = "1" Then
        File.Delete("\program files\eye2see\counter.txt")
        Dim sr As StreamWriter = File.CreateText("\program
files\eye2see\counter.txt")
        sr.Write(Str(counter))
        sr.Close()
        Dim f1 As New Form1
        f1.Close()
        Me.Close()
        Application.Exit()
    End If
End Sub

Private Sub PictureBox5_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles PictureBox5.Click
    Dim f As New Form3
    PictureBox1.Visible = False
    f.Show()
End Sub

Private Sub PictureBox2_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles PictureBox2.Click

    Timer2.Enabled = False

```

```

        capturetofile(PictureBox1.Image)

        Timer2.Enabled = True

    End Sub

    Private Sub PictureBox3_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles PictureBox3.Click

        Dim f As New Form4
        f.Show()
        Me.Close()

    End Sub

    Private Sub TrackBar1_ValueChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar1.ValueChanged
        Dim oldh As Integer
        Dim oldw As Integer
        oldh = PictureBox1.Height
        oldw = PictureBox1.Width

        If TrackBar1.Value > v Then
            PictureBox1.Height = PictureBox1.Height * 1.2
            PictureBox1.Width = PictureBox1.Width * 1.2
            PictureBox1.Top = PictureBox1.Top - (PictureBox1.Height -
oldh) / 2
            PictureBox1.Left = PictureBox1.Left - (PictureBox1.Width -
oldw) / 2
        Else
            PictureBox1.Height = PictureBox1.Height / 1.2
            PictureBox1.Width = PictureBox1.Width / 1.2
            PictureBox1.Top = PictureBox1.Top + (oldh -
PictureBox1.Height) / 2
            PictureBox1.Left = PictureBox1.Left + (oldw -
PictureBox1.Width) / 2

            End If
            v = TrackBar1.Value
        End Sub
    End Class

```

```

Imports System.IO
Imports OpenNETCF.Win32.Notify
Public Class Form3
    Inherits System.Windows.Forms.Form
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents Label3 As System.Windows.Forms.Label
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        MyBase.Dispose(disposing)
    End Sub

    'NOTE: The following procedure is required by the Windows Form
Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents Label4 As System.Windows.Forms.Label
    Friend WithEvents PictureBox2 As System.Windows.Forms.PictureBox
    Friend WithEvents PictureBox3 As System.Windows.Forms.PictureBox
    Friend WithEvents MainMenu1 As System.Windows.Forms.MainMenu
    Friend WithEvents MenuItem1 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem2 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem3 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem4 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem5 As System.Windows.Forms.MenuItem
    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
        Dim resources As System.Resources.ResourceManager = New
System.Resources.ResourceManager(GetType(Form3))
        Me.PictureBox1 = New System.Windows.Forms.PictureBox
        Me.Label1 = New System.Windows.Forms.Label
        Me.Label2 = New System.Windows.Forms.Label
        Me.Label3 = New System.Windows.Forms.Label
        Me.Label4 = New System.Windows.Forms.Label
        Me.PictureBox2 = New System.Windows.Forms.PictureBox
        Me.PictureBox3 = New System.Windows.Forms.PictureBox
        Me.MainMenu1 = New System.Windows.Forms.MainMenu
        Me.MenuItem1 = New System.Windows.Forms.MenuItem
        Me.MenuItem2 = New System.Windows.Forms.MenuItem
        Me.MenuItem3 = New System.Windows.Forms.MenuItem
        Me.MenuItem5 = New System.Windows.Forms.MenuItem
        Me.MenuItem4 = New System.Windows.Forms.MenuItem
        '
        'PictureBox1
        '

```

```

        Me.PictureBox1.Image =
CType(resources.GetObject("PictureBox1.Image"), System.Drawing.Image)
        Me.PictureBox1.Location = New System.Drawing.Point(8, 32)
        Me.PictureBox1.Size = New System.Drawing.Size(64, 96)
        Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'Label1
    '
        Me.Label1.Font = New System.Drawing.Font("Tahoma", 15.0!,
System.Drawing.FontStyle.Bold)
        Me.Label1.Location = New System.Drawing.Point(24, 0)
        Me.Label1.Size = New System.Drawing.Size(176, 28)
        Me.Label1.Text = "eye2see ver. 1.0"
    '
    'Label2
    '
        Me.Label2.Location = New System.Drawing.Point(80, 32)
        Me.Label2.Size = New System.Drawing.Size(136, 104)
        Me.Label2.Text = "This surveillance system was developed by
Sami Souki in August 2004 under the sup" & _
        "ervision of Dr. Ramzi Haraty from LAU."
    '
    'Label3
    '
        Me.Label3.Location = New System.Drawing.Point(0, 240)
        Me.Label3.Size = New System.Drawing.Size(136, 20)
        Me.Label3.Text = "ssouki@lynx.net.lb"
        Me.Label3.TextAlign = System.Drawing.ContentAlignment.TopCenter
    '
    'Label4
    '
        Me.Label4.Location = New System.Drawing.Point(8, 128)
        Me.Label4.Size = New System.Drawing.Size(184, 48)
        Me.Label4.Text = "Main Resources: (D-Link)
DCS-5300 Internet Cam Microsoft Visu" & _
        "al Studio .Net"
    '
    'PictureBox2
    '
        Me.PictureBox2.Image =
CType(resources.GetObject("PictureBox2.Image"), System.Drawing.Image)
        Me.PictureBox2.Location = New System.Drawing.Point(8, 176)
        Me.PictureBox2.Size = New System.Drawing.Size(40, 40)
        Me.PictureBox2.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'PictureBox3
    '
        Me.PictureBox3.Image =
CType(resources.GetObject("PictureBox3.Image"), System.Drawing.Image)
        Me.PictureBox3.Location = New System.Drawing.Point(80, 176)
        Me.PictureBox3.Size = New System.Drawing.Size(120, 40)
        Me.PictureBox3.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'MainMenu1
    '
        Me.MainMenu1.MenuItems.Add(Me.MenuItem1)
    '

```

```

    'MenuItem1
    '
    Me.MenuItem1.MenuItems.Add(Me.MenuItem2)
    Me.MenuItem1.MenuItems.Add(Me.MenuItem3)
    Me.MenuItem1.MenuItems.Add(Me.MenuItem5)
    Me.MenuItem1.MenuItems.Add(Me.MenuItem4)
    Me.MenuItem1.Text = "Settings"
    '
    'MenuItem2
    '
    Me.MenuItem2.Text = "Delay"
    '
    'MenuItem3
    '
    Me.MenuItem3.Text = "Captured Pics"
    '
    'MenuItem5
    '
    Me.MenuItem5.Text = "Schedule Task"
    '
    'MenuItem4
    '
    Me.MenuItem4.Text = "Exit"
    Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None
    '
    'Form3
    '
    Me.BackColor = System.Drawing.Color.LightSkyBlue
    Me.ControlBox = False
    Me.Controls.Add(Me.PictureBox3)
    Me.Controls.Add(Me.PictureBox2)
    Me.Controls.Add(Me.Label4)
    Me.Controls.Add(Me.Label3)
    Me.Controls.Add(Me.Label2)
    Me.Controls.Add(Me.Label1)
    Me.Controls.Add(Me.PictureBox1)
    Me.Icon = CType(resources.GetObject("$this.Icon"),
System.Drawing.Icon)
    Me.Menu = Me.MainMenu1
    Me.Text = "About"
    Me.WindowState = System.Windows.Forms.FormWindowState.Maximized

```

End Sub

#End Region

```

    Private Sub MenuItem4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem4.Click
        Dim f As New form2
        f.Show()
        Me.Close()
    End Sub

```

```

    Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem3.Click
        Dim message, title, defaultValue As String
        Dim myValue As Object
        message = "Default name for captured images" ' Set prompt.
        title = "Captured Images" ' Set title.

```



```

        defaultValue = "captured"    ' Set default value.

        ' Display message, title, and default value.
        myValue = InputBox(message, title, defaultValue)
        If myValue <> "" Then
            Form1.picname = Trim(myValue)
        End If

    End Sub

    Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem2.Click
        Dim message, title, defaultValue As String
        Dim myValue As Object
        message = "Enter Delay Time in Milliseconds"    ' Set prompt.
        title = "Delay Time"    ' Set title.
        defaultValue = "100"    ' Set default value.
        myValue = 100
        ' Display message, title, and default value.
        myValue = InputBox(message, title, defaultValue, 10, 10)
        If myValue <> "" Then
            Form1.delay = Int(myValue)
        End If
    End Sub

    Private Sub MenuItem5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem5.Click
        Dim sr As StreamReader = New StreamReader("\program
files\eye2see\running.txt")
        Dim line As String

        line = sr.ReadToEnd
        sr.Close()

        If Int(line) = 1 Then
            MsgBox("A task is already running")
        Else
            Notify.RunAppAtTime("\program files\schedule\schedule.exe",
Date.Now)
        End If
    End Sub

End Class
Imports System.IO
Public Class Form4
    Inherits System.Windows.Forms.Form
    Friend WithEvents ComboBox1 As System.Windows.Forms.ComboBox
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.

```



```

InitializeComponent()

'Add any initialization after the InitializeComponent() call

End Sub

Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
    MyBase.Dispose(disposing)
End Sub

'NOTE: The following procedure is required by the Windows Form
Designer
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
Friend WithEvents PictureBox4 As System.Windows.Forms.PictureBox
Friend WithEvents Label1 As System.Windows.Forms.Label
<System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
    Dim resources As System.Resources.ResourceManager = New
System.Resources.ResourceManager(GetType(Form4))
    Me.ComboBox1 = New System.Windows.Forms.ComboBox
    Me.PictureBox1 = New System.Windows.Forms.PictureBox
    Me.PictureBox4 = New System.Windows.Forms.PictureBox
    Me.Label1 = New System.Windows.Forms.Label
    '
    'ComboBox1
    '
    Me.ComboBox1.Size = New System.Drawing.Size(240, 22)
    '
    'PictureBox1
    '
    Me.PictureBox1.Location = New System.Drawing.Point(0, 32)
    Me.PictureBox1.Size = New System.Drawing.Size(240, 240)
    Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'PictureBox4
    '
    Me.PictureBox4.Image =
CType(resources.GetObject("PictureBox4.Image"), System.Drawing.Image)
    Me.PictureBox4.Location = New System.Drawing.Point(184, 288)
    Me.PictureBox4.Size = New System.Drawing.Size(56, 32)
    Me.PictureBox4.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'Label1
    '
    Me.Label1.Font = New System.Drawing.Font("Monotype Corsiva",
15.0!, CType((System.Drawing.FontStyle.Bold Or
System.Drawing.FontStyle.Italic), System.Drawing.FontStyle))
    Me.Label1.Location = New System.Drawing.Point(0, 288)
    Me.Label1.Size = New System.Drawing.Size(184, 24)
    Me.Label1.Text = "Browse Captured Pics"
    '
    'Form4
    '
    Me.BackColor = System.Drawing.Color.LightSkyBlue
    Me.ClientSize = New System.Drawing.Size(244, 320)
    Me.Controls.Add(Me.Label1)

```

```
Me.Controls.Add(Me.PictureBox4)
Me.Controls.Add(Me.PictureBox1)
Me.Controls.Add(Me.ComboBox1)
Me.Text = "Form4"
Me.WindowState = System.Windows.Forms.FormWindowState.Maximized

End Sub

#End Region

Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    Dim dirs As String() = Directory.GetFiles("\my
documents\eye2see")
    Dim dir As String

    For Each dir In dirs

        ComboBox1.Items.Add(dir)
    Next
End Sub

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged
    Dim p As Bitmap
    Dim f As File
    p = New Bitmap(ComboBox1.Text)
    PictureBox1.Image = p

End Sub

Private Sub PictureBox4_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles PictureBox4.Click
    Dim f As New form2
    f.Show()
    Me.Close()
End Sub
End Class
```

```

Imports System.Net
Imports System.IO

Public Class Form1
    Inherits System.Windows.Forms.Form
    Friend WithEvents starttimepick As
OpenNETCF.Windows.Forms.DateTimePicker
    Friend WithEvents endtimepick As
OpenNETCF.Windows.Forms.DateTimePicker
    Friend WithEvents MainMenu1 As System.Windows.Forms.MainMenu

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overrides Sub Dispose(ByVal disposing As
Boolean)
        MyBase.Dispose(disposing)
    End Sub

    'NOTE: The following procedure is required by the Windows Form
Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
    Friend WithEvents PictureBox3 As System.Windows.Forms.PictureBox
    Friend WithEvents PictureBox4 As System.Windows.Forms.PictureBox
    Friend WithEvents Timer1 As System.Windows.Forms.Timer
    Friend WithEvents Timer2 As System.Windows.Forms.Timer
    Friend WithEvents Timer3 As System.Windows.Forms.Timer
    Friend WithEvents PictureBox2 As System.Windows.Forms.PictureBox
    Private Sub InitializeComponent()
        Dim resources As System.Resources.ResourceManager = New
System.Resources.ResourceManager(GetType(Form1))
        Me.starttimepick = New OpenNETCF.Windows.Forms.DateTimePicker
        Me.MainMenu1 = New System.Windows.Forms.MainMenu
        Me.endtimepick = New OpenNETCF.Windows.Forms.DateTimePicker
        Me.PictureBox1 = New System.Windows.Forms.PictureBox
        Me.PictureBox3 = New System.Windows.Forms.PictureBox
        Me.PictureBox4 = New System.Windows.Forms.PictureBox
        Me.Timer1 = New System.Windows.Forms.Timer
        Me.Timer2 = New System.Windows.Forms.Timer
        Me.Timer3 = New System.Windows.Forms.Timer
        Me.PictureBox2 = New System.Windows.Forms.PictureBox
        '
        'starttimepick
        '
        Me.starttimepick.CalendarForeColor =
System.Drawing.SystemColors.ControlText
        Me.starttimepick.CalendarMonthBackground =
System.Drawing.SystemColors.Window

```



```

        Me.starttimepick.CalendarTitleBackColor =
System.Drawing.SystemColors.ActiveCaption
        Me.starttimepick.CalendarTitleForeColor =
System.Drawing.SystemColors.ActiveCaptionText
        Me.starttimepick.CalendarTrailingForeColor =
System.Drawing.SystemColors.GrayText
        Me.starttimepick.CustomFormat = "ddd, MMM dd, yyyy,h:mm:ss tt"
        Me.starttimepick.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular)
        Me.starttimepick.Format =
OpenNETCF.Windows.Forms.DateTimePickerFormat.Custom
        Me.starttimepick.Location = New System.Drawing.Point(8, 80)
        Me.starttimepick.MaxDate = New Date(9990, 12, 31, 0, 0, 0)
        Me.starttimepick.MinDate = New Date(1753, 1, 1, 0, 0, 0)
        Me.starttimepick.ShowCheckBox = False
        Me.starttimepick.ShowUpDown = False
        Me.starttimepick.Size = New System.Drawing.Size(232, 24)
        Me.starttimepick.Text = "DateTimePicker1"
        Me.starttimepick.Value = New Date(2004, 8, 31, 22, 14, 46, 946)
    '
    'endtimepick
    '
        Me.endtimepick.CalendarForeColor =
System.Drawing.SystemColors.ControlText
        Me.endtimepick.CalendarMonthBackground =
System.Drawing.SystemColors.Window
        Me.endtimepick.CalendarTitleBackColor =
System.Drawing.SystemColors.ActiveCaption
        Me.endtimepick.CalendarTitleForeColor =
System.Drawing.SystemColors.ActiveCaptionText
        Me.endtimepick.CalendarTrailingForeColor =
System.Drawing.SystemColors.GrayText
        Me.endtimepick.CustomFormat = "ddd, MMM dd, yyyy,h:mm:ss tt"
        Me.endtimepick.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular)
        Me.endtimepick.Format =
OpenNETCF.Windows.Forms.DateTimePickerFormat.Custom
        Me.endtimepick.Location = New System.Drawing.Point(8, 160)
        Me.endtimepick.MaxDate = New Date(9990, 12, 31, 0, 0, 0)
        Me.endtimepick.MinDate = New Date(1753, 1, 1, 0, 0, 0)
        Me.endtimepick.ShowCheckBox = False
        Me.endtimepick.ShowUpDown = False
        Me.endtimepick.Size = New System.Drawing.Size(232, 24)
        Me.endtimepick.Text = "DateTimePicker1"
        Me.endtimepick.Value = New Date(2004, 8, 31, 22, 14, 46, 946)
    '
    'PictureBox1
    '
        Me.PictureBox1.Image =
CType(resources.GetObject("PictureBox1.Image"), System.Drawing.Image)
        Me.PictureBox1.Location = New System.Drawing.Point(72, 32)
        Me.PictureBox1.Size = New System.Drawing.Size(100, 40)
        Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    '
    'PictureBox3
    '
        Me.PictureBox3.Image =
CType(resources.GetObject("PictureBox3.Image"), System.Drawing.Image)
        Me.PictureBox3.Location = New System.Drawing.Point(72, 112)

```

```

        Me.PictureBox3.Size = New System.Drawing.Size(100, 40)
        Me.PictureBox3.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        'PictureBox4
        '
        Me.PictureBox4.Image =
CType(resources.GetObject("PictureBox4.Image"), System.Drawing.Image)
        Me.PictureBox4.Location = New System.Drawing.Point(16, 224)
        Me.PictureBox4.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        'Timer1
        '
        'Timer2
        '
        'Timer3
        '
        'PictureBox2
        '
        Me.PictureBox2.Image =
CType(resources.GetObject("PictureBox2.Image"), System.Drawing.Image)
        Me.PictureBox2.Location = New System.Drawing.Point(136, 224)
        Me.PictureBox2.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        Me.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.Fixed3D
        'Form1
        '
        Me.BackColor = System.Drawing.Color.LightSkyBlue
        Me.ClientSize = New System.Drawing.Size(244, 320)
        Me.Controls.Add(Me.PictureBox2)
        Me.Controls.Add(Me.PictureBox4)
        Me.Controls.Add(Me.PictureBox3)
        Me.Controls.Add(Me.PictureBox1)
        Me.Controls.Add(Me.endtimepick)
        Me.Controls.Add(Me.starttimepick)
        Me.MaximizeBox = False
        Me.Menu = Me.MainMenu1
        Me.Text = "Schedule"
        Me.WindowState = System.Windows.Forms.FormWindowState.Maximized

    End Sub

#End Region
    Public counter As Integer
    Public p As Image
    Public path As String
    Public Sub capturetofile(ByVal p As Image)
        SaveThisImage.BitmapSaver.SaveBitmapToFile(p, path + "\" +
"pic" + Str(counter) + ".jpg")
        counter = counter + 1
    End Sub

    Public Sub getframe(ByRef p As Bitmap)

```

```

        Dim request As WebRequest =
WebRequest.Create("http://212.36.193.4/cgi-bin/video.jpg")

        Dim stream As Stream =
request.GetResponse().GetResponseStream()
        p = New Bitmap(stream)
        stream.Close()

    End Sub

    Private Sub PictureBox4_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles PictureBox4.Click
        Dim f As New Form1
        If starttimepick.Value < Date.Now Then
            MsgBox("Incorrect Start Time")
        ElseIf starttimepick.Value >= endtimepick.Value Then
            MsgBox("Incorrect End Time")
        Else
            Timer1.Enabled = True
            Timer1.Interval = 1000 * DateDiff(DateInterval.Second,
Date.Now, starttimepick.Value)
            Timer3.Interval = 1000 * DateDiff(DateInterval.Second,
starttimepick.Value, endtimepick.Value)

            Dim sr As StreamWriter = File.CreateText("\program
files\eye2see\running.txt")
            sr.Write(Str(1))
            sr.Close()

            Me.Hide()

        End If
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        Dim c As Integer
        c = 1
        If Not Directory.Exists("\storage card") Then
            path = "\my documents\eye2seeCapt"
        Else
            path = "\storage card\eye2seeCapt"
        End If

        If Not Directory.Exists(path) Then
            Directory.CreateDirectory(path)
        End If
        While Directory.Exists(path + "\schedule" + Str(c))
            c = c + 1
        End While
        path = path + "\schedule" + Str(c)
        Directory.CreateDirectory(path)
        starttimepick.Value = Date.Now
        endtimepick.Value = Date.Now
        counter = 1

    End Sub

```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick

    Timer3.Enabled = True
    Timer2.Enabled = True
    Timer2.Interval = 100
    Timer1.Enabled = False
End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick

    getframe(p)
    capturetofile(p)

End Sub

Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer3.Tick

    Timer2.Enabled = False
    Timer3.Enabled = False
    Dim sr As StreamWriter = File.CreateText("\program
files\eye2see\running.txt")
    sr.Write(Str(0))
    sr.Close()
    Application.Exit()

End Sub
Private Sub PictureBox2_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles PictureBox2.Click
    Application.Exit()
End Sub
End Class
```



```

using System;
using System.Drawing;
using System.IO;
using System.Runtime.InteropServices;

namespace SaveThisImage
{
    /// <summary>
    /// Summary description for BitmapSaver.
    /// </summary>
    public class BitmapSaver
    {
        /// <summary>
        /// Saves an image (image) at the location defined
        (szPath). This uses the generic
        /// 24-bit format, so the image is not compressed.
        /// </summary>
        /// <param name="image">The bitmap image to store</param>
        /// <param name="szPath">The path to save at</param>
        static public void SaveBitmapToFile(Image image, string
        szPath)
        {
            FileStream fs = File.Create(szPath);

            SaveBitmapToStream(image, fs);

            fs.Close();
        }

        /// <summary>
        /// Saves an image (image) at the location defined
        (szPath). This uses the generic
        /// 24-bit format, so the image is not compressed.
        /// </summary>
        /// <param name="image">The bitmap image to store</param>
        /// <param name="stream">The stream to save to</param>
        static public void SaveBitmapToStream(Image image, Stream
        stream)
        {

```

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/bitmaps_4vlh.asp.

* Sort of from
 * The below is hardcoded to support saving a 24-bit
 bitmap. Slight changes would be made to certain
 * constants in order to save 16-bit or 256-color (8-
 bit) or whatever. The resulting file is probably
 * the largest of any image format, which is at the
 benefit of speed. There are other ways of saving
 * bitmaps (such as using color indexing) which will
 reduce the bitmap size, but they also add
 * performance hits.

```

*
* ---Data---          ---Bytes---
* "BM"                0x00-0x01
* File Size           0x02-0x05
* Reserved 1          0x06-0x07
* Reserved 2          0x08-0x09
* Constant 0x36      0x0A
* Constant 0x00      0x0B-0x0D

```



```

0x00)
    * Constant 0x28 0x0E (This byte is always 0x28)
    * Constant 0x00 0x0F-0x11 (These bytes are always
    * Bitmap Width 0x12-0x15
    * Bitmap Height 0x16-0x19
    * Constant 0x01 0x1A
    * Constant 0x00 0x1B
    * Constant 0x18 0x1C
    * Constant 0x00 0x1D-0x21
    * Image byte size 0x22-0x25
    * Constant 0x00 0x26-0x35
    * Pixels 0x36-Rest
    * These are 3-byte RGB
values for pixels starting at bottom left, moving right.
    * They are in G-B-R
order, and you have to remember to make the full stride (row) a
    * multiple of 4, meaning
that if you have a width of 3, that's only 9 bytes and you
    * have to pad with 3 to
round it to the next multiple of 4.
    */

```

```

    Bitmap bm = (Bitmap) image;
    const int BYTES_PER_PIXEL = 3;

    // Need 0x36 bytes for the headers, plus all of the
pixel data, so round up to nearest 4
    int nBytes = 0x36 + (bm.Height * ((BYTES_PER_PIXEL *
bm.Width) + 0x03) & ~0x03);

    byte[] BitmapData = new byte[nBytes];
    BitmapData[0x00] = (byte) 'B';
    BitmapData[0x01] = (byte) 'M';

    BitmapData[0x02] = (byte) nBytes;
    BitmapData[0x03] = (byte) (nBytes >> 8);
    BitmapData[0x04] = (byte) (nBytes >> 16);
    BitmapData[0x05] = (byte) (nBytes >> 24);

    BitmapData[0x0A] = 0x36;
    BitmapData[0x0E] = 0x28;

    BitmapData[0x12] = (byte) bm.Width;
    BitmapData[0x13] = (byte) (bm.Width >> 8);
    BitmapData[0x14] = (byte) (bm.Width >> 16);
    BitmapData[0x15] = (byte) (bm.Width >> 24);

    BitmapData[0x16] = (byte) bm.Height;
    BitmapData[0x17] = (byte) (bm.Height >> 8);
    BitmapData[0x18] = (byte) (bm.Height >> 16);
    BitmapData[0x19] = (byte) (bm.Height >> 24);

    BitmapData[0x1A] = 0x01;
    BitmapData[0x1C] = 0x18;

    BitmapData[0x22] = (byte) (nBytes - 0x36);
    BitmapData[0x23] = (byte) ((nBytes - 0x36) >> 8);
    BitmapData[0x24] = (byte) ((nBytes - 0x36) >> 16);

```

```

        BitmapData[0x25] = (byte) ((nBytes - 0x36) >> 24);

        // Stripping bitmap from bottom left, moving right
through the row, then up to the next row
        int index = 0x36;
        for (int h = bm.Height - 1; h >= 0; h--)
        {
            for (int w = 0; w < bm.Width; w++)
            {
                int c = bm.GetPixel(w, h).ToArgb();
                BitmapData[index++] = (byte) c;
                BitmapData[index++] = (byte) (c >> 8);
                BitmapData[index++] = (byte) (c >> 16);
            }

            // Padding the end of the row

            int xtra = (bm.Width * 3) % 4;
            if (xtra != 0)
            {
                index += 4 - xtra;
            }
        }

        // Write the bytes to the stream
        stream.Write(BitmapData, 0, BitmapData.Length);
    }
}
}

```

Setup

After the application is compiled successfully and the build process accomplished in Visual Studio .NET 2003, a CAB file is created and can be transferred to the PDA to be installed.

Basic requirements to run the application:

- 1- The DCS-5300 cam should be connected to the Internet either through a direct real IP or through a PC, notebook or a router. If the cam is connected to a PC (without router) Network Address Translation (NAT) should be used to forward and tunnel the requests of the PDA to access the cam.
- 2- The PDA should be connected to the Internet either through dial-up or GPRS. The latter is preferable for its speed and flexibility.
- 3- After a successful installation of the application, the IP of the cam should be provided to the eye2see software to establish connection.

The attached floppy disk contains the CAB file that should be transferred to the PPC to be installed.

Conclusion

To conclude, it is worth mentioning that mobile computing is a very essential and hot area in this century and should be taken into consideration by implementing a wide variety of application that may simulate existing ones supported only for personal computers or invent new applications.

Future Work

As for future enhancements to the eye2see application the following could be done:

- 1- The capability to transmit audio.
- 2- Transmitting streaming video instead of frames.
- 3- The ability to schedule infinite capturing tasks.
- 4- An interesting feature could be programmed to produce an alert whenever a motion occurs in front of the cam.

References

- 1- MSDN electronic Library (3CDs), Microsoft Corporation
- 2- Visual Studio.Net 2003 Architect Edition documentation, Microsoft Corporation
- 3- <http://www.microsoft.com/newsgroups>
(discussions in dotnet.framework.compact framework)
- 4- "COM programming by example" by swanke, John
- 5- <http://groups.yahoo.com/groups/visual-studio-dotnet> (yahoo group).